



NLR-TR-2008-467

**Solution of the vector wave equation using a
Krylov solver with an algebraic multigrid
approximated preconditioner**

Master Thesis for the Degree of Master of Science in
Applied Mathematics



Executive Summary

Solution of the vector wave equation using a Krylov solver with an algebraic multigrid approximated preconditioner

Master Thesis for the Degree of Master of Science in Applied Mathematics

Problem area

Radar cross section prediction techniques are used to analyze the radar signature of military platforms when the radar signature cannot be determined experimentally. This may be the case when, for example, the platform is in the design, development or procurement phase; or when the platform belongs to a hostile party.

For jet powered fighter aircraft, the radar signature is dominated by the contribution of the jet engine air intake for a large range of forward observation angles. The intake can be regarded as a one-sided open large and deep forward facing cavity. Although the contribution of the outer mould shape of the platform can be efficiently and accurately computed using scattering models, these cannot be used to accurately compute the contribution of the jet engine air intake. The storage requirements of the existing solution algorithm for the jet engine air intake, are too stringent which prohibits the application to the relevant excitation

frequency band.

Description of work

To deal with the storage requirements of the existing solution algorithm, alternative solution methods are analyzed and compared to the original formulation. More specifically, it is analyzed how to incorporate so called *multigrid acceleration* in the existing algorithm. To derive an optimal strategy to solve the current application, a model problem is used for testing purposes.

Results and conclusions

Based on the computational results with the model problem, it is estimated that the application of the algorithms developed in this report, will result in a speedup in both computing time and memory usage of two to three orders in magnitude.

Applicability

The developed technology will be applied for the analysis and possible optimization of jet engine air intake geometries of current intermediate observable and future low observable fighter aircraft.

Report no.

NLR-TR-2008-467

Author(s)

S.M.F. Abdoel

Classification report

Unclassified

Date

February 2009

Knowledge area(s)

Numerical Mathematics

Descriptor(s)

Radar
RCS
Large sparse systems
Iterative methods
Algebraic multigrid
Preconditioning

NLR-TR-2008-467

Solution of the vector wave equation using a Krylov solver with an algebraic multigrid approximated preconditioner

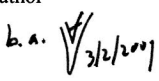
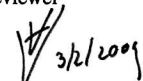
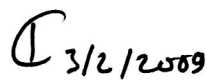
Master Thesis for the Degree of Master of Science in Applied Mathematics

S.M.F. Abdoel

No part of this report may be reproduced and/or disclosed, in any form or by any means, without the prior written permission of the owner.

Customer National Aerospace Laboratory NLR
Contract number ----
Owner National Aerospace Laboratory NLR
Division Aerospace Vehicles
Distribution Limited
Classification of title Unclassified
February 2009

Approved by:

Author b. a.  3/2/2009	Reviewer  3/2/2009	Managing department  3/2/2009
--	--	---

Summary

RADAR (**R**adio **D**etection and **R**anging) is technology to detect aircraft and ships by using electromagnetic waves. A measure of this detectability is the *radar cross section* (RCS). Generally, it is known that the contribution of the jet engine air intake of a modern fighter aircraft accounts for the major part of the RCS of the total aircraft, if the platform is excited from the front. The properties of the scattered electric and magnetic fields can be described by the Maxwell equations whereupon the *vector wave equation* can be derived. This equation is discretized by the finite element method resulting in a large system of linear equations.

In the present implementation, the iterative Krylov subspace method used to solve this linear system is the *Generalized Conjugate Residual* (GCR) method. As the system matrix is highly ill-conditioned, the convergence of the GCR method is generally slow. To improve the convergence rate, the shifted Laplace operator is used as a preconditioner for the discretized vector wave equation. As the memory requirements become difficult to satisfy when the number of degrees of freedom increases, this solution procedure cannot be used for very large systems. To overcome these difficulties, the existing algorithm will be modified such that a *Multigrid solution method* is incorporated.

Multigrid can be classified as geometric or algebraic, depending on the structuredness of the underlying grid. When the grid is unstructured or irregular, *algebraic multigrid* can be used. In this case, the coefficients in the system matrix will be used to specify the coarse grid operator, the prolongation and restriction operator, etc., without any information about the computational grid. The *algebraic multigrid* algorithm chosen in this thesis is a Multilevel (ML) Preconditioning Package developed by Sandia National Laboratories (see ref. 7).

Another favorable consequence of using multigrid for the preconditioner solve, is that it leads to a *constant preconditioner system*. Retaining a constant preconditioner matrix throughout the solution sequence, means that the system can be solved by a short recurrence method e.g. *stabilized Bi-Conjugate Gradient* (Bi-CGSTAB) method or IDR(4), instead of the currently used GCR algorithm.



Acknowledgments

This work could not have been finished without the support of a number of people I would like to thank.

First I would like to thank the National Aerospace Laboratory (NLR) for the possibility to let me perform my master thesis there and in particular the manager of the division Flight Physics and Loads, ir. Koen de Cock. I also would like to thank my direct supervisor on location, Harmen van der Ven for the guidance and patience he had during the time I spent at NLR.

I would also like to thank my supervisors at the Delft University of Technology, prof. dr. ir. Kees Vuik and dr. ir. Duncan van der Heul. Without the guidance of and cooperation with these supervisors it was an impossible task to complete my thesis.

During several meetings in Delft, I consulted some people I would like to thank. They are : ir. P. Sonneveld, dr. ir. M. van Gijzen and dr. D.J.P. Lahaye.

Furthermore, I would like to thank my parents and sister for the good time I had during the home visits in the weekends. During the midweek I had the opportunity to stay with my brother-in-law and I want to thank him for that. Last, but not least, I would like to thank my friend for always being there for me. And his parents for the time I spent with them.

Contents

List of figures	8
List of tables	11
1 Introduction	15
2 An algorithm for full wave analysis of cavity scattering	17
2.1 Introduction	17
2.2 Physical model	17
2.3 Finite element discretization method	23
2.4 Properties of the linear system	30
3 Iterative solution of the discretized vector wave equation	32
3.1 Introduction	32
3.2 Present implementation	32
3.3 Incorporation of algebraic multigrid in the existing algorithm	35
3.4 Numeric results	38
3.5 Conclusions	40
4 Iterative solution of the Helmholtz equation	42
4.1 Introduction	42
4.2 Application of the shifted Laplace preconditioner for the two dimensional Helmholtz equation	43
4.3 Differences between a small cavity scattering model and the three dimensional Helmholtz equation	46
4.4 Conclusion	51
5 Short recurrence Krylov methods for linear systems with complex eigenvalues with an imaginary part relatively larger than their real part	52
5.1 Introduction	52
5.2 The spectral properties of the preconditioner system and the preconditioned system	52
5.3 Conclusions	60

6	Iterative solution of the two dimensional vector wave equation	61
6.1	Introduction	61
6.2	Discretization	66
6.3	Conclusions	77
7	Progress evaluation	78
7.1	Model problems	78
7.2	Conclusions	83
8	Conclusions	85
9	Future research	87
	References	89
	22 Tables	
	24 Figures	
Appendix A	Electromagnetic quantities	91
	(1 Table)	
Appendix B	Useful definitions and fundamental relations	92
Appendix C	Sandia's multilevel preconditioning package	95
C.1	General application of ML	95
C.2	ML in the current application	95
Appendix D	Multigrid appendix	96
D.1	Coarse grid correction scheme, two-grid and multigrid cycle	96
D.2	V and W cycles	97
	(3 Figures)	
Appendix E	Model problems	99
E.1	The two dimensional Poisson equation with Dirichlet boundary conditions	99
E.2	The two dimensional Helmholtz equation with local absorbing boundary conditions	101
	(12 Tables)	

Appendix F (6 Figures)	Comparison between two dimensional Maxwell solver and COMSOL	105
Appendix G G.1 G.2 (8 Figures)	Sparsity patterns of the different preconditioners The two dimensional vector wave equation The three dimensional vector wave equation	109 109 112
Appendix H	Using M_{loc} instead of M_{gl}	114

(114 pages in total)

List of figures

Figure 2.2.1 Schematic view of cylindrical cavity with length L and cross section diameter d .	20
Figure 2.2.2 The RCS σ of a metallic sphere with radius a illustrates the three scattering regions.	22
Figure 2.3.1 The ordering of Table 1 is used here to number the edges of the tetrahedron.	25
Figure 2.3.2 An example of the discretization of the interior of an S-shaped cavity using tetrahedral elements.	26
Figure 2.3.3 Left: The wave front enters the cavity with incidence angle ϕ . Two waves with initial phase difference $\psi_{in} = \frac{\lambda}{4}$ are followed. After reflection through the cavity there is an accumulated phase error ε – Middle: The exact phase difference ψ_{out} – Right: The computed phase difference $\tilde{\psi}_{out}$.	29
Figure 2.4.1 $A \in \mathbb{C}^{N \times N}$, $N = 723$, $h = 0.25$. Dimensions rectangular cavity: $1.5\lambda \times 1.5\lambda \times 0.6\lambda$. Fully populated block has dimension 99. The total number of nonzeros is 19.189. The complex valued part of the matrix consists of the unknowns on the aperture only.	30
Figure 3.4.1 Rectangular cavity with dimensions $1.5\lambda \times 1.5\lambda \times 0.6\lambda$. The discretization contains 2610 elements and 2796 degrees of freedom (N) for mesh size $h = 0.20$ and zeroth order basis functions.	39
Figure 4.2.1 Surface plot of the real part of the solution u .	44
Figure 4.2.2 Logarithm of the real part of the residual vector against the total number of matrix vector operations.	45
Figure 4.3.1 Spectrum for $M^{-1}A$ with shift $(\beta_1, \beta_2) = (1, -0.5)$.	49
Figure 4.3.2 Spectrum for $M^{-1}A$ with shift $(\beta_1, \beta_2) = (-1, 0)$.	50
Figure 4.3.3 Spectrum for $M^{-1}A$ with shift $(\beta_1, \beta_2) = (-1, 0)$ after rotation.	50
Figure 5.2.1 Different spectra for preconditioner $M_1^{-1}A$ and different combinations of the shift (β_1, β_2) .	53
Figure 5.2.2 Different spectra for preconditioner $M_2^{-1}A$ and different combinations of the shift (β_1, β_2) .	54
Figure 5.2.3 Spectrum for M_{11} for $M = M_1$ and different combinations of the shift (β_1, β_2) .	55
Figure 5.2.4 ω_k in the complex plane for preconditioner M_1 and shift $(\beta_1, \beta_2) = (-1, 0)$.	56
Figure 5.2.5 ω_k in the complex plane for preconditioner M_1 and shift $(\beta_1, \beta_2) = (1, -0.5)$.	57
Figure 6.1.1 Flowchart for the two dimensional vector wave equation.	63
Figure 6.1.2 Two dimensional cavity with dimensions $L \times d$.	64

Figure 6.1.3 Contour plot of E_z .	65
Figure 6.1.4 Vector plot of E_x, E_y .	66
Figure 6.1.5 Contour plot of H_z .	66
Figure 6.2.1 The discretized two dimensional cavity with mesh size $h = \frac{1}{8}$.	67
Figure 7.1.1 The CPU time as a function of the number of degrees of freedom on the aperture for the preconditioned GCR method and the frontal solver for a model problem with dimensions $4\lambda \times 1.5\lambda \times 1.5\lambda$ and $N = 79,266$.	83
Figure D.2.1 V-cycles for different coarse grid levels and $\gamma = 1$.	98
Figure D.2.2 W-cycles for different coarse grid levels and $\gamma = 2$.	98
Figure D.2.3 Full multigrid V-cycle.	98
Figure F.0.1 COMSOL surface plot for two dimensional cavity with height 1 and depth 8.	105
Figure F.0.2 Two dimensional Maxwell solver: surface plot for two dimensional cavity with height 1 and depth 8.	106
Figure F.0.3 COMSOL surface plot for two dimensional cavity with height 2 and depth 8.	106
Figure F.0.4 Two dimensional Maxwell solver: surface plot for two dimensional cavity with height 2 and depth 8.	107
Figure F.0.5 COMSOL surface plot for two dimensional cavity with height 4 and depth 8.	107
Figure F.0.6 Two dimensional Maxwell solver: surface plot for two dimensional cavity with height 4 and depth 8.	108
Figure G.1.1 Two dimensional vector wave equation, node based FEM implementation: sparsity pattern for preconditioner M_1 and local absorbing boundary conditions.	109
Figure G.1.2 Two dimensional vector wave equation, node based FEM implementation: sparsity pattern for preconditioner M_2 and local absorbing boundary conditions.	109
Figure G.1.3 Two dimensional vector wave equation, node based FEM implementation: sparsity pattern for preconditioner M_1 and global absorbing boundary conditions.	110
Figure G.1.4 Two dimensional vector wave equation, node based FEM implementation: sparsity pattern for preconditioner M_2 and global absorbing boundary conditions.	110
Figure G.1.5 Two dimensional vector wave equation, edge based FEM implementation: sparsity pattern for preconditioner M_1 and local absorbing boundary conditions.	111

- Figure G.1.6 Two dimensional vector wave equation, edge based FEM implementation: sparsity pattern for preconditioner M_2 and local absorbing boundary conditions. 111
- Figure G.2.1 Three dimensional sparsity pattern for preconditioner M_1 and global absorbing boundary conditions. Zeroth order basis functions are used here and the mesh size $h = 0.15$ for the cavity with dimensions $1.5\lambda \times 1.5\lambda \times 0.6\lambda$. 112
- Figure G.2.2 Three dimensional sparsity pattern for preconditioner M_2 and global absorbing boundary conditions. Zeroth order basis functions are used here and the mesh size $h = 0.15$ for the cavity with dimensions $1.5\lambda \times 1.5\lambda \times 0.6\lambda$. 113

List of tables

Table 1	Edge definition for a tetrahedral element.	25
Table 2	Total number of matrix vector products for the Bi-CGSTAB–ML-AMG algorithm to solve a small cavity model problem. In these experiments M_{new} is used as preconditioner (see Equation (3.2.6)).	39
Table 3	CPU-times for the Bi-CGSTAB–ML-AMG algorithm to solve a small cavity model problem.	40
Table 4	Parameter settings.	44
Table 5	Total number of MAT-VEC-OP for the two dimensional Helmholtz equation: fixed wavenumber $k_0 = 30$, varying N .	46
Table 6	Total number of MAT-VEC-OP for the two dimensional Helmholtz equation: fixed $N = 101^2$, varying k_0 .	46
Table 7	Total number of matrix vector operations for the preconditioned Bi-CGSTAB method with and without rotation of the spectrum for $M^{-1}A$.	51
Table 8	Number of matrix vector multiplications and elapsed time for the three dimensional Helmholtz equation with and without improved computation of ω_k .	58
Table 9	Overview of the model problems with the discretization type used and the absorbing boundary conditions imposed.	64
Table 10	Total number of matrix vector products for the IDR(4)-method to solve a two dimensional cavity model problem with preconditioner M_1 . A node based FEM implementation is considered and the system matrix as well as the preconditioner have local absorbing boundary conditions imposed on the aperture. 71	
Table 11	Total number of matrix vector products for the IDR(4)-method to solve a two dimensional cavity model problem with preconditioner M_2 . A node based FEM implementation is considered and the system matrix as well as the preconditioner have local absorbing boundary conditions imposed on the aperture. 71	
Table 12	Total number of matrix vector products for the IDR(4)-method to solve a two dimensional cavity model problem with preconditioner M_1 . A node based FEM implementation is considered and the system matrix as well as the preconditioner have global absorbing boundary conditions imposed.	72

Table 13	Total number of matrix vector products for the IDR(4)-method to solve a two dimensional cavity model problem with preconditioner M_2 . A node based FEM implementation is considered and the system matrix as well as the preconditioner have global absorbing boundary conditions imposed on the aperture.	73
Table 14	Total number of matrix vector products for the IDR(4) algorithm to solve a two dimensional cavity model problem. M_1 is chosen as preconditioner in this experiment with shift $(\beta_1, \beta_2) = (1, -0.5)$ and an exact solve for the preconditioner system. The mesh size $h = \frac{1}{32}$ and the total number of unknowns $N = 7937$.	73
Table 15	Total number of matrix vector products for the IDR(4) algorithm to solve a two dimensional cavity model problem. M_2 is chosen as preconditioner in this experiment with shift $(\beta_1, \beta_2) = (1, -0.5)$ and an exact solve for the preconditioner system. The mesh size $h = \frac{1}{32}$ and the total number of unknowns $N = 7937$.	73
Table 16	Total number of matrix vector products for the IDR(4) algorithm to solve a two dimensional cavity model problem. M_1 is chosen as preconditioner in this experiment with shift $(\beta_1, \beta_2) = (1, -0.5)$ and an exact solve for the preconditioner system. The mesh size $h = \frac{1}{64}$ and the total number of unknowns $N = 32,257$.	74
Table 17	Total number of matrix vector products for the IDR(4) algorithm to solve a two dimensional cavity model problem. M_2 is chosen as preconditioner in this experiment with shift $(\beta_1, \beta_2) = (1, -0.5)$ and an exact solve for the preconditioner system. The mesh size $h = \frac{1}{64}$ and the total number of unknowns $N = 32,257$.	74
Table 18	Total number of matrix vector products for the IDR(4)-method to solve a two dimensional cavity model problem with preconditioner M_1 . An edge based FEM implementation is considered and the system matrix as well as the preconditioner have local absorbing boundary conditions imposed on the aperture.	76
Table 19	Total number of matrix vector products for the IDR(4)-method to solve a two dimensional cavity model problem with preconditioner M_2 . An edge based FEM implementation is considered and the system matrix as well as the preconditioner have local absorbing boundary conditions imposed on the aperture.	76

Table 20	Total number of matrix vector products for the IDR(4)–ML-AMG algorithm to solve a small cavity model problem. M_1 is chosen as preconditioner in this experiment.	79
Table 21	Total number of matrix vector products for the IDR(4)–ML-AMG algorithm to solve a small cavity model problem. M_2 is chosen as preconditioner in this experiment.	79
Table 22	Total number of matrix vector products for the IDR(4)–ML-AMG algorithm for a cavity model problem of intermediate size. The model problem has dimensions $4\lambda \times 1.5\lambda \times 1.5\lambda$ and $N = 79,428$.	82
Table 23	List of quantities with their units and SI units.	91
Table 24	Two dimensional Poisson equation: varying N .	100
Table 25	Two dimensional Poisson equation: varying the number of multigrid W-cycles.	100
Table 26	Two dimensional Poisson equation: varying the cycle type.	100
Table 27	Two dimensional Poisson equation: varying the number of pre and post smoothing steps.	101
Table 28	Two dimensional Helmholtz equation: varying wavenumber, constant total number of unknowns $N = 101^2$ and using M_1 with optimal real shift $(\beta_1, \beta_2) = (-1, 0)$.	102
Table 29	Two dimensional Helmholtz equation: constant wavenumber $k_0 = 30$ and varying N , using M_1 with optimal real shift $(\beta_1, \beta_2) = (-1, 0)$.	102
Table 30	Two dimensional Helmholtz equation: varying wavenumber, constant total number of unknowns $N = 101^2$ and using M_2 with optimal real shift $(\beta_1, \beta_2) = (-1, 0)$.	102
Table 31	Two dimensional Helmholtz equation: constant wavenumber $k_0 = 30$ and varying N , using M_2 with optimal real shift $(\beta_1, \beta_2) = (-1, 0)$.	102
Table 32	Two dimensional Helmholtz equation: varying wavenumber, constant total number of unknowns $N = 101^2$ and using M_1 with optimal complex shift $(\beta_1, \beta_2) = (1, -0.5)$.	103
Table 33	Two dimensional Helmholtz equation: constant wavenumber $k_0 = 30$ and varying N , using M_1 with optimal complex shift $(\beta_1, \beta_2) = (1, -0.5)$.	103
Table 34	Two dimensional Helmholtz equation: varying wavenumber, constant total number of unknowns $N = 101^2$ and using M_2 with optimal complex shift $(\beta_1, \beta_2) = (1, -0.5)$.	103
Table 35	Two dimensional Helmholtz equation: constant wavenumber $k_0 = 30$ and varying N , using M_2 with optimal complex shift $(\beta_1, \beta_2) = (1, -0.5)$.	104



Preface

This is the Master thesis for the degree of Master of Science in Applied Mathematics at the faculty of Electrical Engineering, Mathematics and Computer Science of Delft University of Technology. The project has a duration of nine months and the graduation is performed in the Numerical Analysis group.

The Master thesis is performed at the National Aerospace Laboratory (NLR), located in Amsterdam. This institute is the key center of knowledge and experience for aerospace technology in the Netherlands. The main subject is the optimization of the solution procedure of a very large system of complex valued linear equations, which result from the discretization of the vector wave equation by the finite element discretization method.

This thesis will be supervised on location by dr. Harmen van der Ven and in Delft by dr. ir. Duncan van der Heul and prof. dr. ir. C. Vuik.

The examination committee for this master thesis consists of:

Prof. dr. ir. C. Vuik

(Delft University of Technology)

Dr. H van der Ven

(National Aerospace Laboratory)

Dr. ir. H.X. Lin

(Delft University of Technology)

Dr. Ir. D. van der Heul

(Delft University of Technology)

Amsterdam, March 6, 2009

Shiraz Abdoel

shiraz@ch.tudelft.nl

1 Introduction

RADAR (**R**adio **D**etection and **R**anging) is technology that can be used to detect aircraft and ships by using electromagnetic waves. As it is important to predict this detectability of platforms in the development stage, theoretical radar signature predicting techniques must be used. A measure to quantify the radar signature is the so called *radar cross section* (RCS).

It is known that the jet engine air intake of a fighter aircraft, a forward facing cavity, accounts for the major part of the RCS for a large angular region, when excited from the *front side*. The electric field scattered by the jet engine air intake can be computed by solving the *vector wave equation* obtained from Maxwell's equation(s) with the appropriate boundary conditions. This equation is discretized using a *finite element discretization method* resulting in a large system of linear equations. The system matrix is complex valued with a sparsely and a fully populated part. In the present implementation the discretized system is solved using a nested Krylov method. More specifically, this system is preconditioned by the *shifted Laplace preconditioner* and the so called *preconditioner system* is solved using the *generalized conjugate residual* (GCR) method. As GCR is a so called *long recurrence method*, the greatest disadvantage of this method is the storage requirement it imposes.

The purpose of this thesis is to investigate alternative solution methods for the preconditioner system, namely a solution method based on *multigrid*. Multigrid (MG) can be classified as geometric or algebraic, depending on the structuredness of the underlying grid. When the grid is unstructured or irregular, *algebraic multigrid* (AMG) can be used. In this case, the coefficients in the system matrix will be used to specify the coarse grid operator, the prolongation and restriction operator, etc., without any information about the computational grid.

Another favorable consequence of using multigrid for the preconditioner solve, is that it leads to a *constant preconditioner system*. Retaining a constant preconditioner matrix throughout the solution sequence, means that the system can be solved by a short recurrence method e.g. *stabilized Bi-Conjugate Gradient* (Bi-CGSTAB) method or IDR(4)¹, instead of the currently used GCR algorithm².

¹See Chapter 5

²Note that in the current implementation, GCR is used to solve the preconditioned and the preconditioner system

Since the current project does not allow for the development of an algebraic multigrid method from scratch and because there are several AMG black box methods freely available, the objective was to choose a black box solver that seemed most suitable to incorporate in the existing algorithm. According to Maclachlan and Oosterlee (ref. 14), the multigrid package with this property was Sandia's Multilevel (ML) preconditioning package (ref. 7).

This report is outlined as follows. In Chapter 2 the governing equations will be discussed, together with the finite element discretization method and the resulting linear system. The choice of elements and basis functions is explained and some important properties of the system matrix are stated.

Chapter 3 considers an iterative solution method of the discretized vector wave equation using the multilevel algebraic multigrid algorithm (ML-AMG) from Sandia's laboratories. The present implementation is explained together with the structure of the preconditioner and preconditioned system. The ML-AMG algorithm is applied to a small cavity problem and it is concluded that multigrid is a very efficient solver for the preconditioner system. Unfortunately, the convergence behaviour of the Krylov method is still unsatisfactory compared to the algorithm of Erlangga (ref. 6).

In Chapter 4 the differences between the original algorithm of Erlangga (ref. 6) and the proposed algorithm are analyzed. It turns out that the approximation of the discrete shifted Helmholtz operator by its block upper triangular part, leads to a preconditioned system that still has an unfavorable spectrum.

It also turns out that changing the Krylov method from Bi-CGSTAB to IDR(4), dramatically improves the convergence. This is the subject of Chapter 5.

In Chapter 6 the influence of all the distinguishing features identified in Chapter 4 is analyzed. Finally, Chapter 7 illustrates the improvements made in the algorithm for a cavity of intermediate size, followed by conclusions and recommendations for future research.

2 An algorithm for full wave analysis of cavity scattering

2.1 Introduction

In this chapter a numerical method for the analysis of cavity scattering is described, based on a finite element discretization of the Maxwell equations. The focus of this project is on the efficient solution of the linear system that results after discretization of the Maxwell equations.

In Section 2.2 the Maxwell equations are introduced after which the dimensionless form is presented in Subsection 2.2.2. The application of the FEM using zeroth order basis functions, unfortunately leads to a large number of unknowns for a large scatterer and has a low convergence rate. To overcome these problems, higher order basis functions can be used. The finite element method, using higher order basis functions to discretize the system, is the subject of Section 2.3. This chapter will be completed with some properties of the resulting linear system in Section 2.4.

2.2 Physical model

In the introduction of this thesis it is already mentioned that for forward observation angles, the field scattered by jet engine air intake of a modern fighter aircraft accounts for the main part of the total scattered field for an *electromagnetic* wave that excites the platform. This air intake is a deep open cavity and is characterized by a large **Length/diameter** ratio: $\frac{L}{d} > 3$. Because of this large ratio, it is not possible to use high frequency asymptotic methods to approximate the solution and therefore full wave methods will be used (see Van der Heul, Van der Ven and Van der Burg, ref. 4).

2.2.1 Maxwell equations

In 1873, James Clerk Maxwell coupled the work of several scientists, covering the equations of electromagnetism. Below they are stated for a general domain Ω in differential form¹:

$$\nabla^* \times \mathcal{E}^* = -\frac{\partial^* \mathcal{B}^*}{\partial^* t^*}, \quad (2.2.1)$$

$$\nabla^* \times \mathcal{H}^* = \frac{\partial^* \mathcal{D}^*}{\partial^* t^*} + \mathcal{J}^*, \quad (2.2.2)$$

$$\nabla^* \cdot \mathcal{D}^* = \mathcal{Q}^*, \quad (2.2.3)$$

$$\nabla^* \cdot \mathcal{B}^* = 0, \quad (2.2.4)$$

$$\nabla^* \cdot \mathcal{J}^* = -\frac{\partial^* \mathcal{Q}^*}{\partial^* t^*}. \quad (2.2.5)$$

¹In this thesis dimensionfull variables are denoted with a *.

Here the following variables are used with their S.I. unit between brackets:

$$\begin{aligned}\mathcal{E}^* &= \text{electric field intensity } \left[\frac{\text{V}}{\text{m}}\right], \\ \mathcal{D}^* &= \text{electric flux density } \left[\frac{\text{C}}{\text{m}^2}\right], \\ \mathcal{H}^* &= \text{magnetic field intensity } \left[\frac{\text{A}}{\text{m}}\right], \\ \mathcal{B}^* &= \text{magnetic flux density } \left[\frac{\text{Wb}}{\text{m}^2}\right], \\ \mathcal{J}^* &= \text{electric current density } \left[\frac{\text{A}}{\text{m}^2}\right], \\ \mathcal{Q}^* &= \text{electric charge density } \left[\frac{\text{C}}{\text{m}^3}\right], \\ t^* &= \text{time } [\text{s}].\end{aligned}$$

From this point on, the assumption is made that the field quantities above are harmonic oscillating functions with an angular frequency ω^* , so called *time-harmonic functions*. ω^* is defined as $\omega^* = 2\pi f^*$, with f^* the frequency measured in hertz (Hz). In the current application discussed in this thesis, $f^* = 10$ GHz.

Let $\mathcal{F}^*(\mathbf{x}^*, t^*)$ denote a time-harmonic function denoted by:

$$\mathcal{F}^*(\mathbf{x}^*, t^*) = \mathbf{F}^*(\mathbf{x}^*)e^{j\omega^*t^*}, \quad (2.2.6)$$

with $j^2 = -1$. Then the derivative of \mathcal{F}^* with respect to time t^* becomes:

$$\frac{\partial \mathcal{F}^*(\mathbf{x}^*, t^*)}{\partial t^*} = j\omega^* \mathcal{F}^*(\mathbf{x}^*, t^*). \quad (2.2.7)$$

Under this assumption for the other variables above, the general equations (2.2.1), (2.2.2) and (2.2.5) above are rewritten as:

$$\nabla^* \times \mathbf{E}^* = j\omega^* \mathbf{B}^*, \quad (2.2.8)$$

$$\nabla^* \times \mathbf{H}^* = j\omega^* \mathbf{D}^* + \mathbf{J}^*, \quad (2.2.9)$$

$$\nabla^* \cdot \mathbf{J}^* = -j\omega^* q^*, \quad (2.2.10)$$

where the quantities \mathbf{E}^* , \mathbf{D}^* , \mathbf{B}^* , \mathbf{H}^* , \mathbf{J}^* , and q^* are the *phasor* quantities corresponding to the variables defined before.

Additional relations are needed to close the problem. These so called *constitutive relations* describe the macroscopic properties of the medium of interest. They are given by:

$$\mathbf{D}^* = \varepsilon^*(\mathbf{x}^*)\mathbf{E}^*, \quad (2.2.11)$$

$$\mathbf{B}^* = \mu^*(\mathbf{x}^*)\mathbf{H}^*, \quad (2.2.12)$$

$$\mathbf{J}^* = \sigma^*(\mathbf{x}^*)\mathbf{E}^*, \quad (2.2.13)$$

where the parameters ε^* , μ^* and σ^* represent:

- $\varepsilon^* = \varepsilon^*(\mathbf{x}^*)$ the permittivity: $[\frac{\text{farads}}{\text{meter}}]$,
- $\mu^* = \mu^*(\mathbf{x}^*)$ the permeability: $[\frac{\text{henrys}}{\text{meter}}]$,
- $\sigma^* = \sigma^*(\mathbf{x}^*)$ the conductivity: $[\frac{\text{siemens}}{\text{meter}}]$.

These parameters are written as a product of the vacuum value ε_0^* and a relatively constant ε_r . So, e.g. $\varepsilon^* = \varepsilon_0^* \varepsilon_r$. For simple problems, these parameters are constant. For so called *radar absorbing materials*, the permittivity and permeability can be complex valued. See appendix A for the vacuum values and the relations to the standard SI-units.

When equations (2.2.8), (2.2.9), (2.2.11) and (2.2.12) are combined, the vector wave equation in the presence of a source $\mathbf{J}^* \neq 0$ can be derived:

$$\nabla^* \times \left(\frac{1}{\mu^*} \nabla^* \times \mathbf{E}^* \right) - \omega^{*2} \varepsilon^* \mathbf{E}^* = -j\omega^* \mathbf{J}^*. \quad (2.2.14)$$

When the following two definitions are used:

1. free-space wavenumber $k_0^* := \omega^* \sqrt{\varepsilon_0^* \mu_0^*}$ and
2. free-space impedance $Z_0^* := \sqrt{\frac{\mu_0^*}{\varepsilon_0^*}}$,

Equation (2.2.14) can be rewritten as:

$$\nabla^* \times \left(\frac{1}{\mu_r^*} \nabla^* \times \mathbf{E}^* \right) - k_0^{*2} \varepsilon_r \mathbf{E}^* = -jk_0^* Z_0^* \mathbf{J}^*. \quad (2.2.15)$$

If there is no source i.e. $\mathbf{J}^* = 0$, as is the case inside the cavity, the vector wave equation is called *homogeneous*.

To define a well posed boundary value problem, appropriate boundary conditions have to be imposed. Therefore, it is necessary to define either the tangential electric field or the tangential magnetic field on the boundary of the domain (see Balanis, ref. 3). The boundary of the cavity consists of the aperture (S_{aperture}) and the mantle (S_{mantle}) of the cavity (see Figure 2.2.1). Firstly, the boundary conditions are stated in equations (2.2.16) and (2.2.17) and afterward, this section will be ended with some notational issues:

$$(\hat{n} \times \mathbf{E}^*)_{S_{\text{mantle}}} = 0, \quad (2.2.16)$$

$$(\hat{n} \times \mathbf{H}_{\text{inc}}^*)_{S_{\text{aperture}}} = 4\hat{n} \times \left\{ \frac{\nabla^{*2} \cdot \mathbf{N}^* + k_0^{*2} \mathbf{N}^*}{j\omega^* \mu_0^*} \right\}, \quad (2.2.17)$$

where:

1. the quantity $\mathbf{K}^*(\mathbf{r}) = \hat{n} \times \mathbf{E}^*(\mathbf{r})$ is a fictitious magnetic current,
2. $\mathbf{H}_{inc}^*(\mathbf{r})$ denotes the incident magnetic field,
3. $\mathbf{N}^*(\mathbf{r}) = \mathbf{K}^*(\mathbf{r}) * G(r, r') = \int \int_{S_{aperture}} \mathbf{K}^*(\mathbf{r}') G(\mathbf{r}, \mathbf{r}') d\mathbf{r}' = \int \int_{S_{aperture}} [\hat{n} \times \mathbf{E}^*(\mathbf{r}') d\mathbf{r}']$,
4. $G(r, r')$ is the three dimensional Green's function:

$$G(r, r') = \frac{e^{-jk|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r}-\mathbf{r}'|},$$
5. '*' denotes the three-dimensional convolution.

In the formulation of boundary condition (2.2.17) it is assumed that the aperture of the cavity is surrounded by an infinite groundplane. Through the presence of the groundplane, the electric current distribution on the aperture vanishes. In the next subsection the procedure to make the vector wave equation dimensionless will be discussed.

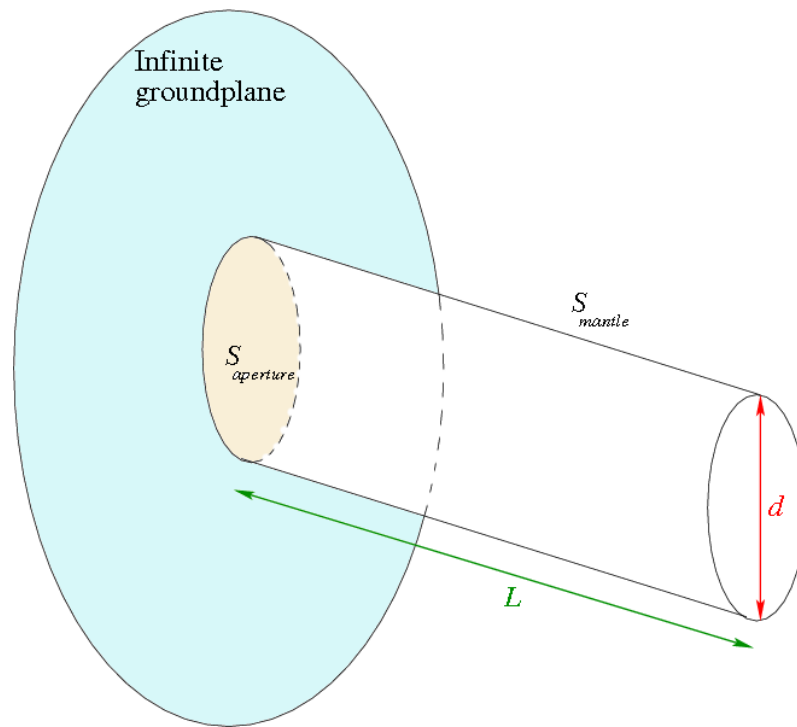


Fig. 2.2.1 Schematic view of cylindrical cavity with length L and cross section diameter d .

2.2.2 Dimensional analysis

Dimensional analysis is a conceptual tool that can be used to reduce the number of parameters in a given system of equations. In this report, dimensional analysis will be applied to the vector wave equation to determine which parameter(s) characterize the problem.

The important parameters which are made dimensionless are stated. There are four scale factors used to transform (\rightarrow) the variables, parameters and operator:

- length: R ,
- mass: M ,
- time: T and
- electric current: I .

Quantities:

$$\mathbf{E}^* : \text{electric field intensity} \rightarrow \mathbf{E} := \mathbf{E}^* \frac{T^3 I}{RM}$$

$$\mathbf{J}^* : \text{electric current density} \rightarrow \mathbf{J} := \mathbf{J}^* \frac{R^2}{I}$$

Position Variables:

$$\text{Define } x, y, z = \frac{x^*}{R}, \frac{y^*}{R}, \frac{z^*}{R} \text{ respectively.}$$

Parameters:

- free space wavenumber $k_0^* \rightarrow k_0 = k_0^* R$.
- intrinsic free space impedance $Z_0^* \rightarrow Z_0 := Z_0^* \frac{I^2 T^3}{MR^2}$.

Operator:

$$\text{gradient } \nabla^* \rightarrow \frac{1}{R} \nabla.$$

Using these definitions, the vector wave equation can be rewritten in the following *dimensionless* form:

$$\nabla \times \nabla \times \mathbf{E} - k_0^2 \varepsilon_r \mathbf{E} = -jk_0 Z_0 \mathbf{J}. \quad (2.2.18)$$

From this dimensionless form it is clear that the most important parameter in the left-hand side of this equation is the dimensionless wavenumber k_0 . It can be shown that a (deep) cavity has two characteristic lengths. One is the depth and the second one is the diameter d . It turns out that the diameter is the most important characteristic length scale: the electric field inside the cavity is directly related to the electric field modes that exist inside the cross section. Therefore, k_0 is defined as $k_0 := dk_0^*$. The next subsection considers the dependency of the so called *radar cross section* on the dimensionless wavenumber.

2.2.3 Dependence of RCS on the dimensionless wavenumber

According to Knott et al. (ref. 5), the RCS of a scattering body has a strong relationship with the non-dimensional wavenumber k_0 (scaled by a characteristic length L). The following classification for the RCS, depending on the value of k_0 , is made:

1. Rayleigh region: $0.1 < k_0 < 1$.

In this region the geometry is not a very important parameter. Only the characteristic dimensions of the object are of importance.

2. Resonance region: $1 < k_0 < 10$.

In this region the geometry has an important role in the interaction between the fields scattered by different components of the body.

3. Optics region: $10 < k_0 < 100$.

In this region there is almost no interaction between different components of the scattering body.

See Figure 2.2.2 for an illustration.

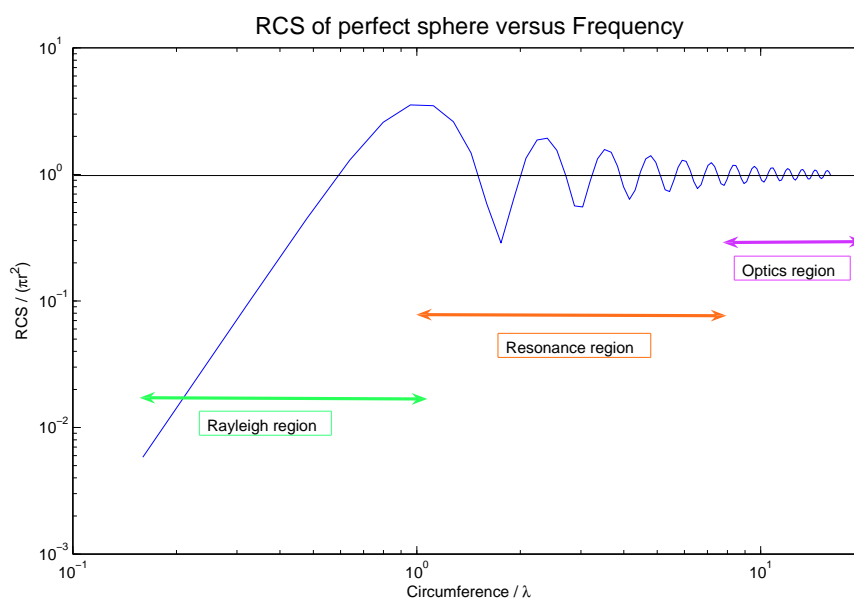


Fig. 2.2.2 The RCS σ of a metallic sphere with radius a illustrates the three scattering regions.

From a computational point of view, the following discussion about a comparison in the dimensionless wavenumber is included. Suppose that the scattering by a certain body is analyzed for two dimensionless wavenumbers k_1 and k_2 corresponding to two different frequencies of the incident electromagnetic waves. Furthermore, assume that $k_1 > k_2$.

It is worth mentioning that in this case with $k_1 > k_2$, there are two important things to note:

1. For equal accuracy of the solution, the total number of unknowns N required for discretization in case k_1 will increase compared to case k_2 with factor $\left(\frac{k_1}{k_2}\right)^3$.
2. There is a negative effect on the indefiniteness² of the system matrix. See Subsection 3.2.1 for more about this subject.

Indefinite matrices are not favorable because:

- The property of indefiniteness has a negative effect on the convergence rate of iterative methods used to solve a system of linear equations.
- Standard multigrid methods do not converge (see Chapter 4 of Abdoel, ref. 1).

The subject of the next section is the finite element discretization method.

2.3 Finite element discretization method

From the ongoing research on computational electromagnetics it is known that the following numerical methods are the most popular to solve electromagnetic scattering problems: the method of moments (MOM)³, the finite difference method (FDM) and the finite element discretization method (FEM)⁴. In problems with inhomogeneous materials, the MOM has the disadvantage that the computational complexity increases rapidly because of the usage of a volume formulation, rather than a surface formulation and a full matrix structure is the result. The application of a FDM however, results in a sparse system which is computationally efficient. The major drawback of FDM's is that they rely on rectangular grids. The FEM can remove all of these difficulties associated with the MOM and the FDM. Another great favorable point associated with the FEM is that it can be used in problems where discontinuous coefficients are involved. In computational electromagnetics, these problems will occur in the case of discontinuities in the material properties (permittivity and permeability). To handle these discontinuities, the so called *weak formulation* is used.

In order to combine efficiency and accuracy in the FEM, higher order *vector* basis functions will be used. Important in higher order methods are the higher order approximations of the geometry and the higher order representation of the unknown field quantities. For the FEM these unknown quantities can be the electric or magnetic field as seen in the Maxwell equations.

To avoid the occurrence of spurious solutions and to ensure that the numerical solution obeys the correct interface combinations at material interfaces, edge elements or so called *Nedelec* elements are used.

²See for the definition of definiteness Section 2.4.

³See Chapter 14 of Jin (ref. 13).

⁴See Jin (ref. 13).

In the context of the FEM used here, the higher order vector basis functions proposed by Graglia, Wilton and Peterson (ref. 16) are used. The following setup is stated:

- Consider a curvilinear tetrahedral element in the xyz -space⁵.
- Tetrahedra can be mapped to a rectilinear element in the ξ -space. The mapping is given by:

$$\mathbf{r} = \sum_{j=1}^{10} \varphi_j(\xi_1, \xi_2, \xi_3, \xi_4) \mathbf{r}_j.$$

- The shape functions φ_j are defined in terms of the parametric coordinates $\xi_1, \xi_2, \xi_3, \xi_4$ as:

$$\begin{aligned} \varphi_1 &= \xi_1(2\xi_1 - 1) & \varphi_2 &= \xi_2(2\xi_2 - 1) & \varphi_3 &= \xi_3(2\xi_3 - 1) \\ \varphi_4 &= \xi_4(2\xi_4 - 1) & \varphi_5 &= 4\xi_1\xi_2 & \varphi_6 &= 4\xi_1\xi_3 \\ \varphi_7 &= 4\xi_1\xi_4 & \varphi_8 &= 4\xi_2\xi_3 & \varphi_9 &= 4\xi_3\xi_4 \\ & & \varphi_{10} &= 4\xi_2\xi_4 & & \end{aligned}$$

Note that $\xi_1 + \xi_2 + \xi_3 + \xi_4 = 1$; $\boldsymbol{\xi} = (\xi_1, \xi_2, \xi_3, \xi_4)$.

- The i -th face of the dimensional tetrahedra is the zero-coordinate surface for the normalized coordinate ξ_i .
- The edges of the faces of the tetrahedra must be consistently numbered for successful implementation (see edge definition in Table 1 and Figure 2.3.1).
- The four nodes of the tetrahedra are labeled as (γ, β, m, n) and the face *opposite* node γ is called γ as well.
- Normalized coordinate ξ_i varies linearly across the element attaining the value 1 at the face opposite the zero-coordinate surface (e.g.: ξ_m or ξ_n has value 1 at node m or n and 0 on face m or n).
- An independent set of three coordinates is selected and indexed in a “right-handed” sense such that $\nabla\xi_3 \cdot (\nabla\xi_1 \times \nabla\xi_2)$ is strictly positive.
- The vector basis function associated with the edge shared by faces γ and β is given by:

$$\mathbf{N}_{\gamma\beta}(\mathbf{r}) = \xi_n \nabla \xi_m - \xi_m \nabla \xi_n. \quad (2.3.1)$$

- It can be shown that the basis functions $\mathbf{N}_{\gamma\beta}$ have tangential components only on faces γ and β and they guarantee the continuity of the tangential field, while allowing the normal component of the field to be discontinuous, as occurs at the interface between two media with different permeability.

In the literature, the vector basis functions defined above are *referred* to as zeroth order basis functions, but obviously that would imply that no grid convergence ($O(1)$) would be achieved.

⁵Tetrahedral elements are the natural (simpler) extension of triangular elements in two dimensions.

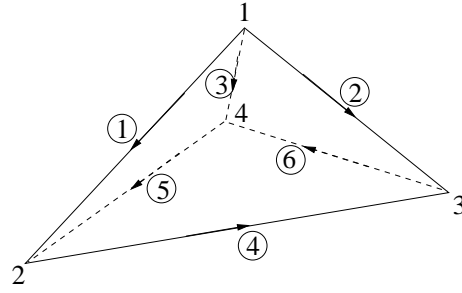


Fig. 2.3.1 The ordering of Table 1 is used here to number the edges of the tetrahedron.

Edge i	Node n_1	Node n_2
1	1	2
2	1	3
3	1	4
4	2	3
5	4	2
6	3	4

Table 1 Edge definition for a tetrahedral element.

These zeroth order basis functions are used to define the higher order interpolatory vector basis functions as follows. The zeroth order basis functions $\mathbf{N}_{\gamma\beta}$ are multiplied by a set of interpolatory polynomial functions, which are complete to specified order, say p .

In this setup, the polynomials of Silvester are used:

$$R_i(p, \xi) = \begin{cases} \frac{1}{i!} \prod_{k=0}^{i-1} (p\xi - k), & 1 \leq i \leq p, \\ 1, & i = 0. \end{cases} \quad (2.3.2)$$

Using these Silvester polynomials to define the *shifted* Silvester polynomials results in:

$$\hat{R}_i(p, \xi) = R_{i-1}\left(p, \xi - \frac{1}{p}\right). \quad (2.3.3)$$

These polynomials are used to effect scalar Lagrangian interpolation on the canonical elements as follows:

$$\hat{\alpha}_{ijkl}(\boldsymbol{\xi}) = \hat{R}_i(p+2, \xi_1) \hat{R}_j(p+2, \xi_2) \hat{R}_k(p+2, \xi_3) \hat{R}_l(p+2, \xi_4). \quad (2.3.4)$$

To define the higher order vector basis functions, the following definitions are made:

- the value $\ell_{\gamma\beta}^{(ijkl)} = |\ell_{\gamma\beta}|$ at the interpolation point

$$\boldsymbol{\xi}_{(ijkl)}^{\gamma\beta} = \left(\frac{i}{p+2}, \frac{j}{p+2}, \frac{k}{p+2}, \frac{l}{p+2} \right),$$

with $i + j + k + l = p + 2$.

- the normalization factor $K_{ijkl}^{\gamma\beta}$ defined as⁶:

$$K_{ijkl}^{\gamma\beta} = \frac{p+2}{p+2-i_\gamma-i_\beta} \ell_{\gamma\beta}^{(ijkl)},$$

where $i_\gamma \in \{i, j, k, l\}$, $\gamma \in \{1, 2, 3, 4\}$ and similarly for i_β .

Using all these preparations the higher order interpolatory vector basis functions are given by:

$$\mathbf{N}_{ijkl}^{\gamma\beta}(\mathbf{r}) = K_{ijkl}^{\gamma\beta} \frac{(p+2)^2 \xi_\gamma \xi_\beta \hat{\alpha}_{ijkl}(\boldsymbol{\xi})}{i_\gamma i_\beta} \mathbf{N}_{\gamma\beta}(\mathbf{r}) \quad (2.3.5)$$

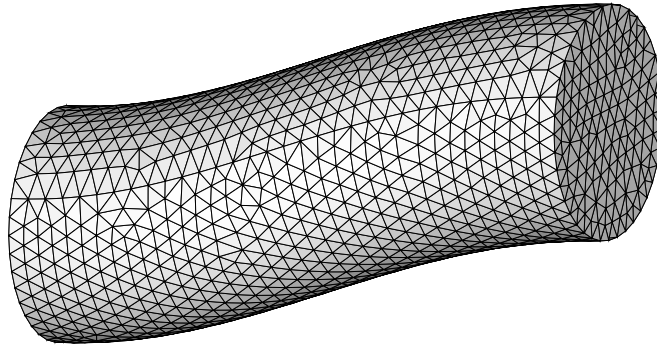


Fig. 2.3.2 An example of the discretization of the interior of an S-shaped cavity using tetrahedral elements.

This subsection will be concluded with some remarks about the *number of degrees of freedom* for the basis functions of order p on a tetrahedron (= number of basis functions needed). Equation (2.3.5) provides one basis function for an interpolation node on an edge of the tetrahedron. As a face has three edges, one face is associated with three basis functions. However, the tangential field on a face is spanned by two independent basis functions and therefore one of the three basis functions associated with a face must be discarded. Taking a closer look at an interior interpolation node, results in six basis functions among which there are obviously only three independent ones. Adding all basis functions results in $\frac{1}{2}(p+1)(p+3)(p+4)$ basis functions/degrees of

⁶The ranges of γ and β are such as to include the six zeroth order basis functions in (2.3.1) i.e. $\gamma < \beta$.

freedom. In the current application $p = 2$ resulting in 45 basis functions, needed for the second order tetrahedral element.

A final remark about the elements used in the current application. In the setup proposed by Graglia et al. (ref. 16), curvilinear elements are considered. In the current application however, rectangular elements are used, to improve the efficiency of the algorithm.

In the next subsection the formulation of the linear system is considered.

2.3.1 Formulation of the linear system

In this subsection it is explained how to obtain the linear system which must be solved. In applying the FEM, two things must be realized. First, inside the cavity volume the space is divided into small elements; in the current application tetrahedral elements are used. The surface field is discretized using compatible triangular elements. For the second order discretization used here, this leads to the following expansion of the electric field inside the volume elements (e) and one on the surface elements (s):

$$\mathbf{E}^e(\mathbf{x}) = \sum_{i=1}^{45} E_i^e \mathbf{N}_i^e(\mathbf{x}) = \{E^e\}^T \{\mathbf{N}^e(\mathbf{x})\}, \quad (2.3.6)$$

$$\hat{z} \times \mathbf{E}^s(\mathbf{x}) = \sum_{k=1}^{15} E_k^e \mathbf{S}_k^s(\mathbf{x}) = \{E^e\}^T \{\mathbf{S}^e(\mathbf{x})\}, \quad (2.3.7)$$

where $\mathbf{S}_k^e = \hat{z} \times \mathbf{N}_i^e$ is a compatible expansion.

Substituting⁷ these relations in the functional and applying Ritz's method, results in the following *functional*:

$$F = \frac{1}{2} \sum_{e=1}^M \{E^e\}^T [K^e] \{E^e\} + \frac{1}{2} \sum_{s=1}^{M_s} \sum_{t=1}^{M_s} \{E^s\} [P^{st}] \{E^t\} - \sum_{s=1}^{M_s} \{E^s\}^T \{b^s\}. \quad (2.3.8)$$

⁷Note that $\{\cdot\}$ is used to denote a vector with elements that are a vector itself.

Here the following is used:

★ M = total number of volume elements in the cavity.

★ M_s = total number of surface elements on the mantle.

$$\star \text{Matrix } [K^e] = \iiint_{V^e} \left[\frac{1}{\mu_r} \{\nabla \times \mathbf{N}^e\} \cdot \{\nabla \times \mathbf{N}^e\} - k_0^2 \varepsilon_r \{\mathbf{N}^e\} \cdot \{\mathbf{N}^e\} \right] dV. \quad (2.3.9)$$

$$\star \{b^s\} = -2jk_0 Z_0 \iint_{S^s} \{\mathbf{S}^s \cdot \mathbf{H}^{inc}\} dS. \quad (2.3.10)$$

★ Matrix $[P^{st}]$ is obtained from the boundary integral and is defined as

$$[P^{st}] = 2 \iint_{S^s} \{\nabla \cdot \mathbf{S}^s\} \left\{ \iint_{S^t} \{\nabla' \cdot \mathbf{S}^t\}^T G_0 dS' \right\} dS - 2k_0^2 \iint_{S^s} \{\mathbf{S}^s\} \cdot \left\{ \iint_{S^t} \{\mathbf{S}^t\}^T G_0 dS' \right\} dS. \quad (2.3.11)$$

The integrals $[K^e]$ and $\{b^e\}$ are computed numerically by Gauss' Quadrature formulas and for the matrix $[P^{st}]$ Duffy's method must be used to handle the singularity in the Green's function. The next subsection deals with the accuracy of the computed RCS pattern by considering the so called *dispersion error*:

2.3.2 Resolution of the field

It can be shown that the accuracy of the computed RCS pattern is dominated by the *dispersion error* ε in the electric field on the aperture, given by⁸:

$$\tilde{\psi}_{out} = \psi_{out} + \varepsilon.$$

Here ψ_{out} denotes the exact phase difference after reflection through the cavity and $\tilde{\psi}_{out}$ the computed phase difference which differs from the exact one. It is very important to note the possibility of waves to fortify or to partially cancel each other. In both cases the result is a specific distribution of maximal and minimal values of the radar cross section. In the case that waves fortify each other, the maximal value will be different compared to the case that waves nearly cancel each other. When the dispersion error is high, the interference will be predicted incorrectly and hence the accuracy of the computed RCS pattern will be poor.

In the following, the influence of the dispersion error is illustrated schematically in the following way (Figure 2.3.3). In the left picture a wave front enters the cavity with incidence angle ϕ . Two waves (red and blue) with initial phase difference $\psi_{in} = \frac{\lambda}{4}$ are followed. After reflection through the cavity there is an accumulated phase error ε . In the middle picture the exact phase

⁸For a more detailed analysis of this subject, the reader is referred to Hooghiemstra (ref. 9, Chapter 6).

difference ψ_{out} is depicted and in the right figure the computed phase difference $\tilde{\psi}_{out}$. Also note the difference in the maximal and minimal values.

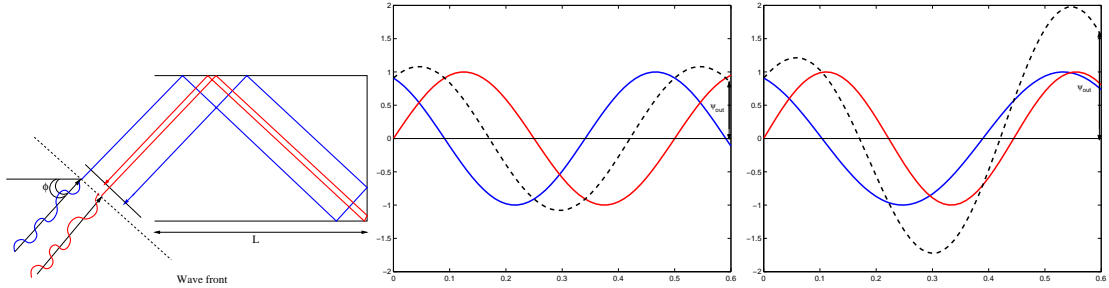


Fig. 2.3.3 **Left:** The wave front enters the cavity with incidence angle ϕ . Two waves with initial phase difference $\psi_{in} = \frac{\lambda}{4}$ are followed. After reflection through the cavity there is an accumulated phase error ε – **Middle:** The exact phase difference ψ_{out} – **Right:** The computed phase difference $\tilde{\psi}_{out}$.

The challenge here is to minimize the dispersion error because in the current application there is a deep cavity: $L \gg d$. In this case the dispersion error accumulates and leads to inaccurate results. One way to achieve this has already been discussed, namely using higher order elements. To get an idea of the total number of unknowns needed for a specified dispersion error, the following outline is given.

As already seen in Subsection 2.2.2, the dimensionless wavenumber k_0 is very important. When the scattering object size and the radar frequency f are known, it holds that:

$$\lambda = \frac{2d\pi}{k_0}.$$

Here d denotes the diameter of the geometrical cross section of the cavity. According to Jin et al (ref. 11), the maximum phase error *per wavelength* is an important quantity in this analysis. It is defined as:

$$\delta_p = \left(\frac{\lambda}{h^*} \right)^{-2(p+\alpha)},$$

where:

- h denotes the mesh size and p the order of the basis functions used,
- $h^* := \frac{h}{p+2}$ denotes the actual spacing of the unknowns,
- $\frac{\lambda}{h^*}$ is the number of unknowns per wavelength and
- $\alpha \in [1, 2]$ is the structuredness of the grid ($\alpha \approx 1$ for a structured mesh).

According to Hooghiemstra (ref. 9) the number of elements per wavelength required to achieve a

accumulated dispersion error ε is:

$$D(p) = \frac{\lambda}{h} = \left(\frac{2L}{\varepsilon \lambda \cos \phi} \right)^{\frac{1}{2(p+\alpha)}} \frac{1}{p+2}.$$

Knowing λ and thus $D(p)$ gives h . Using h as input parameter for the mesh generator, results in the grid which can be used in the problem.

In the final section of this chapter some properties of the linear system are discussed.

2.4 Properties of the linear system

After applying all the operations described in the previous sections, the final discretized system can be written in the form: $Au = f$. In this section some properties of the matrix A from the current application will be listed and discussed:

- Matrix A consists of a sparse part and fully populated part. See Figure 2.4.1.

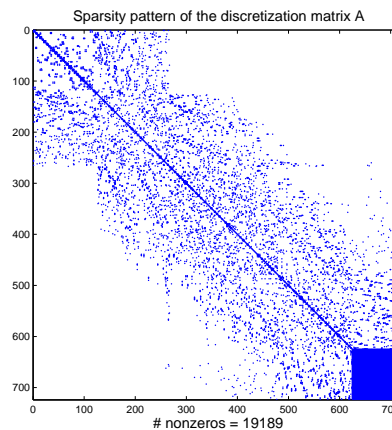


Fig. 2.4.1 $A \in \mathbb{C}^{N \times N}$, $N = 723$, $h = 0.25$. Dimensions rectangular cavity: $1.5\lambda \times 1.5\lambda \times 0.6\lambda$. Fully populated block has dimension 99. The total number of nonzeros is 19.189. The complex valued part of the matrix consists of the unknowns on the aperture only.

- Matrix A is 'nearly' symmetric, but not Hermitian. The sparse part is symmetric as it originates from the Galerkin FEM inside the cavity: the test function equals the basis function. The fully populated part however, is not completely symmetric because this part is the result of the discretization of the boundary integral in equation (2.3.11), in which the outer and inner integrals are evaluated differently when both are evaluated on the same triangular element. Although A is a complex valued matrix, it is not Hermitian (or self-adjoint). This reduces the choice of Krylov subspace methods that can be used.

- Matrix A is ill conditioned and hence the convergence of iterative methods is negatively affected. Ill conditioned means a very large *condition number* $\kappa(A)$. The condition number with respect to some norm $\|\cdot\|$ is defined as:

$$\kappa(A) = \|A\| \cdot \|A^{-1}\|.$$

Here $A \in \mathbb{C}^{N \times N}$ and A nonsingular. This definition depends on the choice of the norm.

- A matrix M is called *positive (semi)definite* if $\langle x, Mx \rangle > (\geq) 0$ and *negative (semi)definite* if $\langle x, Mx \rangle < (\leq) 0$. The consequence of this definition⁹ is that for symmetric or Hermitian matrices M , the real part of the eigenvalues of M are greater than (or equal to) zero or the real part of the- eigenvalues of M are less than (or equal to) zero.

The matrix A in the current application has eigenvalues with both positive and negative real part. A matrix with this property is said to be *indefinite*. The property of indefiniteness of the matrix limits the choice of and has a negative effect on the convergence of iterative solution methods that can be used to solve the linear system $Au = f$.

In the next chapter an iterative solution method for the linear system will be presented.

⁹ $\langle \cdot, \cdot \rangle$ denotes the inner product.

3 Iterative solution of the discretized vector wave equation

3.1 Introduction

In this chapter iterative solution methods for the linear system

$$Au = f \quad (3.1.1)$$

will be considered with $A \in \mathbb{C}^{N \times N}$ a square nonsingular matrix, $u, f \in \mathbb{C}^N$ and N the total number of unknowns. Here A results from the discretization of the vector wave equation with global radiation boundary conditions as described in Chapter 2. Section 3.2 starts with an outline of the present implementation (see Hooghiemstra, 10). The structure and the solution of the preconditioner system will be discussed and Section 3.3 deals with the incorporation of algebraic multigrid in the existing algorithm. This chapter will be concluded with some numerical results and conclusions based on the performed experiments.

3.2 Present implementation

The starting point for this thesis is the present implementation by Hooghiemstra (ref. 10). The solver is a nested preconditioned GCR-algorithm. A block upper triangular preconditioner is constructed from the blocks of the finite element discretization matrix A combined with the shifted Laplace preconditioner. The GCR method has long recurrences and therefore, an explicit orthonormal basis for the Krylov subspace has to be constructed in every iteration which can lead to unacceptable storage requirement, for large dimension N and large number of iterations. In the next subsection the structure of the preconditioner is considered.

3.2.1 Structure of the preconditioner

This subsection gives an overview of the structure of the preconditioner used by Hooghiemstra (ref. 10). The structure of the system matrix A is repeated here for convenience (recall Figure 2.4.1):

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad (3.2.1)$$

with $A_{11} \in \mathbb{C}^{M \times M}$, $A_{22} \in \mathbb{C}^{m \times m}$, $m \ll M$.

Note that A_{22} results from the discretization of the boundary conditions. For the prescribed global radiation conditions, this block is a full matrix. The other matrices stem from the inner region and are sparsely populated.

The system matrix has complex eigenvalues with both positive and negative real part. These complex eigenvalues are caused by the global absorbing boundary conditions imposed on the

boundary. When *Dirichlet* boundary conditions are imposed on the *whole* boundary and when real valued material properties and a pure real shift for the shifted Laplace preconditioner are considered, the eigenvalues of the system are real valued and may be positive or negative. In this case, the system matrix becomes indefinite. To get a more favorable spectrum, a (block)preconditioner matrix M is chosen. Hooghiemstra (ref. 10) used the following preconditioner matrix:

$$M = \begin{bmatrix} M_1 & A_{12} \\ 0 & A_{22} \end{bmatrix}. \quad (3.2.2)$$

Here M_1 results from the finite element discretization of the shifted Laplace operator in *vector form*:

$$\mathcal{W}_{(\beta_1, \beta_2)} := -\Delta - \hat{k}_0^2, \quad (3.2.3)$$

where \hat{k}_0 is the shifted wavenumber defined as: $\hat{k}_0 = (\beta_1 + \imath\beta_2)k_0$, $\beta_1, \beta_2 \in \mathbb{R}$ and $\imath^2 = -1$.

In this thesis another formulation of the shifted Laplace operator in vector form is proposed:

$$\mathcal{M}_{(\beta_1, \beta_2)} := -\Delta - (\beta_1 + \imath\beta_2)k_0^2, \quad \beta_1, \beta_2 \in \mathbb{R} \text{ and } \imath^2 = -1. \quad (3.2.4)$$

The discretization of the latter formulation is denoted by:

$$\tilde{A} = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{bmatrix}, \quad (3.2.5)$$

and the newly proposed preconditioner looks like:

$$M_{new} = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ 0 & \tilde{A}_{22} \end{bmatrix}. \quad (3.2.6)$$

Note that the preconditioner in Equation (3.2.2) combines a block upper triangular matrix with the shifted Laplace preconditioner. The system $Ms = r$ is referred to as *the preconditioner system* and is solved in two steps:

- first solve $A_{22}s_2^k = r_2^{k-1}$ with a precomputed LU-decomposition of A_{22} . Here s denotes a search direction and r denotes the residual. This LU-decomposition is cheap to perform because block A_{22} is fully populated and very small compared to the other blocks (see Equation (3.2.1)). Furthermore, this decomposition is computed only once.
- second solve: $M_1s_1^k = r_1^{k-1} - A_{12}s_2^k$.

Note that the matrix M_1 is a large sparse matrix and that the preconditioner system to be solved is a large linear system. Therefore, the next subsection starts with the solution procedure for the preconditioner system.

3.2.2 Solution of the preconditioner system

Hooghiemstra (ref. 10) used GCR as solution method for the preconditioner system

$$M_1 s_1^k = r_1^{k-1} A_{12} s_2^k. \quad (3.2.7)$$

See the resulting algorithm below:

Algorithm: Preconditioned generalized conjugate residual method

Start GCR-1 for solving $Ax = b$
 Compute initial residual $r^0 = b - Ax^0$ for some initial guess x^0
for $k = 1, 2, \dots, \max k$ **do**
 Solve $A_{22}s_2^k = r_2^{k-1}$ using a precomputed LU-decomposition for A_{22} :
 Solve $L_{22}w = r_2^{k-1}$ (forward substitution)
 Solve $U_{22}s_2^k = w$ (backward substitution)
 Start GCR-2 for solving $M_1 s_1^k = r_1^{k-1} - A_{12}s_2^k$
 Compute initial residual $\tilde{r}^0 = r_1^{k-1} - A_{12}s_2^k$ for initial guess \tilde{s}^0
 for $j = 1, 2, \dots, \max j$ **do**
 $\tilde{s}^j = \tilde{r}^{j-1}$
 $\tilde{v}^j = A_{11}\tilde{s}^j$
 call *orthormalize*($j, \tilde{v}^1, \dots, \tilde{v}^j, \tilde{s}^1, \dots, \tilde{s}^j$)
 $y^j = y^{j-1} + (\tilde{v}^j, \tilde{r}^{j-1})\tilde{s}^j$
 $\tilde{r}^j = \tilde{r}^{j-1} - (\tilde{v}^j, \tilde{r}^{j-1})\tilde{v}^j$
 end for
 $s_1^k = y^j$
 $s^k = (s_1^k, s_2^k)^T$
 $v^k = As^k$
 call *orthonormalize*($k, v^1, \dots, v^k, s^1, \dots, s^k$)
 $x^k = x^{k-1} + (v^k, r^{k-1})s^k$
 $r^k = r^{k-1} - (v^k, r^{k-1})v^k$
end for

A more elaborate discussion of the fine tuning of the shift parameters for this nested GCR algorithm, is given in Chapters 4 and 5 from Hooghiemstra (ref. 10). This approach was chosen in an exploratory study to evaluate the effectiveness of the shifted Laplace preconditioner for the vector wave equation.

It turns out that using GCR for the preconditioner solve leads to an inefficient solver due to the storage requirements. It is expected that using an *algebraic multigrid method* for the preconditioner



tioner solve, in the same way geometric multigrid was included in the robust and efficient algorithm of Erlangga (ref. 6), will significantly improve the efficiency of the existing solver. It is important to emphasize that in this thesis, algebraic multigrid will be used to *approximate* the inverse of the preconditioner using one multigrid cycle. So whenever the terms ‘preconditioner solve’ are used, one algebraic multigrid approximation of the inverse of the preconditioner matrix is meant. The next subsection considers the solution of the preconditioned system.

3.2.3 Solution of the preconditioned system

Recall the linear system $Ax = b$ to be solved and the preconditioner matrix M in Equation (3.2.2). The preconditioned system is then denoted by:

$$M^{-1}Ax = M^{-1}b. \quad (3.2.8)$$

In the present implementation the Krylov solver for this linear system is GCR. As already mentioned, GCR is a long recurrence method, hence the complete Krylov basis has to be stored during the entire solution process. Note that Hooghiemstra (ref. 10) also tried to combine his algorithm with the truncated and restarted GCR method, but the results are not favorable for these approaches. As the total number of unknowns in the current application is very large, the storage requirements for the Krylov basis are unacceptably high. Because the preconditioner solve is also performed by GCR, there is no constant preconditioner. Therefore, a long recurrence method has to be used to solve the preconditioned system.

When the preconditioner solve is performed by algebraic multigrid, the preconditioner will be constant. The preconditioned system can then be solved by a short recurrence method. It is expected that the Bi-CGSTAB method will be a good alternative for the current GCR algorithm. The incorporation of algebraic multigrid in the existing algorithm is the subject of Section 3.3.

3.3 Incorporation of algebraic multigrid in the existing algorithm

As already mentioned in the previous section, the idea is to incorporate multigrid in the existing algorithm in the same way as Erlangga (ref. 6) did to obtain his efficient algorithm. As there is no structured grid available, the classical geometric multigrid methods cannot be used. Therefore, an algebraic multigrid method (AMG) is chosen for incorporation in the current application. More specifically, Sandia’s Multilevel (ML) preconditioning package is chosen. Note that ML will only be used to perform the *preconditioner* solve. The next subsection considers the multilevel-AMG algorithm and Subsection 3.3.2 discusses the limitations due to the application of this multilevel-AMG algorithm.



3.3.1 The ML-AMG algorithm

The multilevel AMG (ML-AMG) package is one of Sandia's laboratories¹ main multigrid preconditioning packages. It is possible to generate a so called *matlab executable file* (`mlmex-file`) in order to easily incorporate algebraic multigrid in Matlab for testing purposes. Using this `mlmex-file` in the Matlab environment it is possible to test the multigrid performance, the effect of the different multigrid parameters and to easily perform the preconditioner solve. More details about ML are included in Appendix C.

3.3.2 Limitations of application of ML

The current implementation of the ML algorithm can only perform computations in real valued arithmetic. The current application however, has a complex valued system matrix and a complex valued right-hand side. The complex valued matrix results from the boundary conditions, a possible imaginary shift in the shifted Laplace preconditioner and the possible complex valued material properties (permittivity and permeability). The complex valued right-hand side is due to the boundary conditions.

When only *real valued* material properties (materials without damping) and a *real shift* ($\beta_1 \in \mathbb{R}, \beta_2 = 0$) are considered, the only complex valued entries in the system matrix A are due to the boundary conditions. These complex entries can be reordered such that they are grouped into block A_{22} . As can be seen in previous section, block A_{22} is used to form the right-hand side for the preconditioner solve, which will be performed by ML. Consider the following algorithm:

¹<http://trilinos.sandia.gov/>

Algorithm: Preconditioned Bi-orthogonal conjugate gradient stabilized method

Start Bi-CGSTAB for solving $Ax = b$

Compute initial residual $r^0 = b - Ax^0$ for some initial guess x^0

Choose \tilde{r}^0 such that $\langle \tilde{r}^0, r^0 \rangle \neq 0$, e.g. $\tilde{r}^0 = r^0$

Set $\rho^0 = \alpha = \omega = 1$

Set $v^0 = p^0 = 0$

for $k = 1, 2, \dots, \max k$ **do**

$$\rho^k = \langle \tilde{r}^0, r^{k-1} \rangle$$

$$\beta = \frac{\rho^k \alpha}{\rho^{k-1} \omega^{k-1}}$$

$$p^k = r^{k-1} + \beta(p^{k-1} - \omega^{k-1}v^{k-1})$$

Solve $A_{22}s_2^k = p_2^k$ using a precomputed LU-decomposition for A_{22} :

Solve $L_{22}w = p_2^k$ (forward substitution)

Solve $U_{22}s_2^k = w$ (backward substitution)

Start ML-AMG for solving $M_1s_1^k = p_1^k - A_{12}s_2^k$

Solve $M_1\tilde{s}_r^k = \text{real}(p_1^k - A_{12}s_2^k)$

Solve $M_1\tilde{s}_i^k = \text{imag}(p_1^k - A_{12}s_2^k)$

Set $s_1^k = \tilde{s}_r^k + \iota\tilde{s}_i^k$

Set $s = [s_1^k, s_2^k]$

$v^k = Ay$

$$\alpha = \frac{\rho^k}{\langle \tilde{r}^0, v^k \rangle}$$

$$s = r^{k-1} - \alpha v^k$$

Solve $A_{22}s_2^k = s_2^k$ using a precomputed LU-decomposition for A_{22} :

Solve $L_{22}w = s_2^k$ (forward substitution)

Solve $U_{22}s_2^k = w$ (backward substitution)

Start ML-AMG for solving $M_1s_1^k = s_2^k - A_{12}s_2^k$

Solve $M_1\tilde{z}_r^k = \text{real}(s_2^k - A_{12}s_2^k)$

Solve $M_1\tilde{z}_i^k = \text{imag}(s_2^k - A_{12}s_2^k)$

Set $z = \tilde{z}_r^k + \iota\tilde{z}_i^k$

Set $t = z$

$$\omega^k = \frac{\langle t, s \rangle}{\langle t, t \rangle}$$

$$x^k = x^{k-1} + \alpha y + \omega z$$

$$r^k = s - \omega^k t$$

end for

Conclusion: when the matrix used by the ML solver is real valued and the right-hand side is complex valued, it is possible to use ML and produce a complex valued result by combining the real part of the solution with the imaginary part. As the system matrix in the current application is not real valued for absorbing materials (permittivity and permeability) and a complex shift, it is only possible to use ML in testcases with a real shift in the shifted Laplace preconditioner. Section 3.4 contains the numerical results to analyze the incorporation of multigrid in the existing algorithm.

3.4 Numeric results

This section considers a cavity scattering problem to illustrate the use of the ML-AMG algorithm for the preconditioner solve. Subsection 3.4.1 describes the model problem and illustrates the results obtained for the numerical experiments.

3.4.1 A cavity scattering model problem

The cavity used for the experiments performed in this subsection has dimensions $1.5\lambda \times 1.5\lambda \times 0.6\lambda$ (see Figure 3.4.1). The wavenumber k_0 is chosen equal to 2π and therefore $\lambda = 1$. The Krylov subspace method used here is the Bi-CGSTAB method and the ML-AMG algorithm is used to perform the preconditioner solve. Due to the limitations of the ML-AMG algorithm, only real valued shifts can be used. Therefore, $\mathcal{M}_{(-1,0)}$ is chosen as shifted Laplace preconditioner and M_{new} is used as preconditioner (see Equation (3.2.6)). Other parameters for the Bi-CGSTAB method are the maximal number of iterations to perform, equal to 1000 and the specified tolerance $\left(\frac{\|r^k\|_2}{\|b\|_2}\right) \leq 10^{-6}$. For multigrid the following parameters are specified: one full MGV-cycle is applied², the Aztec smoother³ is chosen and no restriction on the number of grid levels. Note that there is no multigrid *solve* performed. Multigrid is used to *approximate* the inverse of preconditioner M_{new} .

In the experiments performed for this small cavity problem, several parameters are varied. The order of the basis functions is denoted by p , the mesh size is denoted by h and N denotes the total number of unknowns. The results obtained for the pure real shift $(\beta_1, \beta_2) = (-1, 0)$ are compared to results obtained for the shift $(\beta_1, \beta_2) = (1, -0.5)$. For the first shift the preconditioner is approximated by one ML cycle and for the latter shift, an exact preconditioner solve is performed in Matlab. In Table 2 the results are summarized. Note that in this table, the total number of matrix vector operations are summarized. Therefore the maximum number of matrix vector operations is equal to two times the maximal number of iterations, namely 2000.

²See Appendix D for an illustration of a full multigrid V-cycle.

³See the ML-guide (ref. 6) for more information about the Aztec smoother.

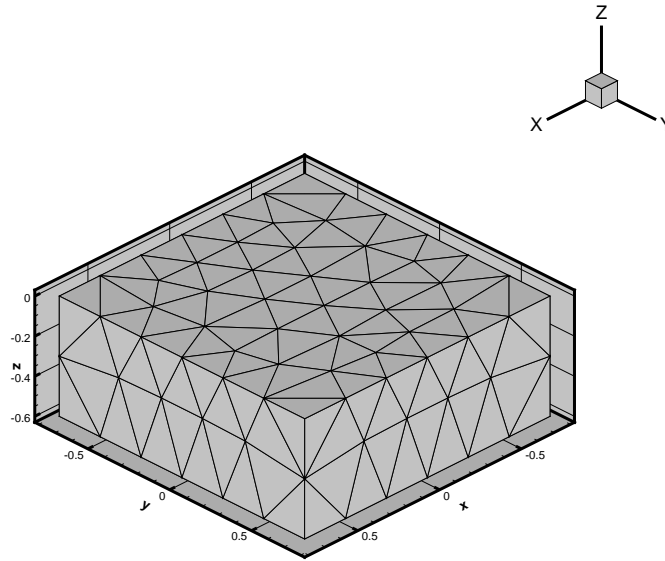


Fig. 3.4.1 Rectangular cavity with dimensions $1.5\lambda \times 1.5\lambda \times 0.6\lambda$. The discretization contains 2610 elements and 2796 degrees of freedom (N) for mesh size $h = 0.20$ and zeroth order basis functions.

p	h	N	$(\beta_1, \beta_2) = (-1, 0)$	$(\beta_1, \beta_2) = (1, -0.5)$	ML-solve for $(\beta_1, \beta_2) = (-1, 0)$
0	0.25	1402	586	490	730
0	0.20	2796	687	968	721
1	0.35	2914	651	1453	660
1	0.30	4344	657	+2000	717
1	0.25	7960	+2000	+2000	1052
2	0.35	5316	+2000	+2000	1253
2	0.30	8730	856	+2000	+2000

Table 2 Total number of matrix vector products for the Bi-CGSTAB–ML–AMG algorithm to solve a small cavity model problem. In these experiments M_{new} is used as preconditioner (see Equation (3.2.6)).

In Table 3 the CPU-times are summarized for the real shift with an exact solve for the preconditioner system versus the ML-AMG algorithm to perform the preconditioner solve.

p	h	N	exact preconditioner solve	ML-AMG for the preconditioner solve
0	0.20	2796	332.58	31.40
0	0.25	1402	70.55	15.23
1	0.25	7960	–	215.98
1	0.30	4344	937.59	98.69
1	0.35	2914	361.23	56.13
2	0.30	8730	2875.50	–
2	0.35	5316	–	436.51

Table 3 CPU-times for the Bi-CGSTAB–ML-AMG algorithm to solve a small cavity model problem.

From these tables the following can be concluded:

- When the optimal real shift $(\beta_1, \beta_2) = (-1, 0)$ is used, the performance of the proposed algorithm is unsatisfactory. The total number of matrix vector products is relatively high compared to the total number of unknowns N . In two cases the Bi-CGSTAB method does not converges.
- When the optimal complex shift $(\beta_1, \beta_2) = (1, -0.5)$ is used, the Bi-CGSTAB method does not converge in four cases considered in Table 2.
- For the zeroth order basisfunctions, it seems that the h -independency for the multigrid performance is maintained. Based on the results for the higher order basisfunctions in Table 2, nothing is this conclusion cannot be made.
- When the CPU-times for the exact preconditioner solve are compared to the CPU-times of the ML-AMG preconditioner approximation, it can be concluded that the ML-algebraic multigrid algorithm is a relatively fast solver to perform the preconditioner approximation.

3.5 Conclusions

Based on the experiments performed in this chapter the following conclusions can be made:

- Although the algebraic multigrid method for the preconditioner solve is incorporated in the same way geometric multigrid was included in the robust and efficient algorithm of Erlangga (ref. 6), the convergence behaviour of Bi-CGSTAB–ML-AMG algorithm is unsatisfactory. Note that for these experiments only the optimal real shift for the shifted Laplace preconditioner can be used due to the limitations of the ML package.

- When the optimal complex shift $(\beta_1, \beta_2) = (1, -0.5)$ from Erlangga (ref. 6) is considered, the performance of the algorithm is worse compared to the results from Erlangga.
- Based on the theoretical analysis in Chapter 4 from Abdoel (ref. 1), the ML-AMG algorithm performs as expected: this algorithm is a relatively fast solver to perform the preconditioner approximation.

It is concluded that the proposed algorithm does not lead to satisfactory results for the vector wave equation, opposed to the algorithm Erlangga (ref. 6) used for the Helmholtz equation. Therefore, the Bi-CGSTAB–ML-AMG algorithm will be applied on the Helmholtz model problem considered by Erlangga to compare the performance of the proposed Bi-CGSTAB–ML-AMG algorithm to the performance of the algorithm of Erlangga.

4 Iterative solution of the Helmholtz equation

4.1 Introduction

In the previous chapter it was concluded that the incorporation of multigrid in the existing algorithm, does not significantly improve the convergence of the chosen Krylov method. However, it is expected that the proposed algorithm should behave in the same way as the algorithm of Erlangga (ref. 6), which is based on *geometric* multigrid acceleration. The two main differences between the current application and the original model problem Erlangga considered are:

- The discretization method: finite difference method versus finite element method.
- The problem type: the Helmholtz equation versus the vector wave equation in the *curl-curl formulation*.
- The block structure of the system matrix in current application versus the matrix of the Helmholtz equation.

To understand where the proposed algorithm differs significantly from the algorithm proposed by Erlangga (ref. 6), a model problem is studied now where the discretization method is chosen to be the same as Erlangga used, namely the finite difference method.

The algorithm proposed in this thesis is applied on a problem chosen by Erlangga (ref. 6). This model problem concerns a scalar wave excited by a pulse. The resulting linear system will be solved by the algorithm presented in Chapter 3. The corresponding PDE with absorbing boundary conditions is stated below:

$$Lu = -S, \quad L = -\left(\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2}\right) + k_0^2 u, \quad (x_1, x_2) \in \Omega_h \quad (4.1.1)$$

$$\frac{\partial u}{\partial n} - \iota k_0 u = 0, \quad (x_1, x_2) \in \Gamma_h = \partial\Omega_h, \text{ where}$$

- $\Omega_h = (0, 1) \times (0, 1) \subset \mathbb{R}^2$,
- $\Gamma_h = \{(0, 0) \cup (0, 1) \cup (1, 0) \cup (1, 1)\}$,
- $N \in \mathbb{N} \rightarrow h = \frac{1}{(N+1)^2}$, with N the total number of unknowns in the x_1 as well as the x_2 direction,
- c is the P-wave velocity as an implicit function of space,
- $\omega^* = 2\pi f^*$ with f^* the frequency¹,
- k_0 is the dimensionless wavenumber defined as: $\frac{\omega}{c}$,
- S is a source term given by $\delta(x_1 - 0.5, x_2 - 0.5)$,
- $\iota^2 = -1$.

¹Note that * denotes a dimensionfull variable.

- The finite differences approximation of the partial differential operator L is given as

$$\tilde{L}_h = \frac{1}{h^2} \begin{bmatrix} & -1 & \\ -1 & 4 - k_0^2 h^2 & -1 \\ & -1 & \end{bmatrix}.$$

The corresponding linear system obtained from the discretization of equation (4.1.1) is denoted by:

$$A_h x = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} x = b. \quad (4.1.2)$$

Note that A_h is complex valued due to the absorbing (or radiation) boundary conditions. Other properties of the system matrix are: symmetric but not Hermitian and ill-conditioned. Furthermore, the matrix has a complex spectrum. The consequence of the system matrix being ill-conditioned and having a complex spectrum can be very important when an iterative method is used to solve the system. Standard iterative methods will show poor or even no convergence when they are used to solve this system. Therefore, Erlangga (ref. 6) considered the *shifted Laplace preconditioner* as special class of preconditioners for the Helmholtz equation. He showed that this shifted Laplace preconditioner is a very effective preconditioner to improve the convergence of Krylov subspace methods. This class of preconditioners is the subject of the next section. Section 4.3 considers the differences between a small cavity scattering model and the *three* dimensional Helmholtz equation. Finally, some conclusions are listed.

4.2 Application of the shifted Laplace preconditioner for the two dimensional Helmholtz equation

The shifted Laplace preconditioner can effectively be used to improve the convergence of iterative Krylov subspace methods (see Chapter 4 from Erlangga, ref. 6 and the references included there). This class of preconditioners is constructed by discretization of the following “shifted Laplace operator” with appropriate boundary conditions:

$$\mathcal{M}_{(\beta_1, \beta_2)} = -\Delta - (\beta_1 + \iota\beta_2)k_0^2, \quad \beta_1, \beta_2 \in \mathbb{R}, \quad \iota^2 = -1. \quad (4.2.1)$$

The goal is to solve the linear system from equation (4.1.2) by using a Krylov subspace method combined with (block)preconditioning techniques as discussed in Chapter 3. Note that this block-structure is a main difference with the work of Erlangga (ref. 6).

The discretization method used to obtain system (4.1.2) is based on finite differences with a standard 5-point stencil. In Figure 4.2.1 a surface plot of the solution is given. Figure 4.2.2 illustrates

the convergence behaviour of Bi-CGSTAB, using multigrid for the preconditioner solve. The multigrid method used here is the ML-AMG algorithm discussed in Subsection 3.3.1 and therefore, only the optimal real shift $(\beta_1, \beta_2) = (-1, 0)$ can be considered.

To obtain the pictures in Figures 4.2.1 and 4.2.2, the Krylov method used here is Matlab's intrinsic Bi-CGSTAB method. The other parameter settings are given in Table 4. The resemblance of Figure 4.2.1 with the result Erlangga (ref. 6) obtained for his first model problem, verify the proposed Bi-CGSTAB–ML-AMG algorithm.

parameter	value
maximum number of Bi-CGSTAB iterations	300
Krylov tolerance	$\frac{\ r^k\ _2}{\ b\ _2} \leq 10^{-6}$
wavenumber k_0	20
total number of unknowns N	$101^2 = 10.201$
multigrid cycle type	one full MGV-cycle
smoother	one Jacobi pre and post smoothing step
number of grid levels	no restriction

Table 4 Parameter settings.

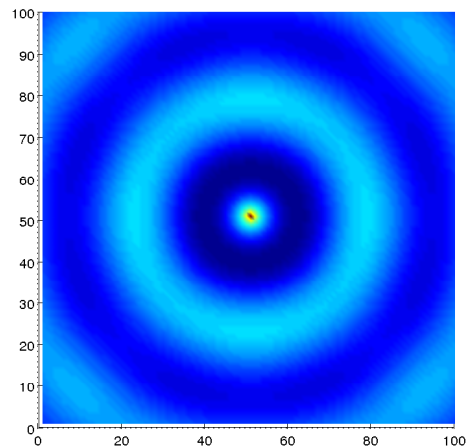


Fig. 4.2.1 Surface plot of the real part of the solution u .

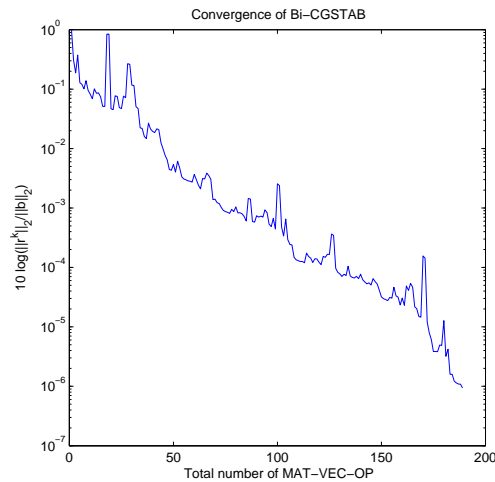


Fig. 4.2.2 Logarithm of the real part of the residual vector against the total number of matrix vector operations.

Erlangga (ref. 6) also concluded that using multigrid for the shifted Laplace preconditioner solve, leads to a robust and efficient iterative method to solve the discrete Helmholtz equation in two and three dimensions at very high wavenumbers. As Erlangga considered a regular, structured grid, it was possible to use geometric multigrid. Note that this is *not* the case in the current application. Other important conclusions based on this robust and efficient iterative method, are listed below:

- The convergence is nearly independent of the gridsize for constant wavenumber problems: h -independency of multigrid.
- The convergence depends almost linearly on the wavenumber for a fixed total number of unknowns.

For the current application with ML-AMG functioning as preconditioner solution method, the conclusions above also hold. This will be illustrated by comparing the following model problem to each other:

1. The two dimensional Helmholtz equation with the setup from Erlangga (ref. 6) without block preconditioning techniques.
2. The setup as in Figures 4.2.1 and 4.2.2 using the blockstructure of the system matrix from current application.

In Table 5 the total number of matrix vector products is illustrated using the blockstructure of the system matrix. The wavenumber $k_0 = 30$ is fixed while the total number of unknowns N is varied. Because of the restriction on real valued arithmetic, the shift considered here is the optimal real shift: $(\beta_1, \beta_2) = (-1, 0)$. Erlangga (ref. 6) showed that this is the optimal real shift

for the *Helmholtz* equation. This experiment illustrates the nearly h -independency of the ML-AMG algorithm: the total number matrix vector operations of the Bi-CGSTAB method, combined with ML-AMG to perform the preconditioner solve does not show a dramatic increase when the grid size is varied. Therefore, it can be concluded that the iterative method combining the Bi-CGSTAB method with ML-AMG as preconditioner solver, leads to a efficient method for the current application.

N	101^2	201^2	301^2	401^2
# MAT-VEC-OPs	322	397	466	477

Table 5 Total number of MAT-VEC-OP for the two dimensional Helmholtz equation: fixed wavenumber $k_0 = 30$, varying N .

Table 6 considers a fixed $N = 101^2$ and a variable wavenumber k_0 . From this table it can be concluded that for increasing wavenumber, the total number of matrix vector products for Bi-CGSTAB increases linearly.

k_0	10	20	30
# MAT-VEC-OPs	93	194	322

Table 6 Total number of MAT-VEC-OP for the two dimensional Helmholtz equation: fixed $N = 101^2$, varying k_0 .

Summarizing: From these experiments the same conclusions can be drawn as the conclusions made by Erlangga (ref. 6) listed above. However, note that for these experiments only the real shift can be considered. Erlangga (ref. 6) considered the optimal complex valued shift to obtain his results. Therefore, he obtained a smaller number of matrix vector operations.

As the goal is to solve a cavity scattering model with a similar block structure for the system matrix in the current application, the three dimensional Helmholtz equation as discussed by Erlangga (ref. 6) will be compared to a small cavity scattering model in the following section.

4.3 Differences between a small cavity scattering model and the three dimensional Helmholtz equation

In this section the same cavity problem as discussed in Section 3.4 is compared to one of Erlangga's model problems (ref. 6), namely the three dimensional Helmholtz equation.

The discretization matrix for the cavity scattering model is obtained for a three dimensional rectangular cavity with dimensions $1.5\lambda \times 1.5\lambda \times 0.6\lambda$ where λ denotes the wave length (see

Figure 3.4.1). The wavenumber k_0 is equal to 2π . It must be noted that this scattering problem is not equivalent to the current application because both the depth-to-diameter ratio $\frac{L}{d}$ and the wavenumber are very small. The purpose of this small scattering model is to analyse the performance of the Bi-CGSTAB method.

The model problem from Erlangga is the three dimensional Helmholtz equation with local absorbing boundary conditions. The following subsections will consider the differences between the small cavity scattering model and the model problem discussed by Erlangga (ref. 6).

4.3.1 Boundary conditions

In this subsection different types of boundary conditions for the two model problems are discussed. For the three dimensional Helmholtz equation the so called first order *local* absorbing boundary conditions are imposed, opposed to the *global* absorbing boundary conditions for the cavity scattering problem.

Furthermore, in the model problem from Erlangga, the absorbing boundary conditions are imposed on the whole boundary, while in the cavity scattering model, absorbing boundary conditions are considered on a small part of the boundary. The remaining part (say 90%) has Dirichlet boundary condition imposed.

4.3.2 Discretization

This subsection will emphasize the difference between the discretization method used for the two model problems. The three dimensional Helmholtz equation is discretized by the finite difference method using a 5-point stencil, while for the cavity scattering model a so called edge based finite element implementation is used.

In Chapter 6 further details about the similarity between a first order node based finite element implementation with a so called ‘lumped’ mass matrix and a second order finite difference implementation will be discussed.

4.3.3 Shift parameters

As already mentioned in Section 4.2, the shifted Laplace preconditioner can be effectively used to improve the convergence of the discretized Helmholtz equation. Another important note is that the scattering problem, described by the vector wave equation is a vector variant from the Helmholtz equation. Hence it is expected that the shifted Laplace preconditioner will also effectively improve the convergence of the linear system obtained from the discretization of the cavity scattering problem.

Erlangga (ref. 6) proved that the combination $(\beta_1, \beta_2) = (1, -0.5)$ is the optimal shift for the Helmholtz equation. When the goal is to analyse the effect of the ML-AMG algorithm, it is only possible to analyse a pure real shift. This because ML-AMG cannot perform computations in complex valued arithmetic. Therefore, the optimal real shift will be considered: $\mathcal{M}_{(-1,0)}$.

When the shift parameters are compared to the shift parameters used by Hooghiemstra (ref. 10, Chapter 5), different values are observed. This is a result of the fact that a different method is used to solve the preconditioner system. When the preconditioner system is solved by GCR, an experimentally observed optimal choice for $(\beta_1, \beta_2) = (0.5, 3.0)$.

4.3.4 The block upper triangular preconditioner matrix

As already mentioned in Subsection 3.2.1, using block preconditioning techniques is an important difference compared to the work and results obtained by Erlangga (ref. 6). Recall the structure of the system matrix A :

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}. \quad (4.3.1)$$

In this thesis, the first idea to obtain a preconditioner for this matrix is to introduce the shifted Laplace preconditioner with appropriate boundary conditions, analogous to Erlangga (ref. 6). The matrix resulting from the discretization of this operator is denoted by \tilde{A} :

$$\tilde{A} = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{bmatrix}. \quad (4.3.2)$$

The preconditioner M_H used by Hooghiemstra (ref. 10) is denoted below:

$$M_H = \begin{bmatrix} \tilde{A}_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}, \quad (4.3.3)$$

and the second idea for a preconditioner proposed in this thesis is to neglect block \tilde{A}_{21} , resulting in the block upper triangular preconditioner matrix M :

$$M = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ 0 & \tilde{A}_{22} \end{bmatrix}. \quad (4.3.4)$$

The matrix M will be used as block upper triangular preconditioner matrix and the effect on the total number of matrix vector products for Bi-CGSTAB will be compared in the experiments in this subsection. Note the difference with preconditioner M compared to the preconditioner

Hooghiemstra (ref. 10) used. In the next chapter, the effect of using \tilde{A} versus M will be analyzed.

Erlangga used $(\beta_1, \beta_2) = (1, -0.5)$ as optimal shift for the Helmholtz equation. Figure 4.3.1 illustrates the spectrum for $M^{-1}A$ when this shift is chosen with preconditioner M . Note that all eigenvalues have a positive real part for this shift.

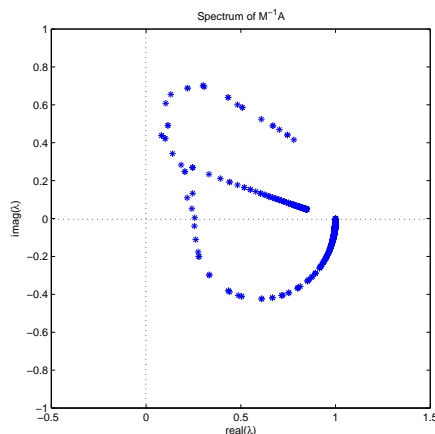


Fig. 4.3.1 Spectrum for $M^{-1}A$ with shift $(\beta_1, \beta_2) = (1, -0.5)$.

When using the ML-AMG algorithm, only real shifts can be chosen and Erlangga (ref. 6) also proved that the optimal real valued shift for the Helmholtz equation is $(\beta_1, \beta_2) = (-1, 0)$. For this combination and preconditioner M , the spectrum of the preconditioned system is given in Figure 4.3.2. When this shift is used, the preconditioned system still has eigenvalues on the left part of the imaginary axis. Note that these eigenvalues have an imaginary part relatively larger than their real part.

When performing the experiments with the Helmholtz equation, Erlangga (ref. 6) also noticed that the convergence of Bi-CGSTAB using the real shift, was poor compared to the case where he considered shift $(\beta_1, \beta_2) = (1, -0.5)$. He analyzed another technique which performs a *rotation* of the eigenvalues and is discussed in the next subsection.

4.3.5 Rotation of the eigenvalues

In this subsection a technique is considered which rotates the eigenvalues of a linear system in such a way that after the transformation, all eigenvalues are located on the right side of the imaginary axis (see Van Gijzen et al., ref. 15).

For the experiments performed in this chapter, the effect of the rotation technique on the performance of the preconditioned Bi-CGSTAB method is analyzed. Figure 4.3.2 illustrates the spec-

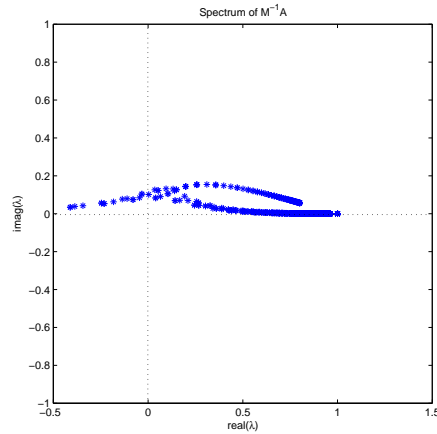


Fig. 4.3.2 Spectrum for $M^{-1}A$ with shift $(\beta_1, \beta_2) = (-1, 0)$.

trum for $M^{-1}A$ when shift $(\beta_1, \beta_2) = (-1, 0)$ is used before the rotation is applied. When the spectrum is rotated around $(0, 0)$ by $+\frac{1}{2}\pi$, all eigenvalues will be located on the right side of the imaginary axis. This situation is shown in Figure 4.3.3.

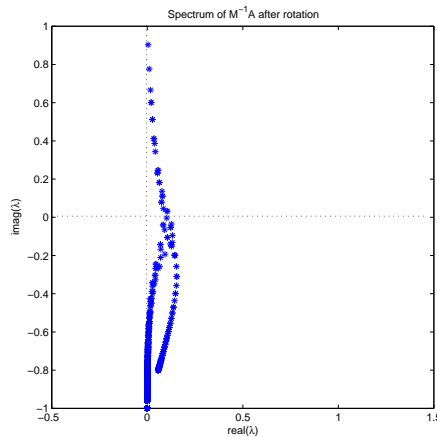


Fig. 4.3.3 Spectrum for $M^{-1}A$ with shift $(\beta_1, \beta_2) = (-1, 0)$ after rotation.

Table 7 summarizes the results for the two dimensional Helmholtz equation analogue to the experiments performed in Section 4.2 for a total number of unknowns equal to $N = 101^2$ and wavenumber $k_0 = 20$. The other parameter settings are the same as denoted in Table 4.

# MAT-VEC-OPs without rotation	# MAT-VEC-OPs with rotation
102	102

Table 7 Total number of matrix vector operations for the preconditioned Bi-CGSTAB method with and without rotation of the spectrum for $M^{-1}A$.

From this table it can be concluded that the rotation technique has no effect on the performance of the preconditioned Bi-CGSTAB method when the pure real shift is used. For more results about using this rotation technique, the reader is referred to Chapter 4 from Erlangga (ref. 6). The conclusions based on the experiments performed in this chapter are listed in the next section.

4.4 Conclusion

From the experiments performed in this subsection the following can be concluded:

- Using the block upper triangular preconditioner for the preconditioned Bi-CGSTAB method with only a real shift, does not lead to a significant improvement of the convergence. From the spectrum it can be concluded that using M as preconditioner defined in Equation (4.3.4), does not lead to a system with a favorable spectrum. More specifically, the imaginary part of the eigenvalues are relatively larger compared to their real part. To understand how the spectrum can be modified in a favorable way, the effect of different shift combinations on the spectrum is analyzed in the following chapter.
- Using M with shift $(\beta_1, \beta_2) = (1, -0.5)$, results in a linear system with a favorable spectrum: all eigenvalues are located on the right side of the imaginary axis resulting in eigenvalues with only positive real part. Hence it is expected that Bi-CGSTAB will converge when this preconditioner is used.
- When the eigenvalues which are located on the left side of the imaginary axis are rotated in such a way that all eigenvalues are mapped onto the right side of the imaginary axis, the convergence of Bi-CGSTAB is not affected. It seems that the convergence of this Krylov method does not depend on the *location* in the complex plane, but that the *distance* of the eigenvalues to the center point $(0, 0)$ is important (see Chapter 5 for more about this phenomenon).

5 Short recurrence Krylov methods for linear systems with complex eigenvalues with an imaginary part relatively larger than their real part

5.1 Introduction

In the previous section it was shown that using the block upper triangular matrix unfortunately does not lead to a system with a favorable spectrum for the two dimensional Helmholtz equation. The convergence of the Bi-CGSTAB method is somewhat improved, but the resulting linear system has eigenvalues with an imaginary part, relatively larger than the real part. When Bi-CGSTAB is used to solve these systems, Sleijpen and Fokkema (ref. 17) showed that the so called *one step minimal residual polynomials* (MR-polynomials) can cause stagnation or breakdown in the algorithm. This will be shortly discussed in Subsection 5.2.1.

Before the convergence of Bi-CGSTAB is discussed, the following section will illustrate the spectra of the preconditioner and the preconditioned system for two different preconditioners and for different values of the shift for the shifted Laplace preconditioner.

5.2 The spectral properties of the preconditioner system and the preconditioned system

The model problem considered in this chapter is the two dimensional Helmholtz equation with local absorbing boundary conditions, discretized by the finite difference method. This model problem is chosen in order to compare the results obtained in this thesis to the results from Erlangga (ref. 6).

In Section 4.3.4 two possible block preconditioners are stated in Equations (4.3.2) and (4.3.4). The purpose here is to investigate the effect of these two preconditioners on the spectrum for the linear system. For future experiments in this chapter, the following notation will be used:

$$M_1 = \tilde{A} = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{bmatrix} \text{ and } M_2 = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ 0 & \tilde{A}_{22} \end{bmatrix}. \quad (5.2.1)$$

When preconditioner M is chosen equal to either M_1 or M_2 , the different effects on the spectrum of the preconditioned system $M^{-1}A$ can be illustrated. Note that \tilde{A} results from the discretization of the shifted Laplace preconditioner with appropriate boundary conditions. Therefore, also the effect of different shift-combinations (β_1, β_2) on the spectra is illustrated.

Figure 5.2.1 illustrates the spectra of $M^{-1}A$ with $M = M_1$ for different combinations of (β_1, β_2) . In the upper left picture, the preconditioner $M = A$ for shift $(1, 0)$ and therefore all eigenvalues for the preconditioned system are equal to one. The upper right picture considers the case when

the shift is chosen equal to $(-1, 0)$. Note that there is a clustering of a part of the eigenvalues around $(1, 0)$ in the complex plane. There are also some eigenvalues with a complex part relatively larger than their real part. The lower pictures consider the complex shift $(0, -1)$ in the left picture. Using this shift still results in eigenvalues with a complex part which is relatively larger than their real part. Therefore, the convergence of Bi-CGSTAB will still not be optimal. The last illustration for this preconditioner considers shift $(\beta_1, \beta_2) = (1, -0.5)$. This shift results in a spectrum with all eigenvalues with positive real part, *away* from $(0, 0)$ (this is expected from the analysis performed in Van Gijzen et al., ref. 15). Therefore, it is expected that Bi-CGSTAB will be able to solve this system showing a satisfying convergence behaviour. Note that these results are analogue to the results in Chapter 3 from Erlangga (ref. 6).

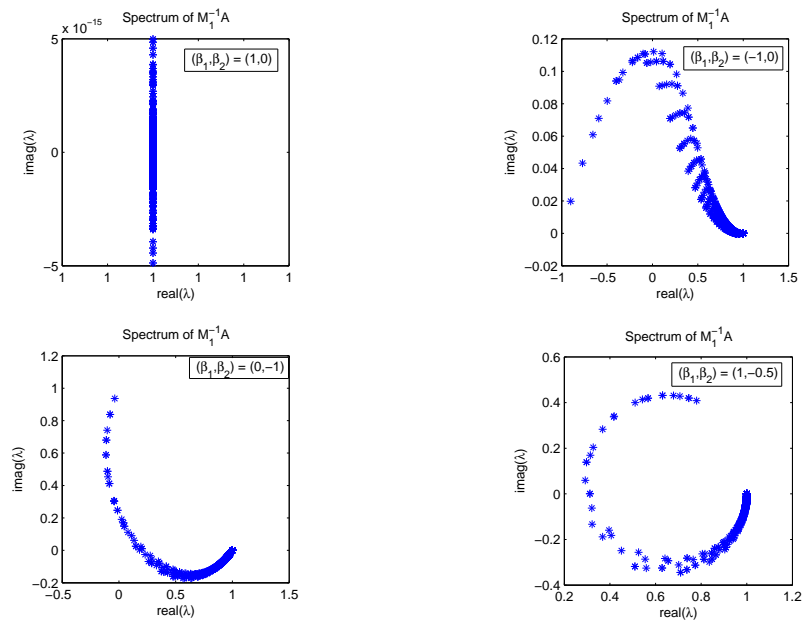


Fig. 5.2.1 Different spectra for preconditioner $M_1^{-1}A$ and different combinations of the shift (β_1, β_2) .

In Figure 5.2.2 the spectra of $M^{-1}A$ with $M = M_2$ are illustrated for the same shifts as in the previous experiment. The upper left picture considers shift $(1, 0)$ and it can be seen that most of the eigenvalues are clustered around zero. Therefore, it is expected that Bi-CGSTAB will show an unsatisfactory convergence behaviour. For the other three shifts, the same conclusions can be drawn as for preconditioner M_1 . Note that removing block M_{21} in case $(\beta_1, \beta_2) = (1, -0.5)$ causes the eigenvalues of preconditioned system $M^{-1}A$ to move a *little towards* the imaginary axis. Hence it is expected that the convergence behaviour of Bi-CGSTAB will *not* be significantly affected. When these preconditioners are used to test their performance on the discretized

Helmholtz equation, it can indeed be concluded that the convergence of Bi-CGSTAB is not significantly affected when M_2 is used instead of M_1 (these results are included in Appendix E). This concludes the discussion for the preconditioned system. Hereafter, the preconditioner system is analyzed.

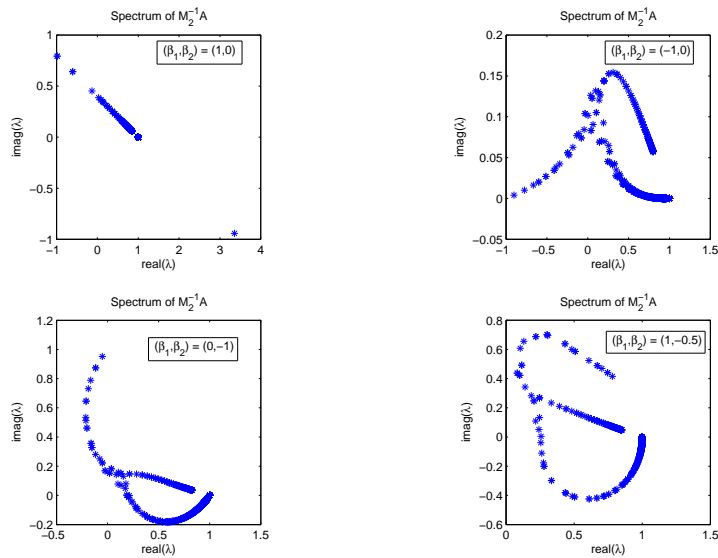


Fig. 5.2.2 Different spectra for preconditioner $M_2^{-1}A$ and different combinations of the shift (β_1, β_2) .

In Figure 5.2.3 the spectrum of block M_{11} is illustrated for $M = M_1$ (for $M = M_2$ the same figures are obtained). Note that for both choices of M , the preconditioner system use the same blockmatrix \tilde{A}_{11} to perform the preconditioner solve. In Chapter 4 of Abdoel (ref. 1) it was shown that when multigrid is used to solve a linear system, the corresponding matrix must have eigenvalues with positive real part. From Figure 5.2.3 it follows that this is the case for block M_{11} . Therefore, the ML-AMG algorithm can be used to perform the preconditioner solve.

Conclusions:

- With the restriction on using the real shift, the matrix of the preconditioned system still has a complex spectrum for both preconditioners M_1 and M_2 . Therefore, the following subsection considers the convergence behaviour of Bi-CGSTAB for linear systems with a complex spectrum.
- The matrix used to perform the preconditioner solve has eigenvalues with positive real part for all the shift combination considered in this section. Therefore, it is possible to use multigrid to solve the shifted Laplace preconditioner system.

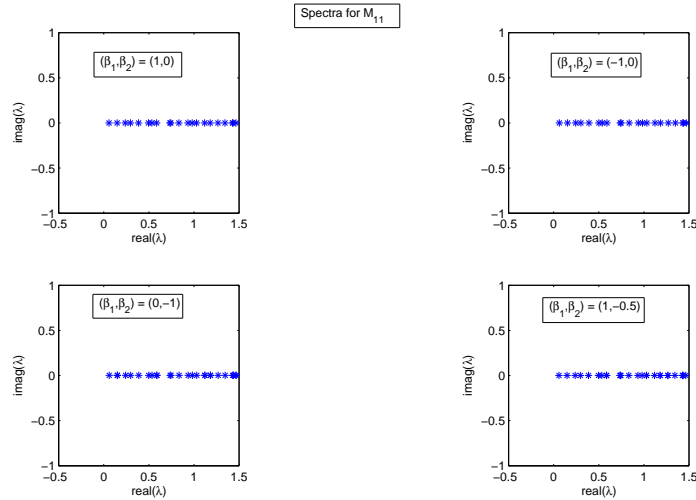


Fig. 5.2.3 Spectrum for M_{11} for $M = M_1$ and different combinations of the shift (β_1, β_2) .

5.2.1 Bi-CGSTAB convergence for linear systems with complex eigenvalues with an imaginary part relatively larger than their real part

This subsection considers the Bi-CGSTAB convergence for linear systems $Ax = b$ with a complex spectrum. This is because the system matrix A in the current application also has a complex spectrum. More specifically, the imaginary part of the eigenvalues are larger than their corresponding real part. When the matrix is a positive definite Hermitian matrix, a complete convergence analysis can be performed for the CG methods. However, when the matrix is not Hermitian or has complex valued eigenvalues, the convergence analysis collapses. For matrices with a complex spectrum, Sleijpen and Fokkema (ref. 17) considered the performance of the so called *Bi-CGSTAB(l)* methods, inspired by the Bi-CGSTAB2 method from Gutknecht (ref. 8).

For $l = 1$, Bi-CGSTAB(1) coincides with the Bi-CGSTAB method.

The family of the Bi-CG methods consider a so called *search correction* that depends on the *true* residual $r_k = b - Ax_k$ and some “*shadow residual*” \tilde{r}_k in each iteration step k . The residuals r_k are made orthogonal to \tilde{r}_k . These r_k can be written as $r_k = p_k(A)r_0$, where p_k is some appropriate polynomial of degree k . The roots of this polynomial are approximations of the eigenvalues. In the *CGS algorithm* of Sonneveld (ref. 18) this Bi-CG polynomial is applied twice, resulting in: $r_k = p_k(A)^2 r_0$.

The Bi-CGSTAB algorithm generates residuals using a product of two polynomials:

$r_k = q_k(A)p_k(A)r_0$, where the polynomials q_k are appropriate *one-step minimal residuals polynomials* (MR polynomials): $1 - \omega_k t$ for some optimal ω_k . When ω_k becomes close to zero, it may cause the Bi-CGSTAB algorithm to stagnate or breakdown: starting with residual r_j results in a

new residual that can be written as $r_{j+1} = \omega_k r_j$. So for small ω_k there is almost no minimization of the residuals.

Sleijpen et al. (ref. 17) showed that when the system matrix has a complex spectrum, it is likely that ω_k becomes very small. They proposed to use l -degree polynomials to handle the situation when the ω_k become nearly zero. This because e.g. second order polynomials are able to approximate a complex root. The methods using these l -degree polynomials are denoted by Bi-CGSTAB(l) methods.

The parameter ω_k is likely to become nearly zero when the system matrix A has nonreal eigenvalues with an imaginary part that is large relative to the real part. For the small cavity problem introduced in Section 3.4.1, Figures 5.2.4 and 5.2.5 illustrate the value of ω_k in the complex plane when preconditioner M_1 is used. In this experiment, zeroth order basis functions are used with mesh size $h = 0.15$. Figure 5.2.4 considers shift $(\beta_1, \beta_2) = (-1, 0)$ and Figure 5.2.5 considers shift combination $(\beta_1, \beta_2) = (1, -0.5)$. As can be seen, a lot of the ω_k are close to the centre $(0, 0)$ for shift $(\beta_1, \beta_2) = (-1, 0)$. When the ω_k are close to zero, the residuals are not effectively minimized. For $(\beta_1, \beta_2) = (1, -0.5)$, the ω_k have strictly positive real part resulting in an relatively effective decrease of the residuals in each iteration.

These experiments explain why the Bi-CGSTAB method in the proposed algorithm leads to an unsatisfactory convergence behaviour, with the restriction on using the pure real shift.

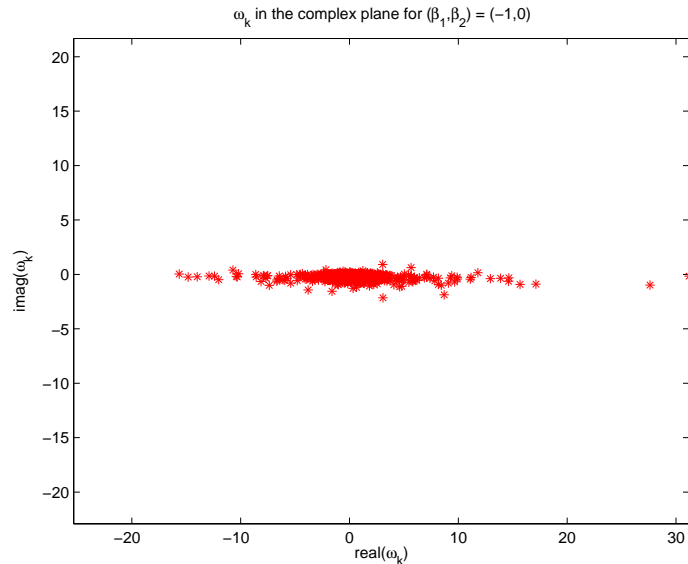


Fig. 5.2.4 ω_k in the complex plane for preconditioner M_1 and shift $(\beta_1, \beta_2) = (-1, 0)$.

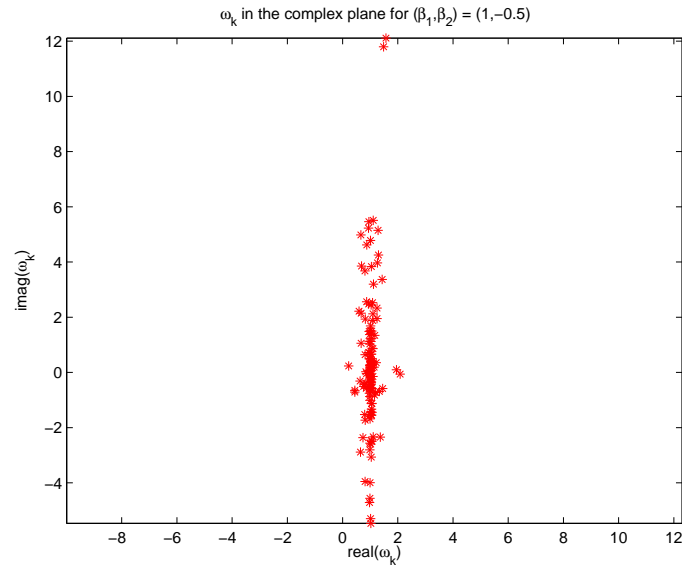


Fig. 5.2.5 ω_k in the complex plane for preconditioner M_1 and shift $(\beta_1, \beta_2) = (1, -0.5)$.

5.2.2 IDR convergence for linear systems with complex eigenvalues with an imaginary part relatively larger than their real part

In this subsection the work of Sonneveld and Van Gijzen (ref. 19) is considered. They analyzed the so called *Induced Dimension Reduction* (IDR) methods and made the extension to the $IDR(s)$ methods as a family of new iterative solution algorithms based on the IDR mechanism. The IDR mechanism uses short recurrences and is able to produce the exact solution of a linear system with dimension N , in at most $2N$ matrix vector operations in exact arithmetic (when $IDR(s)$ is considered, at most $(N + \frac{N}{s})$ matrix vector operations are needed). The parameter s also denotes the number of minimization steps of the residuals r_k . IDR methods also use an iteration polynomial, just as the Bi-CG methods do. The iteration polynomial constructed by IDR is the product of the Bi-CG polynomial with another locally minimizing polynomial. The residuals are now forced to be in subspaces of *decreasing* dimension. For a more in depth discussion of these polynomials and construction of these subspaces, the reader is referred to Sonneveld et al. (ref. 19) and the references within.

Sonneveld et al. (ref. 19, Section 6.4) considered the three dimensional Helmholtz equation discretized on a rectangular grid with a combination of Neumann and local absorbing boundary conditions imposed on the boundary. The system is discretized by the finite element method using tetrahedral elements on a grid with gridsize $h = 8\text{cm}$ and the wavenumber was chosen equal to $k_0 = 100\text{ Hz}$. Furthermore, they used standard $ILU(0)$ as preconditioner. The size of the dis-

cretized system is 132.651 and the number of nonzero diagonals in the matrix is 19. The system matrix for this system is complex valued, symmetric and also has a similar spectrum as the system matrix in current application. For the convergence behaviour of $IDR(s)$, the same problem can occur as in the Bi-CGSTAB case: the parameter ω_k can become very small. To overcome this, analogue to the Bi-CGSTAB(l) algorithm, an improved computation of ω_k can be used. Table 8 summarizes their results:

Method	# MAT-VECs	Time [s]	# MAT-VECs with improved ω_k	Time [s]
IDR(1)	1500	3322	678	1483
IDR(2)	598	1329	474	1051
IDR(4)	353	783	323	716
IDR(6)	310	698	267	601
Bi-CGSTAB(1)	1828	3712	640	1300
Bi-CGSTAB(2)	1008	2045	652	1323
Bi-CGSTAB(4)	656	1362	608	1263
Bi-CGSTAB(8)	608	1337	608	1337

Table 8 Number of matrix vector multiplications and elapsed time for the three dimensional Helmholtz equation with and without improved computation of ω_k .

From this table the following conclusions can be drawn:

- When there is no optimization of the parameter ω_k , Bi-CGSTAB(l) needs more matrix vector multiplications compared to $IDR(s)$.
- When the computation of ω_k is improved, the total number of matrix vector multiplications of Bi-CGSTAB(l) and $IDR(4)$ is similar. The performance however, is not: Bi-CGSTAB performs twice the number of matrix vector multiplications compared to $IDR(4)$. For this reason, $IDR(4)$ is the favorable method.
- The performance of $IDR(4)$ and $IDR(6)$ is almost the same. As the parameter s denotes the number of minimization steps of the residuals r_k , the amount of work of $IDR(6)$ is relatively more compared to $IDR(4)$. Therefore, $IDR(4)$ will be considered in this thesis.

Sonneveld et al. (ref. 19) also noted a close relationship between the convergence behaviour of GMRES (as long recurrence method) and the convergence of the short recurrence $IDR(s)$ methods. It seems that for this problem, the convergence of $IDR(s)$ is bounded from below by the convergence curve of GMRES. For the convergence analysis for matrices with a complex valued spectrum, the same can be said as in the case for Bi-CG: the convergence analysis collapses. In Chapter 6 of Sonneveld et al. (ref. 19), the convergence behaviour of $IDR(s)$, Bi-CGSTAB(l) and

GMRES are compared to each other. It turns out that for this specific problem, the total number of matrix vector operations for Bi-CGSTAB(l), for $l = 1, 2, 4, 8$ is similar to the total number of matrix vector operations performed by IDR(1). The same conclusion can be made when IDR(4) is compared to IDR(6).

For GMRES the convergence can be proved under some conditions. The following theorem is reproduced from Vuik and Oosterlee (ref. 2), and the proof can be found in Saad and Schultz (ref. 20, page 866). This theorem gives an indication of the bounds on the norm of the residual for a general eigenvalue distribution of the eigenvalues of a matrix.

Theorem (Saad and Schultz, ref. 20)

Suppose that matrix A has N eigenvectors and is diagonalizable so that $A = XDX^{-1}$. Here the columns of X are the eigenvectors and D is a diagonal matrix with on the diagonal the eigenvalues of A . Let P_m be the space of all polynomials of degree less than m and let $\sigma = \{\lambda_1, \dots, \lambda_N\}$ represent the spectrum of A .

Define:

$$\varepsilon^{(m)} := \min_{\substack{p \in P_m \\ p(0)=1}} \max_{\lambda_i \in \sigma} |p(\lambda_i)|$$

$$K(X) := \|X\|_2 \cdot \|X^{-1}\|_2$$

Then the residual norm of the m -th iterate satisfies:

$$\|r^m\|_2 \leq K(X) \varepsilon^{(m)} \|r^0\|_2$$

If furthermore all eigenvalues are enclosed in a circle centered at $C \in \mathbb{R} : C > 0$ and having radius $R < C$, then

$$\varepsilon^{(m)} \leq \left(\frac{R}{C}\right)^m$$

When the optimal complex shift is chosen in the current application, Figures 5.2.1 and 5.2.2 illustrate that the eigenvalues are contained in a circle. This theorem states that full GMRES will converge and based on the experiments performed by Sonneveld et al. (ref. 19), it is likely that the IDR(s) methods will also converge for this three dimensional Helmholtz equation. The next section lists the conclusions based on the analysis performed in this chapter.

5.3 Conclusions

Based on the experiments performed in this chapter the following conclusions can be drawn:

- With the restriction on the pure real shift, the preconditioned system matrix has a complex valued spectrum with the imaginary part relatively larger than their real part. Therefore, the parameter ω_k is nearly zero which leads to a stagnation in the minimization step of the Bi-CGSTAB method.
- Based on the work of Sonneveld et al. (ref. 19), it seems that the Bi-CGSTAB(ℓ) and IDR(s) methods show better convergence compared to the Bi-CGSTAB method. Furthermore, the amount of work from Bi-CGSTAB(ℓ) is twice the amount of work of IDR(s). Therefore, IDR(4) is chosen in this thesis.
- It seems that the IDR(4) method is a good alternative for the existing long recurrence GCR algorithm of Hooghiemstra (ref. 10). Therefore, the *second proposed algorithm* in this thesis is: ***IDR(4) combined with the ML-AMG algorithm to perform the preconditioner solve.***

6 Iterative solution of the two dimensional vector wave equation

6.1 Introduction

In Chapters 3 and 4, respectively, the three dimensional vector wave equation and the two dimensional Helmholtz equation have been discussed. It was expected that the performance of the proposed Bi-CGSTAB–ML-AMG algorithm in this thesis should be as good as the performance of the algorithm of Erlangga (ref. 6) for the Helmholtz equation. In Chapter 5 it was explained why this was not the case and a newly proposed algorithm was stated. To test the performance of the newly proposed IDR(4)–ML-AMG algorithm, the *two dimensional* vector wave equation will be considered in this chapter. It is expected that results obtained for the two dimensional vector wave equation can be extended to the three dimensional vector wave equation as seen in Chapters 2 and 3. Obviously, in two dimensions the total number of unknowns is relatively small for cavity scattering model problems of intermediate size compared to the three dimensional case. This means that larger cavities can be studied for relatively high values of the wavenumber.

The analysis of the iterative solution method will be done using several model problems, starting with a model problem that resembles the two dimensional Helmholtz equation as closely as possible. In this way, the performance of the solution procedure can be directly compared to the results obtained in Chapter 3 and the results from Erlangga (ref. 6). The flowchart in Figure 6.1.1 illustrates the main differences (which had been identified in Chapter 4), between the different solution algorithms presented in this thesis for the finite element discretization of the two dimensional vector wave equation which are considered in this chapter and the original algorithm used by Hooghiemstra (ref. 10). The aim is not to find an alternative formulation of the equations to be solved, but to study the behaviour of the iterative method for different formulations of the finite element discretization and the boundary conditions. The main differences which had been identified in Chapter 4, are denoted below:

- A *node* based finite element discretization versus an *edge* based finite element discretization. When in the node based implementation the entries of the element matrices are ‘*lumped*’¹, the resulting stencil is identical to the finite difference stencil of Erlangga (ref. 6) obtained for the Helmholtz equation. In this way the iterative solution method in this chapter can be directly compared to the results of Erlangga. The edge based implementation is considered because this discretization method is used in the original application used by Hooghiemstra (ref. 10). In the original algorithm edge based basis functions are chosen because of their natural continuity properties and their ability suppress so called spurious solutions.

¹The off diagonal entries in the matrix are added to the main diagonal element.

- The application of *local* versus *global* absorbing boundary conditions. Erlangga (ref. 6) considered local absorbing boundary conditions for the Helmholtz equation and in the current application (Hooghiemstra, ref. 10) global absorbing boundary conditions are imposed. These global boundary conditions are used, because this guarantees the correct farfield behaviour of the solutions.

It is important to note that when the node based implementation is considered, the E_z -component is solved from a PDE similar to a two dimensional Helmholtz equation. When the edge based implementation is considered, the E_x and E_y -components are solved. Hence, the latter case actually solves the H_z -problem (see Chapter 10 from Jin, ref. 13):

- Write $\mathbf{H} = (0, 0, H_z)^T$ and $\mathbf{E} = (E_x, E_y, 0)^T$.
- It holds that $\nabla \times \mathbf{E} = -\iota\omega\mu\mathbf{H}$.

$$\Rightarrow \nabla \times \mathbf{E} = \nabla \times (E_x, E_y, 0)^T = \begin{vmatrix} \underline{i} & \underline{j} & \underline{k} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & 0 \\ E_x & E_y & 0 \end{vmatrix} = \iota\omega(0, 0, H_z)^T.$$

The corresponding PDE for the two dimensional vector wave equation is given below for the E_z problem, using the identities $\nabla \times \nabla E_z = -\Delta E_z - \nabla(\nabla \cdot E_z)$ and $\nabla \cdot E_z = 0$:

$$-\Delta E_z - k_0^2 \varepsilon_r E_z = 0 \quad (6.1.1)$$

Here, E_z denotes the electric field, k_0 is the dimensionless wavenumber and J denotes the source.

The corresponding PDE for the two dimensional vector wave equation is given below for the H_z problem:

$$\nabla \times \nabla \times \mathbf{E} - k_0^2 \varepsilon_r \mathbf{E} = 0 \quad (6.1.2)$$

Here vector $\mathbf{E} = (E_x, E_y, 0)$ denotes the electric field, k_0 is the dimensionless wavenumber and J denotes the source.

The first model problem considers the two dimensional vector wave equation discretized on a square grid with only *local* absorbing boundary conditions imposed on the whole boundary. For this discretization, a node based finite element implementation is used. Furthermore, the element matrices are ‘lumped’ to get a discretization stencil identical to a stencil obtained for the finite differences discretization method used in Chapter 4. Recall that when the node based FEM implementation is used to numerically solve Equation (6.1.1), E_z is calculated. When the entries in the element matrices are lumped, or when the Newton Cotes numerical integration rule

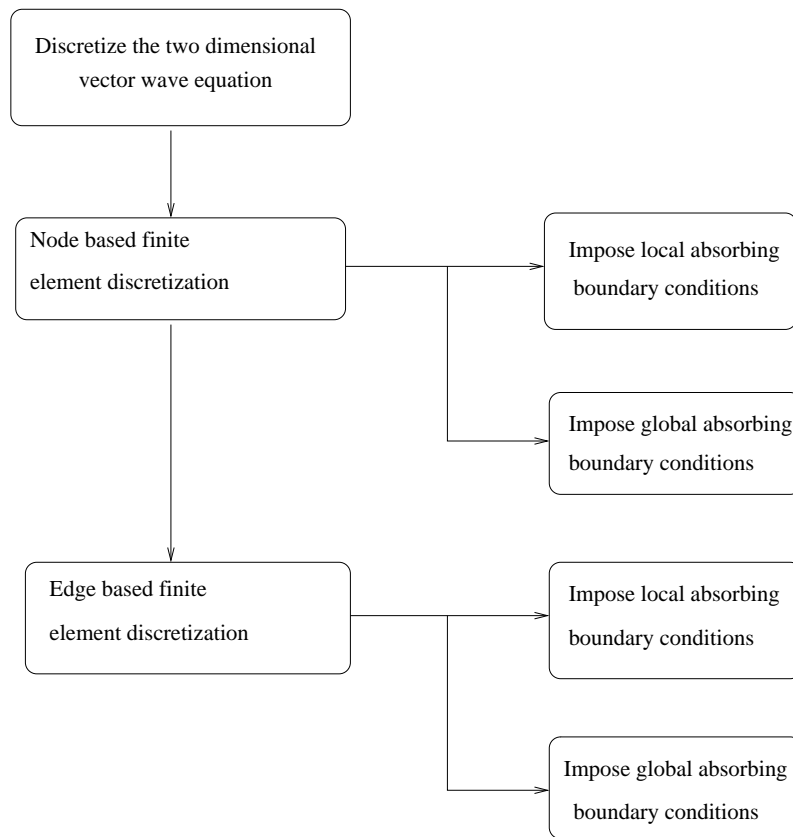


Fig. 6.1.1 Flowchart for the two dimensional vector wave equation.

is used to calculate the integrals in the Ritz formulation, the resulting stencil for this two dimensional vector wave equation is similar to the stencil obtained in Chapter 4 for the two dimensional Helmholtz equation. The results obtained for this model problem resemble the results for the two dimensional Helmholtz equation. Therefore, these results are not included in this chapter.

The second problem considers the two dimensional cavity discretized using a *node* based finite element implementation with *local* absorbing boundary conditions imposed on the aperture. The third model problem has the same setup with the local absorbing boundary conditions replaced by *global* boundary conditions. These global boundary conditions imposed on the aperture are identical to the boundary conditions imposed in the current application, but the discretization method is not: this model problem is discretized by a node based FEM implementation while in the current application, an edge based FEM implementation is used. Therefore, the last two model problems consider an *edge* based finite element implementation. See Table 9 below.

Model problem	Discretization type	Absorbing boundary condition type
#1	node based + lumping	local
#2	node based	local
#3	node based	global
#4	edge based	local
#5	edge based	global

Table 9 Overview of the model problems with the discretization type used and the absorbing boundary conditions imposed.

In Figure 6.1.2 an example of a two dimensional cavity with dimensions $L \times d$ is given. In this chapter $L = 8$ and $d = 1$. The green dotted line denotes the aperture.

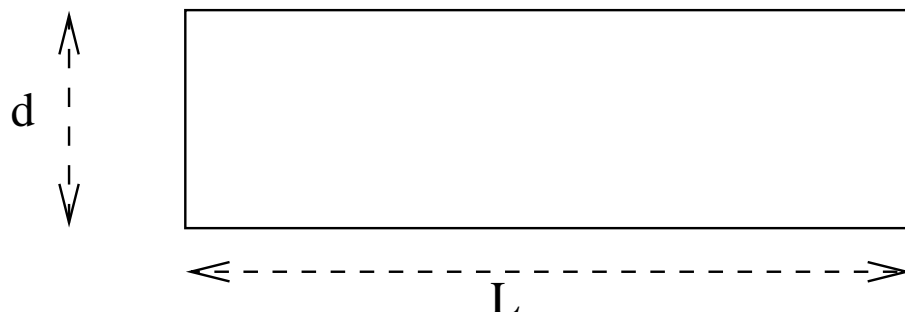


Fig. 6.1.2 Two dimensional cavity with dimensions $L \times d$.

Discretization of the PDE mentioned above together with the absorbing or integral boundary conditions, leads to a linear system:

$$Au = f, \quad \text{with } A \in \mathbb{C}^{N \times N} \text{ and } u, f \in \mathbb{C}^N \quad (6.1.3)$$

Here N denotes the total number of unknowns and it is important to note that A is complex valued because of the inclusion of the absorbing boundary conditions. These boundary conditions lead to complex valued entries similar to the local absorbing boundary conditions considered in the two dimensional Helmholtz problem. Global boundary conditions lead to complex valued entries because of the complex valued Green's function (see Subsection 2.3.1).

Other important properties (block structure and ill-conditionedness) of the system matrix A are discussed in Chapter 2. To get a system matrix with a similar block structure as the block structure of the system matrix in the current application, a reordering procedure is performed to arrange all the complex valued entries, due to the boundary conditions, in the lower right block of the matrix.

In Figure 6.1.3 the contour plot is illustrated for the two dimensional cavity. Here the wavenumber $k_0 = 2\pi$ and the mesh size $h = \frac{1}{32}$. For this figure the node based implementation is used and therefore, the E_z field is reproduced. It can be seen how the wave travels through the inlet and is reflected on the bottom.

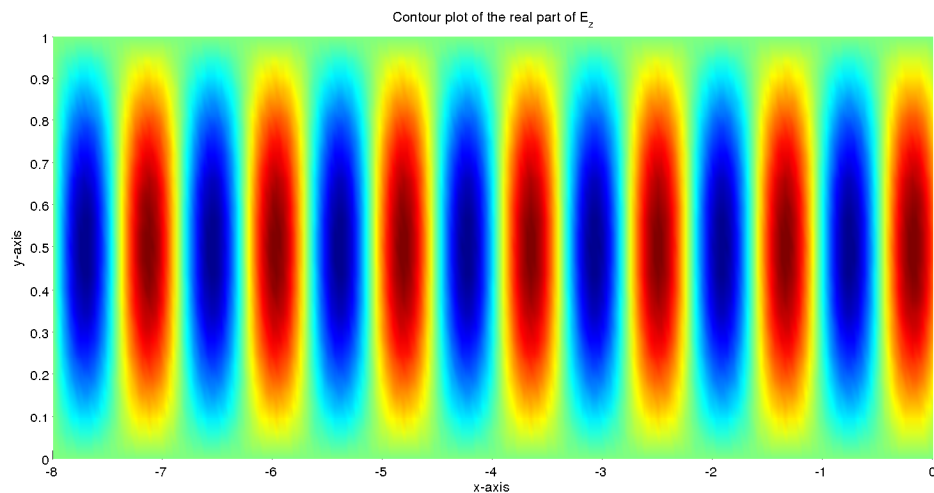


Fig. 6.1.3 Contour plot of E_z .

Figure 6.1.4 illustrates the vector plot of E_x and E_y for the two dimensional cavity. In Figure 6.1.5 the contour plot is illustrated for the two dimensional cavity obtained with the edge based

implementation. Therefore, the H_z is reproduced in this figure. The wavenumber $k_0 = 2\pi$ and the mesh size $h = \frac{1}{32}$.

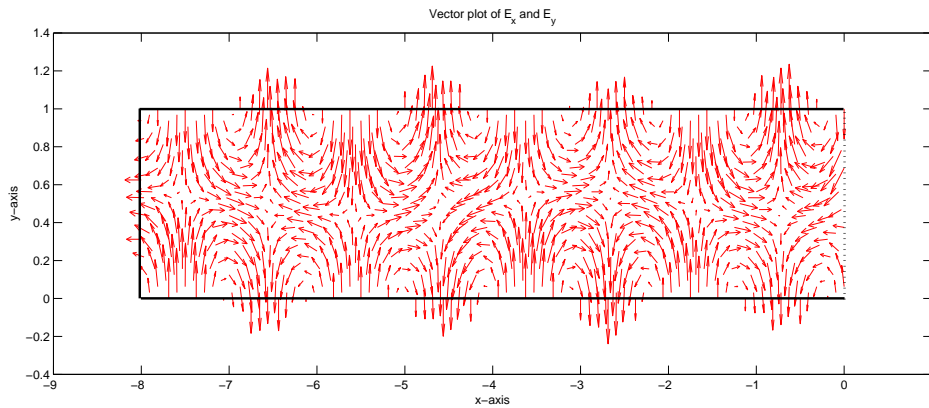


Fig. 6.1.4 Vector plot of E_x, E_y .

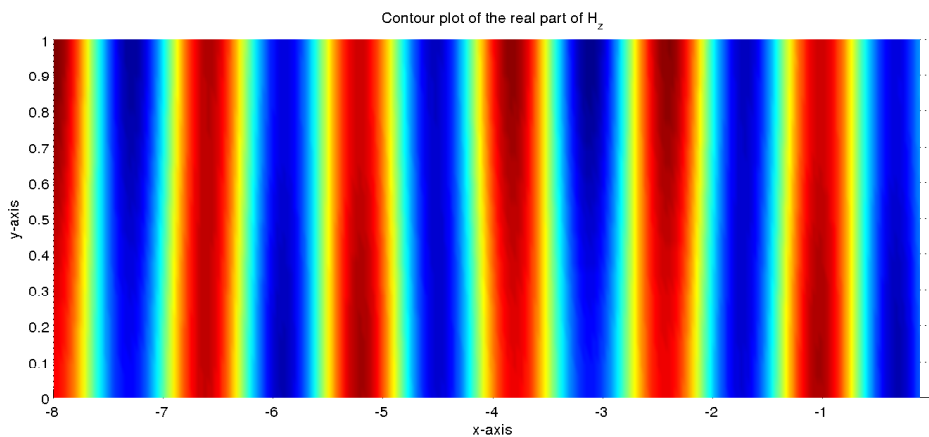


Fig. 6.1.5 Contour plot of H_z .

Section 6.2 considers the discretization method used in this chapter and discusses the node based versus the edge based finite element implementation. This chapter is ended with the conclusions based on the performed experiments.

6.2 Discretization

Equation (6.1.1) is discretized by the finite element method (FEM). The first and perhaps, the most important step in the FEM is the manner in which the domain is discretized, because this choice may affect the computer storage requirements, the computation time and the accuracy of the numerical results. In this thesis, the FEM is discretized using a so called *node* based implementation (Subsection 6.2.1) and an *edge* based implementation (Subsection 6.2.2). In Figure

6.2.1 the discretization of the two dimensional cavity with dimensions 8×1 and mesh size $h = \frac{1}{8}$ is illustrated.

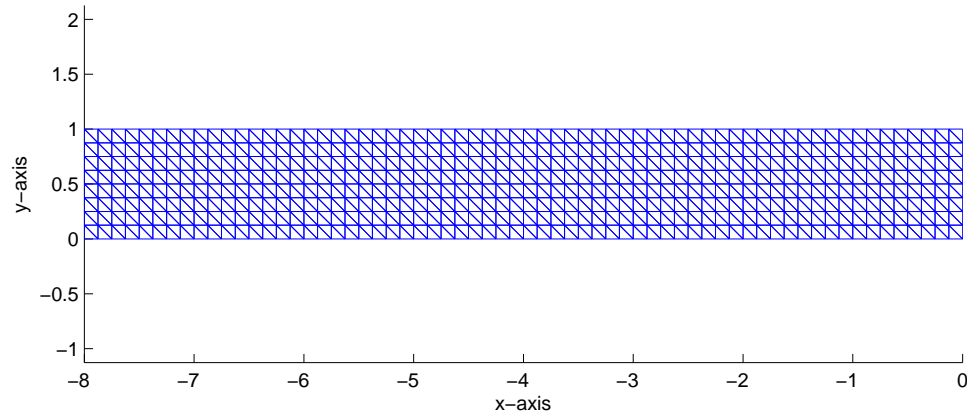


Fig. 6.2.1 The discretized two dimensional cavity with mesh size $h = \frac{1}{8}$.

6.2.1 Node based finite element discretization

When the node based FEM implementation is considered for the two dimensional vector wave equation, the domain, say Ω , is divided into a number of two dimensional triangular elements with no overlap nor gaps between elements. Each element and the nodes per element can then be labeled with separate sets of integers for identification. In this case, each triangular elements has three nodes and a so called *connectivity array* is needed to relate the three nodes to their corresponding triangle. For more about these basics of the FEM implementation, several textbooks in Numerical Methods are available. See for example Van Kan, Segal and Vermolen (ref. 12) or Jin (ref. 13).

When the domain is discretized, the unknown function must be approximated within each element. Let ϕ denote the unknown function and let e denote the element number. The approximation is denoted by:

$$\phi^e(x_1, x_2) = a^e + b^e x_1 + c^e x_2. \quad (6.2.1)$$

The constants a^e , b^e and c^e in Equation (6.2.1) can be determined using the coordinates (x_1^e, x_2^e) of the three nodes for each element². When these constants are calculated, they can be used to

²For each node i , these constants are written as a_i^e , b_i^e and c_i^e , for $i = 1, 2, 3$.

calculate the area Δ^e of an element after which the element matrix can be calculated. For Equation (6.1.1), the discretization matrix A can be written as a sum of the stiffness part and the mass part:

$$A = A_{\text{mass}} + A_{\text{stiff}}. \quad (6.2.2)$$

For each element e , A_{mass}^e and A_{stiff}^e are given by:

$$A_{\text{mass}}^e = \begin{bmatrix} A_{11}^m & A_{12}^m & A_{13}^m \\ A_{21}^m & A_{22}^m & A_{23}^m \\ A_{31}^m & A_{32}^m & A_{33}^m \end{bmatrix} \quad \text{and} \quad A_{\text{stiff}}^e = \begin{bmatrix} A_{11}^s & A_{12}^s & A_{13}^s \\ A_{21}^s & A_{22}^s & A_{23}^s \\ A_{31}^s & A_{32}^s & A_{33}^s \end{bmatrix}, \quad (6.2.3)$$

where:

$$A_{\text{mass},(ij)}^e = \frac{1}{24} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \quad \text{and} \quad A_{\text{stiff},(ij)}^e = \Delta^e (b_1^i b_1^j + c_1^i c_1^j), \quad \text{for } i, j = 1, 2, 3. \quad (6.2.4)$$

To obtain the elements from $A_{\text{mass},(ij)}^e$, the general integration rule is used (see Van Kan et al., ref. 12, p. 111). Finally, the system matrix A for the inner elements, can be obtained by *assembling* the element matrices. For a more elaborate discussion about the assembling procedure for the node based as well as the edge based implementation, the reader is referred to Jin (ref. 13).

The discretization of the different boundary conditions is treated in the corresponding sections. The next subsection briefly discusses the discretization matrix for the inner elements when the edge based implementation is considered.

6.2.2 Edge based finite element discretization

When the edge based FEM discretization is considered for two dimensional triangular elements, the system matrix can also be decomposed similar to the decomposition in Equation (6.2.2). For the stiffness part, the following element matrices are stated for an element e :

$$A_{\text{stiff},(ij)}^e = \frac{\ell_i^e \ell_j^e}{\Delta^e}, \quad \text{for } i, j = 1, 2, 3. \quad (6.2.5)$$

For the mass part, the elements of the symmetric (3×3) element matrix are stated below:

$$\begin{aligned}
 A_{mass,(11)}^e &= \frac{(\ell_1^e)^2}{24\Delta^e} (f_{22} - f_{12} + f_{11}), \\
 A_{mass,(12)}^e &= \frac{\ell_1^e \ell_2^e}{48\Delta^e} (f_{23} - f_{22} - 2f_{13} + f_{12}), \\
 A_{mass,(13)}^e &= \frac{\ell_1^e \ell_3^e}{48\Delta^e} (f_{21} - 2f_{23} - f_{11} + f_{13}), \\
 A_{mass,(22)}^e &= \frac{(\ell_2^e)^2}{24\Delta^e} (f_{33} - f_{23} + f_{22}), \\
 A_{mass,(23)}^e &= \frac{\ell_2^e \ell_3^e}{48\Delta^e} (f_{31} - f_{33} - 2f_{21} + f_{23}), \\
 A_{mass,(33)}^e &= \frac{(\ell_3^e)^2}{24\Delta^e} (f_{11} - f_{13} + f_{33}),
 \end{aligned}$$

where $f_{ij} = b_i^e b_j^e + c_i^e c_j^e$, ℓ_i^e denotes the length of edge i .

After the calculation of the element matrices, the assembling procedure can be performed to obtain the system matrix for the inner elements.

To obtain a well posed problem, non trivial boundary conditions have to be imposed on the boundary of the cavity in Figure 6.1.2. For the two dimensional cavity in this chapter, *Dirichlet* boundary conditions are imposed on the whole boundary, except on the aperture. On the aperture, *absorbing* boundary conditions are imposed. In this thesis, two types of absorbing boundary conditions are analyzed, namely the *local* absorbing boundary conditions and the *global* absorbing boundary conditions.

6.2.3 Node based FEM discretization with local absorbing boundary conditions imposed on the aperture

In this subsection, the two dimensional cavity with dimensions 8×1 is considered illustrated in Figure 6.1.2. To obtain a well posed problem, non trivial boundary conditions have to be imposed on the boundary of the cavity. On the aperture local absorbing boundary conditions are imposed. A node based FEM implementation is used to solve the system. The discretization matrix A for this model problem is forced to have the same block structure as the discretization matrix for the current application by arranging the complex valued entries due to the boundary conditions into the lower right block A_{22} . The main difference between the global absorbing boundary conditions in Chapter 3 and the local absorbing boundary conditions in this application, is the fully versus the sparsely populated block A_{22} .

The local absorbing boundary conditions imposed on the aperture of the two dimensional cavity, are stated below:

$$\frac{\partial E_z}{\partial n} + \iota k_0 E_z = 0, \quad \iota^2 = -1. \quad (6.2.6)$$

To simplify the expressions for the elements of the boundary element matrix, consider the following setup:

- Write the boundary condition as: $\frac{\partial E_z}{\partial n} + \gamma E_z = 0$.
 - Define $\gamma = \gamma_1 + \gamma_2 \frac{\partial^2}{\partial s^2}$, with
- $$\gamma_1 = \iota k_0 \text{ and } \gamma_2 = \frac{\iota}{2k_0}. \quad (6.2.7)$$

When $\gamma_2 = 0$, the *first order* local absorbing boundary condition is considered. With $\gamma_2 \neq 0$, the *second order* local absorbing boundary conditions are imposed.

Note that on the aperture, the unknown *surface elements* s are computed using a *line integral* and therefore the resulting boundary element matrix is a (2×2) -matrix with the following elements:

$$K_{11}^s = K_{22}^s = \gamma_1^s \frac{\ell^s}{3} - \frac{\gamma_2^s}{\ell^s}, \quad (6.2.8)$$

$$K_{12}^s = K_{21}^s = \gamma_1^s \frac{\ell^s}{6} + \frac{\gamma_2^s}{\ell^s}, \quad (6.2.9)$$

where γ_1^s and γ_2^s denote the average value of γ_1 and γ_2 within the s -th segment and ℓ^s denotes the length of the s -th segment.

For the boundary elements on the aperture is it assumed that the incoming incident wave has the following form:

$$E^{inc}(x_1, x_2) = e^{\iota k_0(x_1 \cos(\varphi^{inc}) + x_2 \sin(\varphi^{inc}))}. \quad (6.2.10)$$

Here φ^{inc} denotes the angle of incidence. For $i = 1, 2$, the elements f_i^s of the (2×1) -element vector are then given by:

$$f_1^s = f_2^s = \ell^s (\iota k_0 \cos(\varphi^{inc}) e^{\iota k_0(x_1^c \cos(\varphi^{inc}) + x_2^c \sin(\varphi^{inc}))} + \iota k_0 e^{\iota k_0(x_1^c \cos(\varphi^{inc}) + x_2^c \sin(\varphi^{inc}))}), \quad (6.2.11)$$

where x_1^c and x_2^c denote the centers of the edges considered.

The following aspects of the solution algorithm will be analyzed:

- The grid size dependence.
- The influence of the wavenumber.
- The influence of using either preconditioner $M_1 = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{bmatrix}$ or $M_2 = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ 0 & \tilde{A}_{22} \end{bmatrix}$.

In Table 10 the performance of the IDR(4) method is analyzed where an exact solve is performed for the preconditioner solve. The specified maximal number of iterations to perform is equal to

1000 and the tolerance is equal to 10^{-6} . The shift for the shifted Laplace preconditioner is chosen equal to $(\beta_1, \beta_2) = (1, -0.5)$. Note that for these experiments, the performance of the ML-AMG algorithm is not included. This is postponed to the next chapter.

k_0	N	2π	3π	6π
$h = \frac{1}{32}$	7937	62	123	361
$h = \frac{1}{64}$	32,257	61	128	432

Table 10 Total number of matrix vector products for the IDR(4)-method to solve a two dimensional cavity model problem with preconditioner M_1 . A node based FEM implementation is considered and the system matrix as well as the preconditioner have local absorbing boundary conditions imposed on the aperture.

k_0	N	2π	3π	6π
$h = \frac{1}{32}$	7937	70	131	363
$h = \frac{1}{64}$	32,257	79	145	440

Table 11 Total number of matrix vector products for the IDR(4)-method to solve a two dimensional cavity model problem with preconditioner M_2 . A node based FEM implementation is considered and the system matrix as well as the preconditioner have local absorbing boundary conditions imposed on the aperture.

From Tables 10 and 11 the following conclusions can be made:

- The linear solver is nearly h -independent.
- The total number of matrix vector products increases nearly linear with the wavenumber k_0 .
- There is no significant effect on the total number of matrix vector products when preconditioner M_1 is replaced by its reduced form M_2 .

6.2.4 Node based FEM discretization with global absorbing boundary conditions imposed on the aperture

In this subsection the same model problem as in the previous subsection is discussed with the local absorbing boundary conditions replaced by global absorbing boundary conditions. The global boundary conditions imposed on the two dimensional aperture of the cavity, are identical to the global boundary conditions discussed in Chapter 2. Therefore, only the elements of the (2×2) pseudo-boundary element matrix will be given. The pseudo-element matrix is computed for each combination of two elements and added to the system matrix. The pseudo-boundary element ma-

trix has four identical elements. However, the evaluation of the singularity of the Green's function must be handled with care (see Jin, ref. 13, p.416). The four elements of this matrix can be approximated by either:

$$P^{ss} = \frac{\ell}{8}(k_0\ell^s)^2 \left[1 - \frac{2\ell}{\pi} \ln(0.1638k_0\ell^s) \right] - \frac{\ell}{4}k_0\ell^s H_1^{(2)}(0.5k_0\ell^s) \quad (6.2.12)$$

or

$$P^{st} = \frac{\ell}{8}k_0^2\ell^s\ell^t H_0^{(2)}(k_0|x_s - x_t|) + \frac{\ell}{8}k_0\ell^s [\pm H_1^{(2)}(k_0|x_s - x_t - 0.5\ell^t|) \mp H_1^{(2)}(k_0|x_s - x_t + 0.5\ell^t|)], \quad x_s \geq x_t, \quad s \neq t. \quad (6.2.13)$$

Here ℓ^s denotes the length of the s -th segment with x_s being its midpoint. Furthermore, for $i \in \{0, 1\}$, $H_i^{(2)}$, denotes the i -th order Hankel function of the second kind.

The results obtained in this subsection are summarized in Table 12 with preconditioner M_1 and Table 13 with preconditioner M_2 . From these tables the same conclusion can be drawn as in from Tables 10 and 11 in the previous subsection:

- The linear solver is nearly h -independent.
- The total number of matrix vector products increases nearly linear with the wavenumber k_0 .
- There is no significant effect on the total number of matrix vector products when preconditioner M_1 is replaced by its reduced form M_2 .

Hence, there is no significant difference in the behaviour of the algorithm when it is applied to the vector wave equation using local or global absorbing boundary conditions.

k_0	N	2π	3π	6π
$h = \frac{1}{32}$	7937	62	124	275
$h = \frac{1}{64}$	32,257	62	128	350

Table 12 Total number of matrix vector products for the IDR(4)-method to solve a two dimensional cavity model problem with preconditioner M_1 . A node based FEM implementation is considered and the system matrix as well as the preconditioner have global absorbing boundary conditions imposed.

6.2.5 Use a preconditioner based on local absorbing boundary conditions for the original system with global absorbing boundary conditions imposed

In this subsection the goal is to analyze the effect of using a preconditioner with different boundary conditions compared to the boundary conditions in the original system. The model problem

k_0	N	2π	3π	6π
$h = \frac{1}{32}$	7937	70	135	298
$h = \frac{1}{64}$	32,257	75	146	427

Table 13 Total number of matrix vector products for the IDR(4)-method to solve a two dimensional cavity model problem with preconditioner M_2 . A node based FEM implementation is considered and the system matrix as well as the preconditioner have global absorbing boundary conditions imposed on the aperture.

used to perform these experiments is the same as in previous section and the same setup is used. The system matrix A with local absorbing boundary conditions is denoted by A_{loc} . When global absorbing boundary conditions are imposed, the matrix is denoted by A_{gl} . This is idem for the preconditioner M chosen equal to M_1 or M_2 .

The big advantage of using M_{loc} is that it can be solved more efficiently (e.g. using complex AMG) because of its sparse structure, as opposed to the partly fully and partly sparsely populated matrix which is obtained for M_{gl} .

Tables 14, 15, 16 and 17 summarize the results and the conclusions are listed afterward.

k_0	2π	4π	6π
(A_{loc}, M_{loc})	60	180	342
(A_{gl}, M_{loc})	75	190	283
(A_{gl}, M_{gl})	60	178	263

Table 14 Total number of matrix vector products for the IDR(4) algorithm to solve a two dimensional cavity model problem. M_1 is chosen as preconditioner in this experiment with shift $(\beta_1, \beta_2) = (1, -0.5)$ and an exact solve for the preconditioner system. The mesh size $h = \frac{1}{32}$ and the total number of unknowns $N = 7937$.

k_0	2π	4π	6π
(A_{loc}, M_{loc})	70	178	347
(A_{gl}, M_{loc})	70	200	290
(A_{gl}, M_{gl})	70	200	288

Table 15 Total number of matrix vector products for the IDR(4) algorithm to solve a two dimensional cavity model problem. M_2 is chosen as preconditioner in this experiment with shift $(\beta_1, \beta_2) = (1, -0.5)$ and an exact solve for the preconditioner system. The mesh size $h = \frac{1}{32}$ and the total number of unknowns $N = 7937$.

k_0	2π	4π	6π
(A_{loc}, M_{loc})	60	199	432
(A_{gl}, M_{loc})	79	235	412
(A_{gl}, M_{gl})	62	215	350

Table 16 Total number of matrix vector products for the IDR(4) algorithm to solve a two dimensional cavity model problem. M_1 is chosen as preconditioner in this experiment with shift $(\beta_1, \beta_2) = (1, -0.5)$ and an exact solve for the preconditioner system. The mesh size $h = \frac{1}{64}$ and the total number of unknowns $N = 32, 257$.

k_0	2π	4π	6π
(A_{loc}, M_{loc})	83	225	440
(A_{gl}, M_{loc})	79	247	379
(A_{gl}, M_{gl})	75	239	427

Table 17 Total number of matrix vector products for the IDR(4) algorithm to solve a two dimensional cavity model problem. M_2 is chosen as preconditioner in this experiment with shift $(\beta_1, \beta_2) = (1, -0.5)$ and an exact solve for the preconditioner system. The mesh size $h = \frac{1}{64}$ and the total number of unknowns $N = 32, 257$.

From these tables, the following conclusions are drawn:

- For preconditioner M_1 and variable mesh size h , there is no significant difference in the total number of matrix vector operations. For preconditioner M_2 there is relatively more influence when h is changed.
- When the mesh size is fixed and the performance with preconditioners M_1 or M_2 is compared, it can be concluded that the total number of matrix vector products is not significantly affected.
- The performance of the linear solver is almost the same when A_{gl} is preconditioned by M_{loc} with M_{loc} either M_1 or M_2 with local absorbing boundary conditions imposed. This experiment shows that indeed M_{loc} can be used to precondition A_{gl} . It is recommended to evaluate this algorithm in the three dimensional formulation which is considered in the original algorithm used by Hooghiemstra (ref. 10).

6.2.6 Edge based FEM discretization

In this subsection the same model problem as in the previous subsections is considered. The main difference is that in this subsection the edge based implementation is considered opposed to the node based implementation. For the discretization using edge based basisfunctions, an absorbing boundary condition can be formulated in the following way:

$$\begin{aligned} & \iint_S \{(\nabla \times \mathbf{E}) \cdot (\nabla \times \mathbf{E}) + k_0^2 \mathbf{E} \cdot \mathbf{E}\} dS + 2\iota k_0 \int_{\partial S} (\hat{\mathbf{n}} \times \mathbf{E}) \cdot (\hat{\mathbf{n}} \times \mathbf{E}) d\partial S = \quad (6.2.14) \\ & -2 \int_{\partial S} \mathbf{E} \cdot \mathbf{U}^{\text{inc}} d\partial S, \\ & \mathbf{U}^{\text{inc}} = \iota k_0 \hat{\mathbf{n}} \times (\hat{\mathbf{n}} \times \mathbf{E}^{\text{inc}}) + \hat{\mathbf{n}} \times (\nabla \times \mathbf{E}^{\text{inc}}). \end{aligned}$$

The basisfunctions which span the tangential electric field on the boundary ∂S are those associated with edges located on the boundary. For the zeroth order boundary functions, the basisfunctions for the magnetic current are given by:

$$\hat{\mathbf{n}} \times \mathbf{W}_i|_{\mathbf{r} \in \partial S} = \hat{\mathbf{n}} \times l_i (L_i \nabla L_j - L_j \nabla L_i) = l_i (\nabla L_i \times \nabla L_j), \quad (6.2.15)$$

where l_i denotes the length of edge i .

Therefore, the following will hold:

- Because in each boundary element of ∂S there is only a single degree of freedom with support. The boundary element matrix is therefore of order 1.

- For the regular triangulation used in this two dimensional study, the boundary elements all have the same orientation and the same length, with respect to the boundary.

It is straightforward to show that the boundary element matrix is given by:

$$K_i = 2\iota k_0 \int_{\partial S} (\hat{\mathbf{n}} \times \mathbf{W}_i) \cdot (\hat{\mathbf{n}} \times \mathbf{W}_i) d\partial S = 4\iota k_0 l_i. \quad (6.2.16)$$

The excitation with a vertically polarized plane wave travelling along a direction with angle ϕ with the boresight direction of the cavity, is given as:

$$\mathbf{E}^{\text{inc}} = \mathbf{E}_0 e^{\iota k_0 (x \cos \phi + y \sin \phi)}, \quad (6.2.17)$$

and the element vector is now given as

$$f_i = -\iota k_0 (\mathbf{E}_0)_y l_i. \quad (6.2.18)$$

In Tables 18 and 19 the results for the IDR(4) method are summarized.

k_0	N	2π	3π	6π
$h = \frac{1}{32}$	24.321	93	213	–
$h = \frac{1}{64}$	97.793	95	230	793

Table 18 Total number of matrix vector products for the IDR(4)-method to solve a two dimensional cavity model problem with preconditioner M_1 . An edge based FEM implementation is considered and the system matrix as well as the preconditioner have local absorbing boundary conditions imposed on the aperture.

k_0	N	2π	3π	6π
$h = \frac{1}{32}$	24.321	108	262	–
$h = \frac{1}{64}$	97.793	114	285	939

Table 19 Total number of matrix vector products for the IDR(4)-method to solve a two dimensional cavity model problem with preconditioner M_2 . An edge based FEM implementation is considered and the system matrix as well as the preconditioner have local absorbing boundary conditions imposed on the aperture.

From these tables the same conclusions can be made as the node based FEM implementation with identical boundary conditions.

6.2.7 Edge based FEM implementation with global absorbing boundary conditions imposed on the aperture

In the previous sections it was concluded that the change of local absorbing boundary conditions to global boundary conditions showed no significant difference in the performance of the linear solver. Therefore, it is expected that this is also the case in the edge based implementation. Therefore, this model problem is not investigated in this thesis.

6.3 Conclusions

Based on the experiments performed in this chapter, the following conclusions are listed for the iterative solution method for the two dimensional vector wave equation:

- There is no significant impact on the performance of the linear solver when preconditioner M_1 is compared to preconditioner M_2 .
- The total number of matrix vector operations is not significantly affected when the local absorbing boundary conditions are compared to the global boundary conditions.
- When the original system with *global* absorbing boundary conditions is preconditioned using a preconditioner based on *local* absorbing boundary conditions, there is almost no effect on the performance of the linear solver. Therefore, it is recommended to analyze if this is also the case in the original three dimensional algorithm used by Hooghiemstra (ref. 10). Additionally, implementing a preconditioner based on local absorbing boundary conditions is relatively easy to realize in the current three dimensional solver: the evaluation of the matrix $[P^{st}]$ in Chapter 2 has to be performed using the result in Equation (6.2.15).

Chapter 7 will analyze the second proposed algorithm in this thesis: *the IDR(4)–ML-AMG algorithm*. The performance of this algorithm will be compared to the first proposed Bi-CGSTAB–ML-AMG algorithm for the same model problems seen in Chapter 3 and to the nested GCR algorithm of Hooghiemstra (ref. 10) to assess the improvement the new method offers over the existing algorithm.

7 Progress evaluation

7.1 Model problems

In this chapter the performance of the newly proposed *IDR(4)–ML-AMG algorithm* is analyzed for several model problems. In the first subsection this algorithm is tested on the model problems seen in Subsection 3.4.1. The second subsection considers a three dimensional small cavity scattering problem and compares the performance of this algorithm to the algorithm of Hooghiemstra (ref. 10) to assess the progress made in this project. For these model problems the preconditioners share the same boundary conditions as the system matrix, namely the global absorbing boundary conditions.

7.1.1 Small cavity scattering model problem

In this subsection the same model problem is considered which was introduced in Subsection 3.4.1. The cavity for this small cavity scattering problem has dimensions $0.6\lambda \times 1.5\lambda \times 1.5\lambda$. The wavenumber k_0 is chosen equal to 2π and therefore $\lambda = 1$. The Krylov subspace method used here is the IDR(4) method and the ML-AMG algorithm is used to solve the shifted Laplace preconditioner system with optimal real shift. The maximal number of iterations to perform is equal to 1000 and the specified tolerance is $(\frac{\|r^k\|_2}{\|b\|_2}) \leq 10^{-6}$. For multigrid the following parameters are specified: one full MG cycle is applied, the Aztec smoother¹ is chosen and no restriction on the number of grid levels.

Until now only zeroth order basis functions were considered in Chapter 6. In this subsection the behaviour of the solution method will also be investigated for higher order basis functions used in the three dimensional cavity scattering model problems seen in Chapter 3.

In the experiments performed in this chapter, several parameters are varied. The order p of the basisfunctions is varied for a fixed mesh size h and vice versa. The results obtained for the optimal *real* shift $(\beta_1, \beta_2) = (-1, 0)$ are compared to results obtained for the optimal *complex* shift $(\beta_1, \beta_2) = (1, -0.5)$. For the latter shift, an *exact* preconditioner solve is performed. For the optimal real shift, the exact solve for the preconditioner is compared to the ML-approximation of the inverse of the preconditioner.

In Table 20 the results are summarized when preconditioner M_1 is used. Note that in this table, the total number of matrix vector operations are summarized. For the IDR(s) methods, this number is equal to the total number of iterations. In Table 21, the same setting is used as in Table 20 except that now M_2 is used as preconditioner.

¹See the ML-guide (ref. 6) for more information about the Aztec smoother.

p	h	N	$(\beta_1, \beta_2) = (-1, 0)$	$(\beta_1, \beta_2) = (1, -0.5)$	ML-solve for $(\beta_1, \beta_2) = (-1, 0)$
0	0.25	1402	285	58	305
0	0.20	2796	369	54	353
1	0.35	2914	436	45	407
1	0.30	4344	610	47	433
1	0.25	7960	907	45	538
2	0.40	5316	925	40	539
2	0.35	8730	1371	40	862

Table 20 Total number of matrix vector products for the IDR(4)–ML-AMG algorithm to solve a small cavity model problem. M_1 is chosen as preconditioner in this experiment.

p	h	N	$(\beta_1, \beta_2) = (-1, 0)$	$(\beta_1, \beta_2) = (1, -0.5)$	ML-solve for $(\beta_1, \beta_2) = (-1, 0)$
0	0.25	1402	245	150	308
0	0.20	2796	301	226	374
1	0.35	2914	343	270	394
1	0.30	4344	342	427	417
1	0.25	7960	413	459	530
2	0.35	5316	390	477	546
2	0.30	8730	473	494	833

Table 21 Total number of matrix vector products for the IDR(4)–ML-AMG algorithm to solve a small cavity model problem. M_2 is chosen as preconditioner in this experiment.

When Tables 20 and 21 are compared, the following conclusions can be drawn:

- The performance for M_1 and M_2 is almost similar for the optimal *real* shift and for the zeroth order basis functions.
- When the optimal *complex* shift is used, the total number of matrix vector products is nearly constant for preconditioner M_1 for all the type of basis functions considered here.
- When the optimal complex shift is used, the block upper triangular preconditioner M_2 leads to a higher number of matrix vector operations for the IDR(4)–ML-AMG algorithm, compared to preconditioner M_1 . Furthermore, it seems that for increasing order of the basis functions, the performance deteriorates.
- For the multigrid performance it can be concluded that for the zeroth and first order basis-function, the h -independency is maintained.
- For the zeroth order basis functions a similar performance is noted when the *exact* solve for both preconditioners is compared to the *ML* solve. For the higher order basis functions considered here, the total number of matrix vector multiplications with the ML solve is smaller compared to the case when an exact solve is performed.

When the results from Table 21 are compared with the performance of the Bi-CGSTAB method in Table 2, the following can be concluded:

- With the restriction on the pure real shift and for first order basis functions, Bi-CGSTAB did not converge within the specified maximum number of iterations (1000). However, IDR(4) converges for this optimal real shift within the specified number of iterations.
- When the optimal complex shift is used, the IDR(4)–ML-AMG algorithm performs less matrix vector products compared to the Bi-CGSTAB–ML-AMG algorithm in Chapter 3.

To assess the progress made in this project, the next subsection analyzes the performance of the IDR(4)–ML-AMG algorithm compared to the performance of the nested GCR algorithm of Hooghiemstra (ref. 10) using a cavity scattering model problem of intermediate size.

7.1.2 Cavity scattering model problem of intermediate size

In this subsection a cavity scattering model problem is considered where the dimensions of the cavity are given as $4\lambda \times 1.5\lambda \times 1.5\lambda$. The wavenumber k_0 is equal to 2π and hence $\lambda = 1$. For this experiment, zeroth order basis functions are used. For higher order basis functions, the model problem could not be loaded in Matlab. The discretization contains 71,479 elements and 79,428 degrees of freedom N for mesh size $h = 0.10$. The goal here is to compare the total number of matrix vector products which are performed and not the CPU times needed.

The performance of the nested GCR algorithm of Hooghiemstra (ref. 10) is tested on a nearly identical model problem as considered in this subsection. This algorithm uses GCR for both

the preconditioner and the preconditioned system. The average number of inner GCR iterations is denoted by $\bar{I}t_{in}$ and the outer GCR iterations are denoted by $I t_{out}$. The inner iterations are needed to perform the preconditioner solve with specified tolerance $\varepsilon = 10^{-3}$. The maximum number of iterations to perform, is equal to 1000 and the outer tolerance is given by $\frac{\|r^k\|_2}{\|b\|_2} < 10^{-4}$. For this model problem the total number of unknowns N is equal to 79,266. This nested GCR algorithm performed optimal for shift $(\beta_1, \beta_2) = (0.5, 3.0)$ with the following results²:

- $I t_{out} = 896$. For the outer GCR iterations, the total number of vector updates is approximated by $900^2 = 810,000$. When it is assumed that 30 vector updates correspond with 1 matrix vector operation, this results in about $\frac{810,000}{30} = 27,000$ matrix vector operations for the outer GCR iteration.
- $\bar{I}t_{in} = 97.18$. For the inner GCR iterations, the total number of vector updates is approximated by $100^2 = 10,000$, which results in about $\frac{10,000}{30} \approx 340$ matrix vector operations. As this number of matrix vector operations is performed for each outer iteration, the total number of performed matrix vector products due to inner iterations is estimated by $900 * 340 = 306,000$.
- **Conclusion:** a rough estimate of the total number of matrix vector operations for the nested GCR algorithm is equal $900 * 100 + 306,000 + 27,000 = 90,000 + 306,000 + 27,000 = 423,000$.

To analyze the performance of the IDR(4)–ML-AMG algorithm, the optimal real shift is used and the preconditioner solve is performed by ML. For this experiment, the specified tolerance is chosen equal to the tolerance Hooghiemstra (ref. 10) used for his outer GCR iterations, namely $\frac{\|r^k\|_2}{\|b\|_2} < 10^{-4}$. With this setup, the following results are obtained for the new algorithm:

- Total number iterations for IDR(4) = 513. For each iteration, IDR(4) performs 10 vector updates. Therefore, the total number of matrix vector operations for this algorithm is approximated by $\frac{513 \cdot 10}{30} = 171$.
- One full multigrid V-cycle roughly estimated by 10 matrix vector operations.
- **Conclusion:** a rough estimate of the total number of matrix vector operations for the IDR(4)–ML-AMG algorithm with optimal real shift is approximately $513 * 10 + 171 = 5307$. *This corresponds with a gain in the total number of matrix vector multiplications with a factor approximately equal to $\frac{423,000}{5307} \approx 80$.*

²In obtaining these results, the matrix vector products necessary for the evaluation of the inner products during the solution process are neglected. Furthermore, note that this optimal shift is used because of the different method used.

7.1.3 Expected performance of the IDR(4)–ML-AMG algorithm with complex AMG

In this subsection some extra experiments are performed to analyze the performance of the IDR(4)–ML-AMG algorithm for the three dimensional cavity used in the previous subsection with the optimal *complex* shift. In Table 22 these results are summarized using the same setup used in previous subsection with the exception that now $\frac{\|r^k\|_2}{\|b\|_2} < 10^{-6}$.

preconditioner	ML solve $(\beta_1, \beta_2) = (-1, 0)$	$(\beta_1, \beta_2) = (1, -0.5)$
M_1	–	85
M_2	782	1142

Table 22 Total number of matrix vector products for the IDR(4)–ML-AMG algorithm for a cavity model problem of intermediate size. The model problem has dimensions $4\lambda \times 1.5\lambda \times 1.5\lambda$ and $N = 79,428$.

Based on this experiment it is expected that a complex AMG solver will improve the results obtained in the previous subsection. From earlier experiments it was concluded that there was no significant difference in the total number of matrix vector operations when an exact preconditioner solve was compared to the ML preconditioner solve. Therefore, it is expected that a complex AMG solver will result in approximately the same number of matrix vector operations as is obtained for the exact solve, namely 85. The gain obtained for the optimal complex shift compared to the results in previous subsection for the optimal real shift, is expected to be:

- Total number iterations for IDR(4) = 85. Therefore, the total number of matrix vector operations for this algorithm is approximated by $\frac{85 \cdot 10}{30} \approx 30$.
- One full multigrid V-cycle is roughly estimated by 10 matrix vector operations.
- **Conclusion:** a rough estimate of the total number of matrix vector operations for the IDR(4)–ML-AMG algorithm with optimal complex shift is approximately $513 * 10 + 30 = 543$.

This corresponds with a gain in the total number of matrix vector multiplications with a factor approximately equal to $\frac{423,000}{543} \approx 800$.

In Figure 7.1.1 the CPU time is illustrated as a function of the number of degrees of freedom on the *aperture* for the nested GCR algorithm used by Hooghiemstra (ref. 10) and the frontal solver for a similar model problem. For this model problem the total number of degrees of freedom on the aperture is approximately 650. When the performance of the nested GCR algorithm is compared to the performance of the frontal solver for this number, it can be concluded that the frontal solver outperforms the nested GCR algorithm.

Based on the before mentioned improvement ratios of the IDR(4)–ML-AMG algorithm with respect to the original algorithm of Hooghiemstra (ref. 10), it is expected that when the optimal

real shift is used, the performance of the IDR(4)–ML-AMG algorithm is similar to the performance of the frontal solver. When the optimal complex shift is considered, it is expected that the IDR(4)–ML-AMG algorithm will show a relatively better performance compared to the frontal solver.

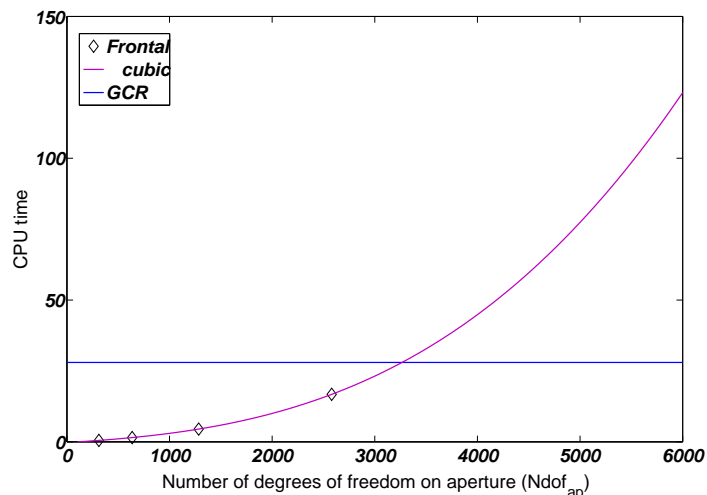


Fig. 7.1.1 The CPU time as a function of the number of degrees of freedom on the aperture for the preconditioned GCR method and the frontal solver for a model problem with dimensions $4\lambda \times 1.5\lambda \times 1.5\lambda$ and $N = 79,266$.

The last remark made here considers the gain in *memory*. For this estimate only the outer loop is considered. The memory needed to perform the preconditioner solve is neglected as well as the memory needed to store the system matrix (the system matrix has to be stored for both methods). When the nested GCR algorithm is used, 900 outer iterations are necessary. This corresponds with storing about $900 \cdot 2 = 1800$ vectors.

When the IDR(4) method is used, about $4.2 = 8$ vectors are required. This results in an approximated gain in memory of about $\frac{1800}{8} = 225$.

7.2 Conclusions

Based on these results the following can be concluded when the nested GCR algorithm of Hooghiemstra (ref. 10) is compared to the IDR(4)–ML-AMG algorithm:

- With the restriction to real valued arithmetic, the IDR(4)–ML-AMG algorithm leads to a reduction in the total number of matrix vector multiplications by factor 80.
- Using algebraic multigrid to solve the shifted Laplace preconditioner system leads to a constant preconditioner system. Therefore, the *long recurrence* GCR method is replaced by the *short recurrence* IDR(4) method. Using this short recurrence method, there is no need to store the complete Krylov basis during the entire solution process for the orthogo-

nalization procedure. This leads to a dramatic improvement of the storage requirement and the CPU time.

- When the optimal complex shift is used with preconditioner M_1 , the total number of matrix vector products is relatively small: 85 to solve a system with $N = 79,428$. When M_2 is used, the performance of the linear solve is worse. It seems that using the upper triangular preconditioner M_2 leads to no improvement in the spectrum of the preconditioned system.
- In Chapter 6 it was concluded that the original system with global absorbing boundary conditions can be preconditioned using a preconditioner based on local absorbing boundary conditions. When this is combined with the fact that using M_1 with the optimal complex shift, results in 85 matrix vector operations, it is expected that a complex algebraic multigrid solver will significantly improve the performance of the IDR(4)–ML-AMG algorithm. Compared to the nested GCR algorithm of Hooghiemstra (ref. 10), the total number of matrix vector multiplications will be reduced by factor 800.
- When the GCR algorithm is compared to the IDR(4) method, the expected gain in memory is approximately 225.

8 Conclusions

In this thesis an algebraic multigrid solution method is considered in order to accelerate the solution of the discretized vector wave equation. This equation is discretized by the finite element discretization method, using tetrahedral elements and higher order *vector* based basis functions. This results in a linear system, where the system matrix has a very unfavorable spectrum, is ‘nearly’ symmetric but not Hermitian, partly sparse and partly fully populated. From the analysis of the model problems studied, the following conclusions can be drawn:

- **Use the ML-AMG algorithm.**

Algebraic multigrid can be used effectively to accelerate the solution of the linear system stemming from the discretization of the Maxwell vector wave equation, by indirect application for the solution of the shifted Laplace preconditioner system.

- **Do not use the optimal real shift.**

The optimal *real* valued shift will improve the spectrum of the preconditioned system, but not to the same degree as the optimal *complex* valued shift. In the latter case, the convergence of the linear solver is optimal. There is a significant decrease in the performance of the linear solver when the optimal real valued shift is used compared to the case when the optimal complex shift is used.

- **Do not use the block upper preconditioner.**

To understand the effect of different preconditioners on the convergence of the linear solver, the two dimensional vector wave equation is analyzed. For this two dimensional case two preconditioners are analyzed. The first preconditioner is analogous to the preconditioner Erlangga (ref. 6) used. The second preconditioner is a *block upper* preconditioner. For this model problem it can be concluded that the total number of matrix vector operations is similar for both preconditioners. However, in the three dimensional case, using the block upper triangular matrix results in a significant difference in the total number of matrix vector operations performed. The performance of the linear solver deteriorates in this case.

- **Use IDR.**

In this thesis two Krylov subspace methods are analyzed: the Bi-CGSTAB method and the IDR(4) method. For these linear solvers the following can be concluded:

1. The Bi-CGSTAB method locally minimizes the residuals and this process is stagnated when the pure real shift is used. This stagnation occurs because in this case the eigenvalues of the preconditioned system are still complex valued. More specifically: the eigenvalues have a imaginary part which is relatively large compared to the real part.
2. The IDR(4) method also minimizes the residuals locally, but with an improvement in the minimization procedure. The result is that for the model problems considered in this thesis, no stagnation of the IDR(4) method occurred.

- **Use a preconditioner based on local absorbing boundary conditions.**

For the two dimensional vector wave equation it was analyzed how the total number of matrix vector operations was affected when the boundary conditions of the preconditioner are changed. The original system has global boundary conditions imposed and for the preconditioner local absorbing boundary conditions were prescribed. In this case it can be concluded that there is no significant impact on the performance of the linear solver.

When the IDR method is used combined with an complex algebraic multigrid package for the preconditioner solve, the total number of matrix vector products will be decreased with a factor 1000. The expected gain in memory is approximately 225.

9 Future research

In this thesis a start has been made to investigate the effect of algebraic multigrid to accelerate the solution of the linear system stemming from the discretization of the Maxwell vector wave equation, by indirect application for the solution of the shifted Laplace preconditioner system. To further improve the efficiency of the existing algorithm, the following recommendations for future research are made:

- **Inclusion of an algebraic multigrid solver which is able to perform calculations in complex valued arithmetic.**

Based on the performance of the IDR(4)–ML-AMG algorithm it was concluded that when the pure real shift is used for the shifted Laplace preconditioner, the ML performance is as expected. The total number of iterations is relatively constant compared to an exact solve for the preconditioner solve. However, the CPU times for the ML solve are smaller compared to the CPU times needed to perform the exact solve. As the pure real shift is not the optimal choice for the vector wave equation, it is expected that when the optimal complex shift is used, a great gain in the CPU time and in the total number of matrix vector multiplications can be obtained compared to the optimal real valued shift.

- **Fortran compatible version of the IDR(s)-algorithm.**

In this thesis it is concluded that using IDR(4) as Krylov subspace method results in a significant improvement of the performance, compared to the Bi-CGSTAB method in two as well as three dimensional problems. Therefore, it is advisable to implement a Fortran version of the IDR(s)-method in the existing cavity scattering solver.

- **Use a preconditioner based on local absorbing boundary conditions.**

For the two dimensional vector wave equation it was concluded that when the original system with *global* absorbing boundary conditions is preconditioned by a preconditioner based on *local* absorbing boundary conditions, the total number of matrix vector operations was not significantly affected. The solution of the preconditioner based on local absorbing boundary conditions can be performed much more efficiently than is the case when global absorbing boundary conditions are used: there is no blockstructure and the preconditioner is sparsely populated. Therefore, it is advisable to implement local absorbing boundary conditions in the preconditioner system for the vector wave equation. This is relatively easy to accomplish (see Appendix H).

- **Reduce the bandwidth of the system matrix.**

If it is no option to use a preconditioner based on local absorbing boundary conditions, it is recommended to use the upper triangular block preconditioner with a reduced bandwidth of the system matrix. When the bandwidth is minimized, less information is left out when the lower left block of the preconditioner is neglected. In this case, it seems that the convergence behaviour of the Krylov solver is not dramatically affected compared to the case when the lower left block is included in the preconditioner solve.

References

1. S.M.F. Abdoel. Multigrid acceleration of a preconditioned Krylov method for the solution of the discretized vector wave equation. Report NLR-TR-2008-282, NLR, 2008.
2. C. Vuik and C.W. Oosterlee. Lecture notes: Scientific Computing. Technical report, TU DELFT, 2005.
3. C.A. Balanis. *Advanced Engineering Electromagnetics*. John Wiley and Sons, 1989.
4. Duncan R. van der Heul, Harmen van der Ven and Jan-Willen van der Burg. Full Wave Analysis of the Influence of the Jet Engine Air Intake on the Radar Signature of Modern Fighter Aircraft. *ECCOMAS CFD*, 2006.
5. E.F. Knott, J.F. Shaeffer and M.T. Tuley. *Radar Cross Section*. Artech House, Inc., 1985.
6. Yogi Ahmad Erlangga. A robust and efficient iterative method for the numerical solution of the Helmholtz equation. Thesis, TU DELFT, 2005.
7. M.W. Gee, C.M. Siefert, J.J. Hu, R.S. Tuminaro, and M.G. Sala. ML 5.0 Smoothed Aggregation User's Guide. Technical Report SAND2006-2649, Sandia National Laboratories, 2006.
8. M.H. Gutknecht. Variants of BiCGStab for matrices with complex spectrum. *IPS Research*, 91-14, 1991.
9. P.B. Hooghiemstra. Full Wave Analysis of the Contribution to the Radar Cross Section of the Jet Engine Air Intake of a Fighter Aircraft. Report NLR-TR-2007-310, NLR, 2007.
10. P.B. Hooghiemstra. The nested generalized conjugate residual method with shifted Laplace preconditioning for the solution of the finite element discretization of the vector wave equation. Report NLR-TR-2007-741, NLR, 2007.
11. Z. Lou J. Jin, J. Lui and C.S.T. Liang. A fully high-order-finite-element simulation of scattering by deep cavities. *IEEE Trans. Magn.*, 51(9):2420–2429, 2003.
12. J. van Kan, A. Segal, F. Vermolen. *Numerical Methods in Scientific Computing*. VSSD, 2008.
13. J. Jin. *The Finite Element Method in Electromagnetics*. Wiley & sons, 1993.
14. Scott P. Maclachlan and Cornelis W. Oosterlee. Private conversation.
15. Y.A. Erlangga M.B. van Gijzen and C. Vuik. Spectral analysis of the discrete helmholtz operator preconditioned with a shifted laplacian. *SIAM J. Sci. Comput.*, 29(5):1942–1958, 2007.
16. Andrew F. Peterson Robert D. Graglia, Donald R. Wilton. Higher Order Interpolatory Vector Bases for Computational Electromagnetics. *Journal of Aircraft*, 1997.

17. Gerard L.G. Sleijpen and Diederik R. Fokkema. BICGSTAB(L) for linear equations involving unsymmetric matrices with complex spectrum. *Electronic Transactions on Numerical Analysis*, 1:11–32, 1993.
18. P. Sonneveld. CGS, a fast Lanczos-type solver for nonsymmetric linear systems. *SIAM J. Sci. Stat. Comp.*, 10:36–52, 1989.
19. Peter Sonneveld and Martin B. van Gijzen. IDR(s): A Family of Simple and Fast Algorithms for Solving Large Non-Symmetric Linear System. 07(07), 2007.
20. Y. Saad and M.H. Schultz. *GMRES: a generalized minimal residual algorithm for solving non-symmetric linear systems*. SIAM, Philadelphia, 1986.

Appendix A Electromagnetic quantities

In this appendix the basic SI (International System of Units) are discussed. In Table 23 the quantities used in Chapter 2 are given with their units and corresponding SI units.

In Chapter 2 also the vacuum values ε_0 and μ_0 were introduced. Their values are given by:

$$\mu_0 = 4\pi * 10^{-7} \frac{F}{m} \quad \text{and} \quad \varepsilon_0 = \frac{1}{c^2 \mu_0} = 8.8542 * 10^{-12} \frac{Wb}{Am},$$

where c is the speed of light with value $c = 2.9979 * 10^8 \frac{m}{s}$

Quantity	Name	Units	SI units
ε	permittivity	$\left[\frac{\text{farads}}{m} \right]$	$[kg^{-1}m^{-3}A^2s^4]$
μ	permeability	$\left[\frac{\text{henry}}{m} \right]$	$[kgms^{-2}A^{-2}]$
σ	conductivity	$\left[\frac{\text{siemens}}{m} \right]$	$[kg^{-1}m^{-3}s^3A^2]$

Table 23 List of quantities with their units and SI units.

Appendix B Useful definitions and fundamental relations

In this appendix some useful definitions and important relations are recalled.

(Vector) Inner product

An inner product on a (complex) vector space \mathbb{X} is any mapping s from $\mathbb{X} \times \mathbb{X}$ into \mathbb{C} ,

$$x, y \in \mathbb{X} \rightarrow s(x, y) \in \mathbb{C},$$

that satisfies the following conditions:

1. $s(x, y)$ is *linear* with respect to x :

$$s(\lambda_1 x_1 + \lambda_2 x_2, y) = \lambda_1 s(x_1, y) + \lambda_2 s(x_2, y) \quad \forall x_1, x_2 \in \mathbb{X}, \forall \lambda_1, \lambda_2 \in \mathbb{C}$$

2. $s(x, y)$ is *Hermitian*:

$$s(y, x) = \overline{s(x, y)} \quad \forall x, y \in \mathbb{X}$$

3. $s(x, x)$ is *positive definite*:

$$s(x, x) \geq 0 \quad \text{and } s(x, x) = 0 \text{ iff } x = 0$$

An inner product will be denoted by: (\cdot, \cdot)

Vector norm

A vector norm on a vector space \mathbb{X} is a real-valued function $x \rightarrow \|x\|$ on \mathbb{X} that satisfies the following three conditions:

1. $\|x\| \geq 0 \quad \forall x \in \mathbb{X} \quad \text{and } \|x\| = 0 \text{ iff } x = 0$
2. $\alpha \|x\| = |\alpha| \|x\| \quad \forall x \in \mathbb{X} \quad \forall \alpha \in \mathbb{C}$
3. $\|x + y\| \leq \|x\| + \|y\| \quad \forall x, y \in \mathbb{X} \quad (\text{triangle inequality})$

Hölder p-norms

The most commonly used vector norms in numerical linear algebra are special cases of the Hölder norms:

$$\|x\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}$$

These special cases are $p = 1, 2$ or $p = \infty$:

$$\|x\|_1 = |x_1| + |x_2| + \dots + |x_n|$$

$$\|x\|_2 = [|x_1|^2 + |x_2|^2 + \dots + |x_n|^2]^{\frac{1}{2}}$$

$$\|x\|_\infty = \max_{i=1, \dots, n} |x_i|$$

Matrix norms

For a general matrix $A \in \mathbb{C}^{n \times m}$ the following is defined:

$$\|A\|_{pq} = \sup_{x \in \mathbb{C}^m, x \neq 0} \frac{\|Ax\|_p}{\|x\|_q}$$

Subspaces

A subspace of \mathbb{C}^n is a subset of \mathbb{C}^n that is also a complex vector space. The set of all linear combinations of a set of vectors G of \mathbb{C}^n is a vector subspace called the *linear span* of G .

Two important subspaces that are associated with a matrix $A \in \mathbb{C}^{n \times n}$ are its:

- $\text{Ran}(A) = \{Ax | x \in \mathbb{C}^m\}$
- $\text{Ker}(A) = \{x \in \mathbb{C}^m | Ax = 0\}$

Remark The range of A is equal to the linear span of its columns.

Fundamental relation I

$$\mathbb{C}^n = \text{Ran}(A) \oplus \text{Ker}(A^T) \quad (\text{FR I})$$

Projector A projector P is any linear mapping from \mathbb{C}^n to itself that is idempotent:

$$P^2 = P$$

Fundamental relation II

If P is a projector, then so is $(I - P)$ and the following relation holds:

$$\text{Ker}(P) = \text{Ran}(I - P) \quad (\text{FR II})$$

and the following two important properties:

- the two subspace $\text{Ker}(P)$ and $\text{Ran}(P)$ intersect only at the element zero
- $\mathbb{C}^n = \text{Ker}(P) \oplus \text{Ran}(P)$

A_h -orthogonal

A_h -orthogonality is denoted by $(\cdot, \cdot)_{A_h}$ and is defined as

$$(x, y)_{A_h} := (A_h x, y) \quad \text{for } x, y \in \mathbb{C}^n$$

Energy norm

When a matrix B is symmetric and positive definite, the mapping

$$x, y \rightarrow (x, y)_B := (Bx, y)$$

from $\mathbb{C}^n \times \mathbb{C}^n$ to \mathbb{C} is a proper inner product on \mathbb{C}^n .

The associated norm is referred to as the *energy norm* or *B-norm*:

$$\|\cdot\|_B := \sqrt{(x, y)_B}$$

Rayleigh quotient

An eigenvalue λ of any matrix A satisfies the relation

$$\lambda = \frac{(Au, u)}{(u, u)} \tag{B.0.1}$$

where u is an associated eigenvector.

Define the (complex) scalars $\mu(x)$ as

$$\mu(x) = \frac{(Ax, x)}{(x, x)} \tag{B.0.2}$$

for any nonzero vector $x \in \mathbb{C}^n$.

The ratios in (B.0.1) and (B.0.2) are called *Rayleigh quotients*.

A small Rayleigh quotient implies that a vector v is a linear combination of the eigenvectors of A with smallest eigenvalues.

The set of all possible Rayleigh quotients is bounded by the 2-norm of A :

$$|\mu(x)| \leq \|A\|_2, \forall x \in \mathbb{C}^n$$

and is called *the field of values of A*.



Appendix C Sandia's multilevel preconditioning package

C.1 General application of ML

In this appendix some properties of Sandia's main multigrid preconditioning package, *ML*, are discussed. For more details, the reader is referred to the *ML*-guide (ref. 7).

ML is designed to solve large sparse linear systems of equations arising primarily from elliptic PDE discretizations. *ML* is used to define and build multigrid solvers and preconditioners, and it contains black-box classes to construct highly-scalable smoothed aggregation preconditioners. *ML* preconditioners have been used on thousands of processors for a variety of problems, e.g. the incompressible Navier-Stokes equations with heat and mass transfer, linear and nonlinear elasticity equations, the Maxwell equations and semiconductor equations.

ML can also be used as a framework to generate new multigrid methods. Using *ML*'s internal aggregation routines and Galerkin products, it is possible to focus on new types of inter-grid transfer operators without having to address the cumbersome aspects of generating an entirely new parallel algebraic multigrid code. This flexibility can be used to produce special multilevel methods using coarse grid finite element functions to serve as inter-grid transfers.

The primary goal of the developers at Sandia has been to provide state-of-the-art iterative methods that perform well on parallel computers (applications on over 3000 processors have been run) and that at the same time are easy to use for application engineers. In addition to providing algebraic multilevel methods to engineers, the *ML* library is also used in ongoing research on preconditioners.

C.2 ML in the current application

In the current application the system matrix used for computational issues, is complex valued due to the boundary conditions, complex valued material properties (when absorbing coating is considered) or a complex valued shift in the shifted Laplace preconditioner. Therefore, something must be said about using *ML* for complex valued arithmetic. In Subsection 3.3.2 it is explained how *ML* is used in the algorithm presented in this thesis.

Another way of how *ML* could be used is by using the so called *equivalent real formulation of a complex valued system*. For more about this formulation, the reader is referred to Chapter 4 from Abdoel (ref. 1) and the *ML*-guide (ref. 7).

Appendix D Multigrid appendix

In this appendix the V and W cycles are illustrated together with the coarse grid correction scheme, the two-grid cycle and the multigrid cycle. For a detailed discussion about these schemes, the reader is referred to Chapter 4 from Abdoel (ref. 1).

D.1 Coarse grid correction scheme, two-grid and multigrid cycle

Coarse grid correction scheme $u_h^m \rightarrow u_h^{m+1}$

- | | |
|--|-----------------------------------|
| - Compute the defect | $d_h^m = f_h - L_h u_h^m$ |
| - Restrict the defect (fine-to-coarse transfer) | $d_H^m = I_h^H d_h^m$ |
| - Solve on Ω_H | $L_H \hat{v}_H^m = d_H^m$ |
| - Interpolate the correction (coarse-to-fine transfer) | $\hat{v}_h^m = I_H^h \hat{v}_H^m$ |
| - Compute a new approximation | $u_h^{m+1} = u_h^m + \hat{v}_h^m$ |

Two-grid cycle $u_h^{m+1} = \text{TGCYCLE}(u_h^m, L_h, f_h, \nu_1, \nu_2)$

1. Presmoothing

- Compute \bar{u}_h^m by applying $\nu_1 \geq 0$ steps of a given smoothing procedure (e.g. Jacobi or Gauss-Seidel) to u_h^m :

$$\bar{u}_h^m = \text{SMOOTH}^{\nu_1}(u_h^m, L_h, f_h)$$

2. Coarse grid correction

- | | |
|--|---|
| - Compute the defect | $\bar{d}_h^m = f_h - L_h \bar{u}_h^m$ |
| - Restrict the defect (fine-to-coarse transfer) | $\bar{d}_H^m = I_h^H \bar{d}_h^m$ |
| - Solve on Ω_H | $L_H \hat{v}_H^m = \bar{d}_H^m$ |
| - Interpolate the correction (coarse-to-fine transfer) | $\hat{v}_h^m = I_H^h \hat{v}_H^m$ |
| - Compute the corrected approximation | $u_h^{m, \text{after CGC}} = \bar{u}_h^m + \hat{v}_h^m$ |

3. Postsmoothing

- Compute u_h^{m+1} by applying $\nu_2 \geq 0$ steps of the given smoothing procedure to $u_h^{m, \text{after CGC}}$:

$$u_h^{m+1} = \text{SMOOTH}^{\nu_2}(u_h^{m, \text{after CGC}}, L_h, f_h)$$

Multigrid cycle $u_k^{m+1} = \text{MGCYCLE}(k, \gamma, u_k^m, L_k, f_k, \nu_1, \nu_2)$

1. Presmoothing

- Compute \bar{u}_k^m by applying $\nu_1 \geq 0$ smoothing steps to u_k^m :

$$\bar{u}_k^m = \text{SMOOTH}^{\nu_1}(u_k^m, L_k, f_k)$$

2. Coarse grid correction

- Compute the defect $\bar{d}_k^m = f_k - L_k \bar{u}_k^m$
- Restrict the defect (fine-to-coarse transfer) $\bar{d}_{k-1}^m = I_k^{k-1} \bar{d}_k^m$
- Compute an approximate solution \hat{v}_{k-1}^m of the defect equation on Ω_{k-1} :

$$L_{k-1} \hat{v}_{k-1}^m = \bar{d}_{k-1}^m, \text{ using the following} \quad (\text{D.1.1})$$

-
- ▶ If $k = 1$, use a direct or fast iterative solver for (D.1.1)
 - ▶ If $k > 1$, solve (D.1.1) approximately by performing $\gamma (\geq 1)$ k -grid cycles using the zero grid function as a first approximation:

$$\hat{v}_{k-1}^m = \text{MGCYCLE}^\gamma(k-1, \gamma, 0, L_{k-1}, \hat{d}_{k-1}^m, \nu_1, \nu_2) \quad (\text{D.1.2})$$

-
- Interpolate the correction (coarse-to-fine transfer) $\hat{v}_{k-1}^m = I_{k-1}^k \hat{v}_{k-1}^m$
 - Compute the corrected approximation on Ω_k $u_k^{m, \text{after CGC}} = \bar{u}_k^m + \hat{v}_k^m$

3. Postsmoothing

- Compute u_k^{m+1} by applying $\nu_2 \geq 0$ smoothing steps to $u_k^{m, \text{after CGC}}$:

$$u_k^{m+1} = \text{SMOOTH}^{\nu_2}(u_k^{m, \text{after CGC}}, L_k, f_k)$$

D.2 V and W cycles

In (D.1.2) in the multigrid cycle, the parameter γ appears twice. As argument of the MGCYCLE it indicates which *cycle type* must be used and the appearance as a power, indicates the *number of cycles* to be performed on the current coarse grid level. The case $\gamma = 1$ is referred to as a V-cycle and the case $\gamma = 2$ as a W-cycle. See Figures D.2.1 and D.2.2.

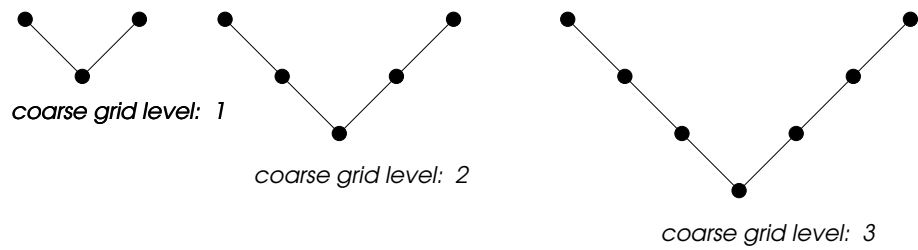


Fig. D.2.1 V-cycles for different coarse grid levels and $\gamma = 1$.

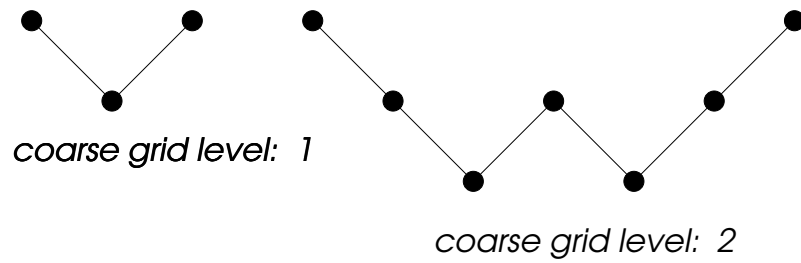


Fig. D.2.2 W-cycles for different coarse grid levels and $\gamma = 2$.

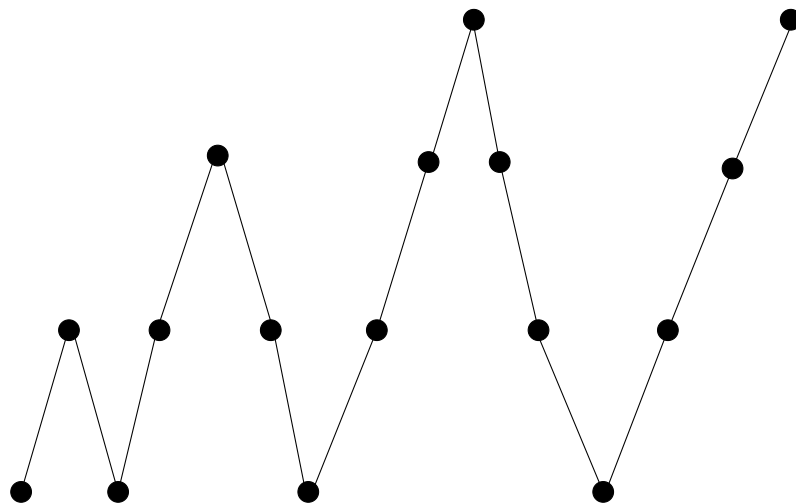


Fig. D.2.3 Full multigrid V-cycle.

Appendix E Model problems

E.1 The two dimensional Poisson equation with Dirichlet boundary conditions

In this section the two dimensional Poisson equation with Dirichlet boundary conditions is considered and defined below:

$$Lu = f, \text{ with } Lu = -\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right), (x, y) \in \Omega \quad (\text{E.1.1})$$

$$u(x, y) = 0, (x, y) \in \Gamma = \partial\Omega, \text{ where}$$

- $\Omega = (0, 1) \times (0, 1) \subset \mathbb{R}^2$,
- $\Gamma = \{(0, 0) \cup (0, 1) \cup (1, 0) \cup (1, 1)\}$,
- $N \in \mathbb{N} \rightarrow h = \frac{1}{(N+1)^2}$ with N the total number of unknowns in the x as well as the y direction,
- finite differences approximation of the partial differential operator L :

$$\tilde{L}_h = \frac{1}{h^2} \begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}.$$

In the several experiments performed in this section several multigrid parameters are varied to illustrate the effect on the total work.

In the first experiment the chosen Krylov method is Matlab's intrinsic *Bi-CGSTAB method* and the total number of unknowns is equal to N . The maximum number of Krylov iterations is taken equal to 200 and the specified tolerance is $\frac{\|r_k\|_2}{\|r_0\|_2} \leq 10^{-6}$.

In Table 24 below, N is varied and the total number of matrix vector operations is summarized. The preconditioner is chosen equal to the system matrix, say A , itself and multigrid (here ML) is used to *approximate* A^{-1} .

Note that when A^{-1} is computed exactly, one Krylov iteration is necessary to solve the system $Ax = b$ and when no preconditioner is chosen, the total number of iterations for $N = 200^2$ is equal to 350, for the same specified tolerance.

For the multigrid solution phase one W-cycle is performed and one pre and one post smoothing step are performed using Jacobi as the smoothing method. In all experiments the specified tolerance was reached.

In Table 24 it can be seen that the total number of matrix vector operations remains constant from $N = 20$ on. Therefore it can be concluded that multigrid is indeed h -independent for this model problem. Also note the difference in the number of matrix vector operations *without* a preconditioner, 350 for $N = 200^2$ against 13 when a preconditioner is used.

N	10^2	20^2	50^2	100^2	200^2	400^2
# MAT-VEC-OPs	1	12	11	13	13	13

Table 24 Two dimensional Poisson equation: varying N .

In Table 25, the same setup from Table 24 is used with $N = 400^2$. It is illustrated that the total number of matrix vector operations decreases when the number of multigrid W-cycles increases, as expected. It also seems that there is some sort of balance for the CPU time between the increasing number of multigrid cycles versus the decreasing number of iterations.

#W-cycles	1	2	3	4	5	10
# MAT-VEC-OPs	13	9	8	7	6	4
CPU time	4.90	4.72	4.80	4.79	4.71	5.02

Table 25 Two dimensional Poisson equation: varying the number of multigrid W-cycles.

In Table 26 below the multigrid cycle type is varied. Once again $N = 400^2$, one cycle is performed and one pre and post smoothing step are performed using Jacobi as smoother. MGW stands for a multigrid V-cycle, MGW for multigrid W-cycle. It seems that using a full-MGV cycle¹, leads to the smallest CPU time.

cycle type	MGV	MGW	full-MGV
# MAT-VEC-OPs	15	13	7
CPU time	5.36	5.23	2.94

Table 26 Two dimensional Poisson equation: varying the cycle type.

In Table 27 the number of pre and post smoothing steps is varied. $N = 400^2$ and full-MGV is chosen. From this table it can be concluded that for this two dimensional Poisson equation, the total number of iterations does not strongly depends on the number of pre and post smoothing steps. This may be explained because of the fact that the Poisson solutions are relatively ‘nice’ and smooth. Therefore the error might become relatively smooth after one smoothing step (when there are no smoothing steps performed, multigrid does not converge, as expected).

It is, however, remarkable that the CPU time does not increase with the number of smoothing steps. This is what one would expect because the smoothing procedure is one of the main processes in a multigrid cycle.

¹See Appendix D for an illustration of the different multigrid cycles

# steps	1	2	3	4	5
# MAT-VEC-OPs	7	9	6	8	6
CPU time	3.44	4.65	3.47	4.70	3.95

Table 27 Two dimensional Poisson equation: varying the number of pre and post smoothing steps.

This concludes the experiments for the two dimensional Poisson equation. In the next section, the two dimensional Helmholtz equation will be discussed.

E.2 The two dimensional Helmholtz equation with local absorbing boundary conditions

In this section the two dimensional Helmholtz equation with local absorbing boundary conditions is considered and defined below:

$$Lu = -S, \text{ with } Lu = -\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + k_0^2 u, (x, y) \in \Omega \quad (\text{E.2.1})$$

$$\frac{\partial u}{\partial n} - \iota k_0 u = 0, (x, y) \in \Gamma = \partial\Omega, \text{ where}$$

- $\Omega = (0, 1) \times (0, 1) \subset \mathbb{R}^2$
- $\Gamma = \{(0, 0) \cup (0, 1) \cup (1, 0) \cup (1, 1)\}$
- $N \in \mathbb{N} \rightarrow h = \frac{1}{(N+1)^2}$ with N the total number of unknowns in the x as well as the y direction
- c is the P-wave velocity as an implicit function of space
- $\omega^* = 2\pi f^*$ with f^* the frequency²
- k_0 is the dimensionless wavenumber defined as: $\frac{\omega^2}{c^2}$
- S is a source term
- $\iota^2 = -1$
- finite differences approximation of the partial differential operator L :

$$\tilde{L}_h = \frac{1}{h^2} \begin{bmatrix} & -1 & \\ -1 & 4 - k_0^2 h^2 & -1 \\ & -1 & \end{bmatrix}$$

This model problem is used to illustrate the performance of the IDR(4) method using two different preconditioners M_1 and its reduced form M_2 . See Chapter 5 for the definitions of these preconditioners.

²Note that ‘*’ denotes a dimensionfull variable

k_0	10	20	30
# MAT-VEC-OPs	15	33	65

Table 28 Two dimensional Helmholtz equation: varying wavenumber, constant total number of unknowns $N = 101^2$ and using M_1 with optimal real shift $(\beta_1, \beta_2) = (-1, 0)$.

N	101^2	201^2	301^2
# MAT-VEC-OPs	65	65	67

Table 29 Two dimensional Helmholtz equation: constant wavenumber $k_0 = 30$ and varying N , using M_1 with optimal real shift $(\beta_1, \beta_2) = (-1, 0)$.

k_0	10	20	30
# MAT-VEC-OPs	23	49	83

Table 30 Two dimensional Helmholtz equation: varying wavenumber, constant total number of unknowns $N = 101^2$ and using M_2 with optimal real shift $(\beta_1, \beta_2) = (-1, 0)$.

N	101^2	201^2	301^2
# MAT-VEC-OPs	83	90	97

Table 31 Two dimensional Helmholtz equation: constant wavenumber $k_0 = 30$ and varying N , using M_2 with optimal real shift $(\beta_1, \beta_2) = (-1, 0)$.

When the preconditioners are used with the optimal real shift, the following conclusions can be drawn for the total number of matrix vector operations:

- A linear increase with the wavenumber k_0 .
- Nearly constant for fixed wavenumber and increasing N .
- There is no significant difference when M_1 is replaced by its reduced form M_2 .

k_0	10	20	30
# MAT-VEC-OPs	10	19	27

Table 32 Two dimensional Helmholtz equation: varying wavenumber, constant total number of unknowns $N = 101^2$ and using M_1 with optimal complex shift $(\beta_1, \beta_2) = (1, -0.5)$.

N	101^2	201^2	301^2
# MAT-VEC-OPs	27	27	24

Table 33 Two dimensional Helmholtz equation: constant wavenumber $k_0 = 30$ and varying N , using M_1 with optimal complex shift $(\beta_1, \beta_2) = (1, -0.5)$.

k_0	10	20	30
# MAT-VEC-OPs	17	27	42

Table 34 Two dimensional Helmholtz equation: varying wavenumber, constant total number of unknowns $N = 101^2$ and using M_2 with optimal complex shift $(\beta_1, \beta_2) = (1, -0.5)$.

When the preconditioners are used with the optimal complex shift, the following conclusions can be drawn for the total number of matrix vector operations:

- A linear increase with the wavenumber k_0 .
- Nearly constant for fixed wavenumber and increasing N .
- There is no significant difference when M_1 is replaced by its reduced form M_2 .

N	101^2	201^2	301^2
# MAT-VEC-OPs	42	45	45

Table 35 Two dimensional Helmholtz equation: constant wavenumber $k_0 = 30$ and varying N , using M_2 with optimal complex shift $(\beta_1, \beta_2) = (1, -0.5)$.

Appendix F Comparison between two dimensional Maxwell solver and COMSOL

In this appendix the solution of the two dimensional Maxwell solver is compared to the surface plots obtained using COMSOL¹ rendering thanks to Dr. D.J.P. Lahaye. The figures obtained by the two dimensional Maxwell solver are mirrored compared to the output of COMSOL. Furthermore, the node based implementation is used to reproduce these figures. For the edge based implementation, similar results were obtained.

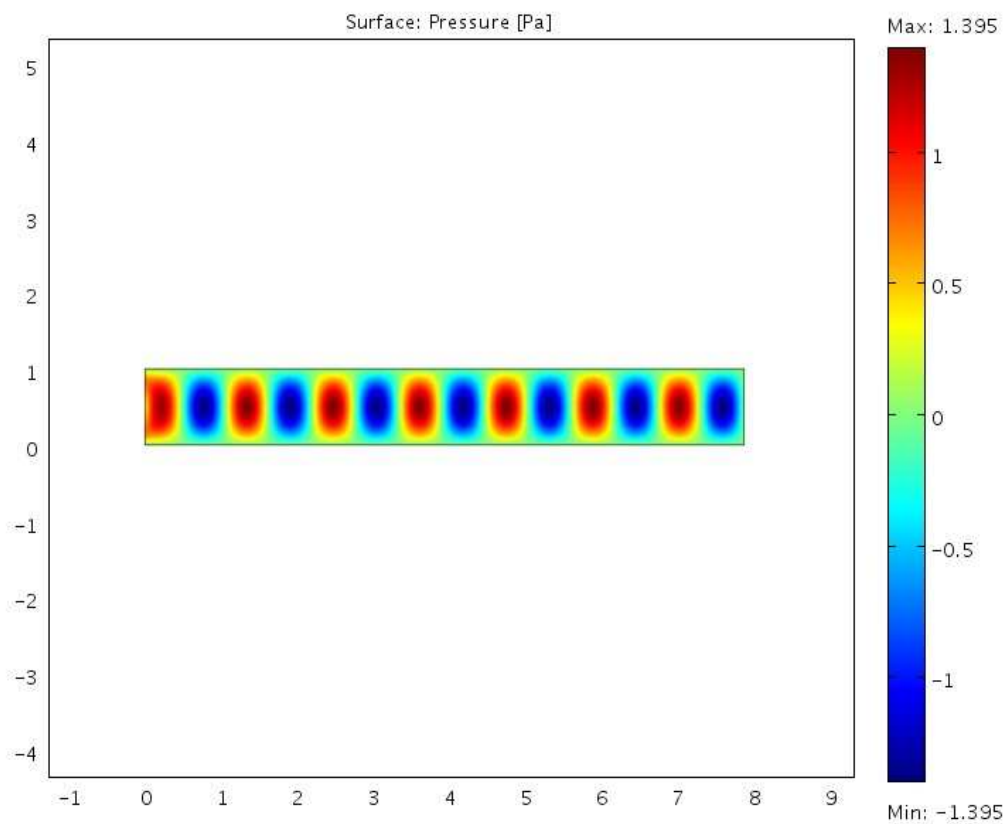


Fig. F.0.1 COMSOL surface plot for two dimensional cavity with height 1 and depth 8.

¹See <http://www.comsol.nl/products/reaction/> for more information about COMSOL.

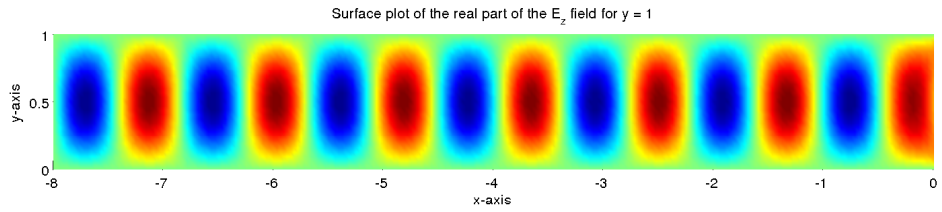


Fig. F.0.2 Two dimensional Maxwell solver: surface plot for two dimensional cavity with height 1 and depth 8.

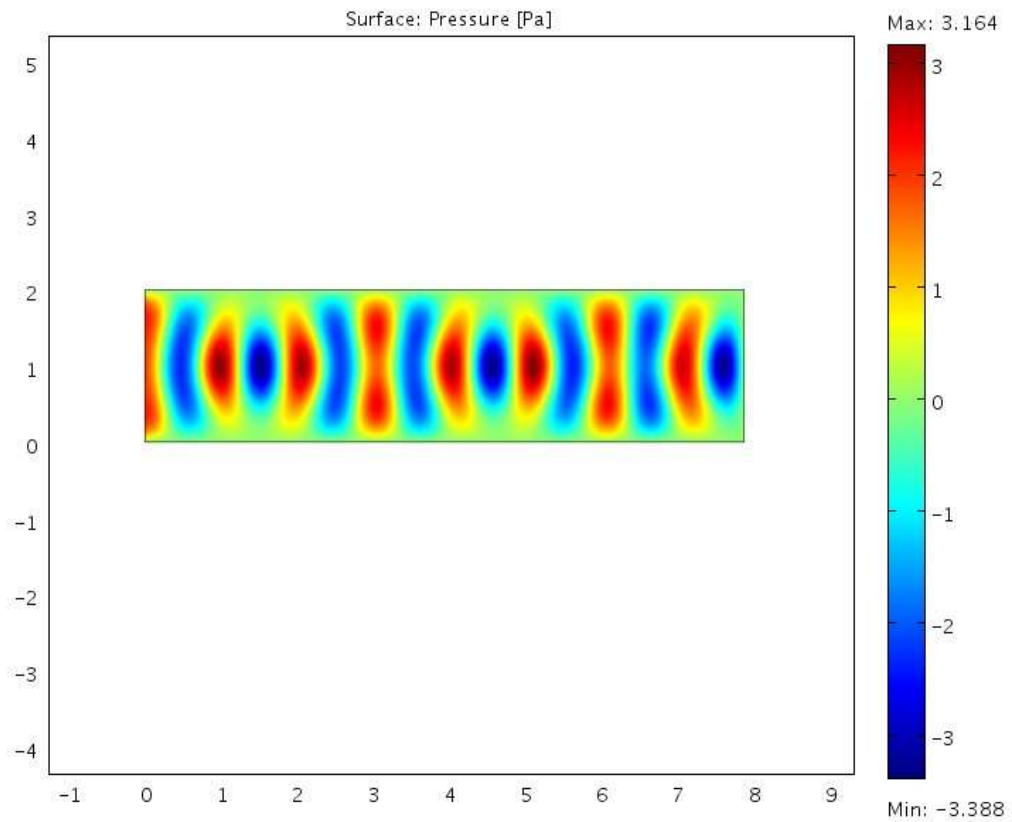


Fig. F.0.3 COMSOL surface plot for two dimensional cavity with height 2 and depth 8.

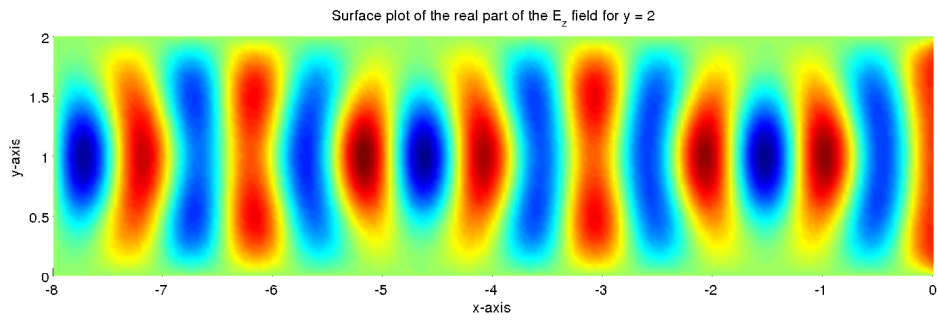


Fig. F.0.4 Two dimensional Maxwell solver: surface plot for two dimensional cavity with height 2 and depth 8.

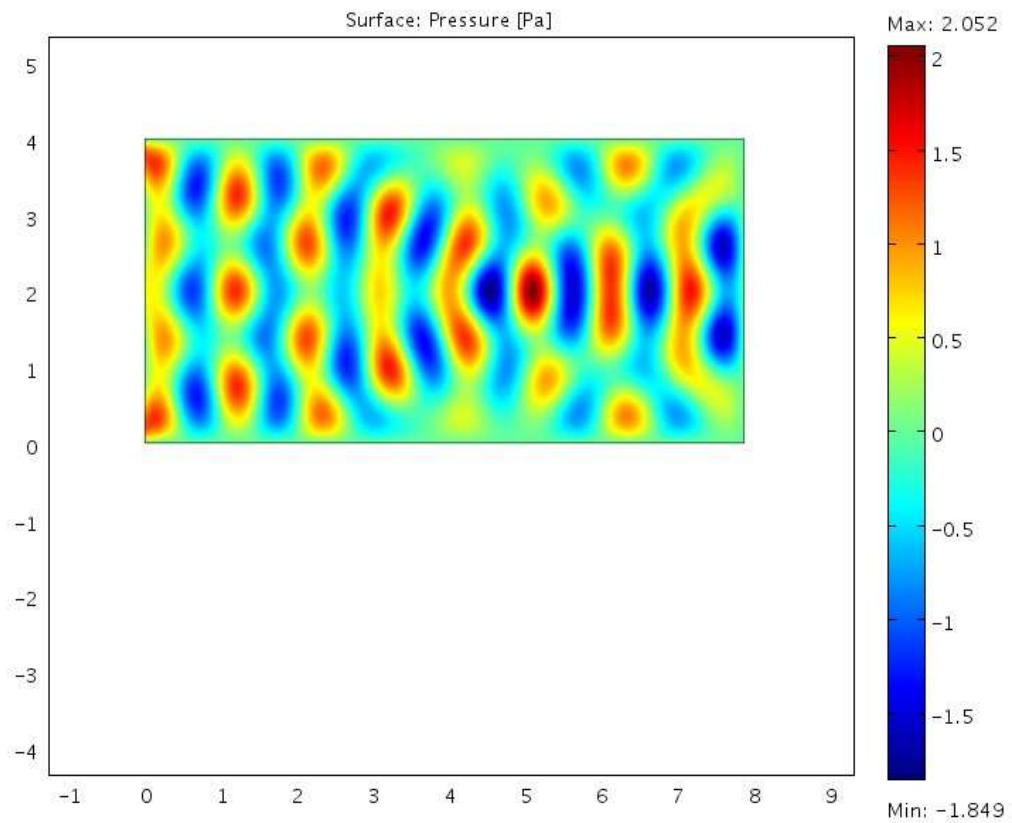


Fig. F.0.5 COMSOL surface plot for two dimensional cavity with height 4 and depth 8.

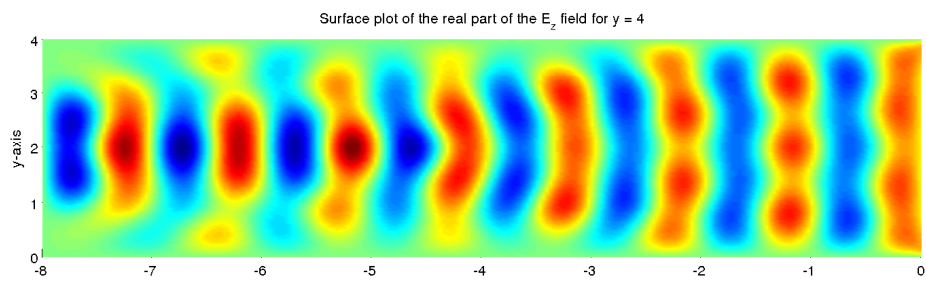


Fig. F.0.6 Two dimensional Maxwell solver: surface plot for two dimensional cavity with height 4 and depth 8.

Appendix G Sparsity patterns of the different preconditioners

G.1 The two dimensional vector wave equation

In this section the different sparsity patterns are included obtained using the two dimensional vector wave discretization. In each figure it is specified which preconditioner is chosen, which FEM implementation type is considered and which type of absorbing boundary conditions are imposed on the boundary. In all these cases,

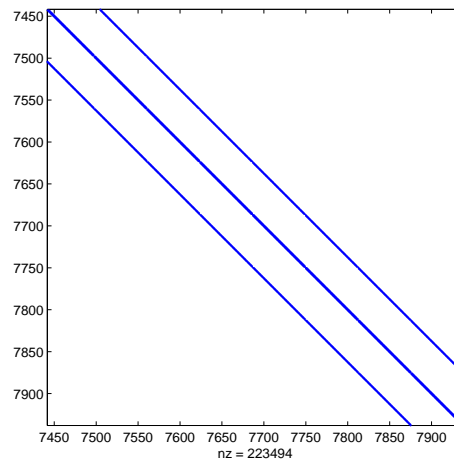


Fig. G.1.1 Two dimensional vector wave equation, node based FEM implementation: sparsity pattern for preconditioner M_1 and local absorbing boundary conditions.

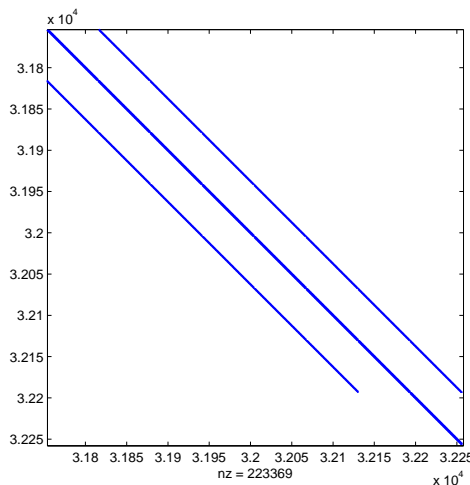


Fig. G.1.2 Two dimensional vector wave equation, node based FEM implementation: sparsity pattern for preconditioner M_2 and local absorbing boundary conditions.

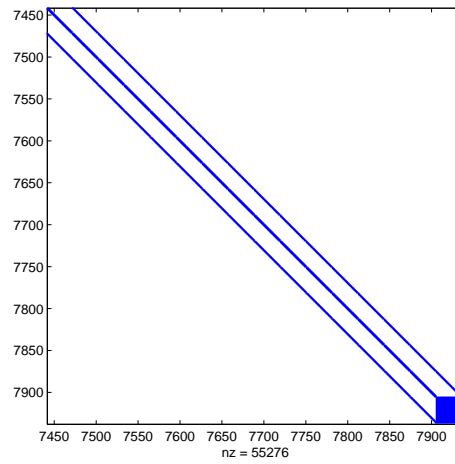


Fig. G.1.3 Two dimensional vector wave equation, node based FEM implementation: sparsity pattern for preconditioner M_1 and global absorbing boundary conditions.

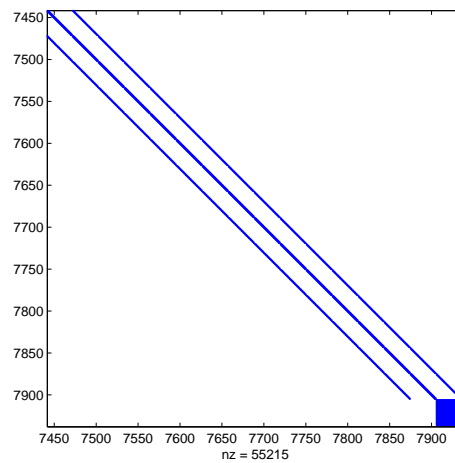


Fig. G.1.4 Two dimensional vector wave equation, node based FEM implementation: sparsity pattern for preconditioner M_2 and global absorbing boundary conditions.

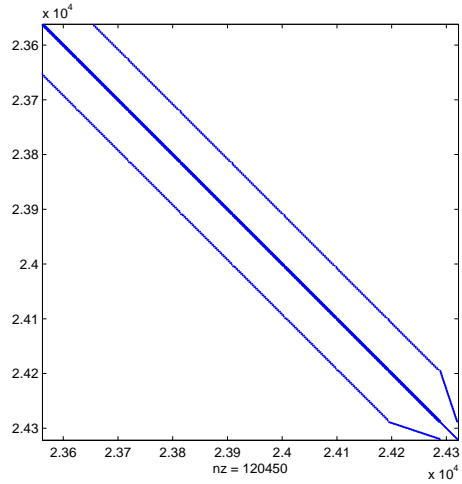


Fig. G.1.5 Two dimensional vector wave equation, edge based FEM implementation: sparsity pattern for preconditioner M_1 and local absorbing boundary conditions.

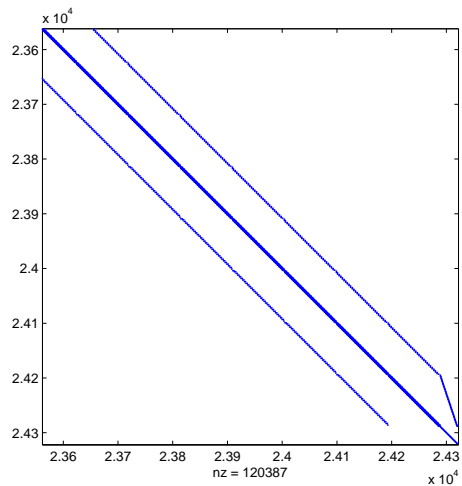


Fig. G.1.6 Two dimensional vector wave equation, edge based FEM implementation: sparsity pattern for preconditioner M_2 and local absorbing boundary conditions.

G.2 The three dimensional vector wave equation

In this section the different sparsity patterns are included obtained using the three dimensional vector wave discretization. In each figure it is specified which preconditioner is chosen. In this case global boundary conditions are considered.

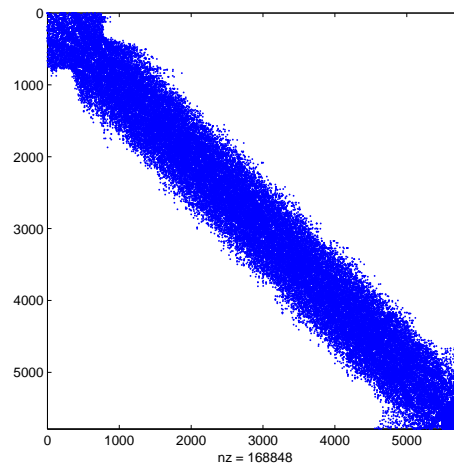


Fig. G.2.1 Three dimensional sparsity pattern for preconditioner M_1 and global absorbing boundary conditions. Zeroth order basis functions are used here and the mesh size $h = 0.15$ for the cavity with dimensions $1.5\lambda \times 1.5\lambda \times 0.6\lambda$.

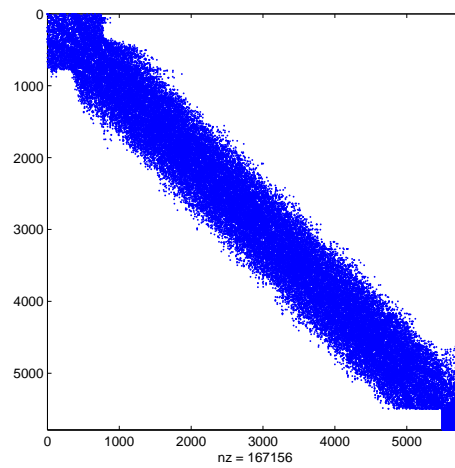


Fig. G.2.2 Three dimensional sparsity pattern for preconditioner M_2 and global absorbing boundary conditions. Zeroth order basis functions are used here and the mesh size $h = 0.15$ for the cavity with dimensions $1.5\lambda \times 1.5\lambda \times 0.6\lambda$.

Appendix H Using M_{loc} instead of M_{gl}

For the two dimensional vector wave equation it was concluded that when the original system with *global* absorbing boundary conditions is preconditioned by a preconditioner based on *local* absorbing boundary conditions, the total number of matrix vector operations was not significantly affected. The solution of the preconditioner based on local absorbing boundary conditions can be performed much more efficiently than is the case when global absorbing boundary conditions are used: there is no blockstructure and the preconditioner is sparsely populated. Therefore, it is advisable to implement local absorbing boundary conditions in the preconditioner system for the vector wave equation. This is relatively easy to accomplish.

Recall the fully populated matrix $[P_{st}]$ from Chapter 2:

$$[P^{st}] = 2 \iint_{S^s} \{\nabla \cdot \mathbf{S}^s\} \left\{ \iint_{S^t} \{\nabla' \cdot \mathbf{S}^t\}^T G_0 dS' \right\} dS - 2k_0^2 \iint_{S^s} \{\mathbf{S}^s\} \cdot \left\{ \iint_{S^t} \{\mathbf{S}^t\}^T G_0 dS' \right\} dS. \quad (\text{H.0.1})$$

This matrix is obtained from the boundary integral. In this case the *global* absorbing boundary conditions are imposed. When *local* absorbing boundary conditions are imposed, the boundary element matrix K_i becomes:

$$K_i = 2\iota k_0 \int_{\partial S} (\hat{\mathbf{n}} \times \mathbf{W}_i) \cdot (\hat{\mathbf{n}} \times \mathbf{W}_i) d\partial S = 4\iota k_0 l_i. \quad (\text{H.0.2})$$

The latter equation leads to a sparsely populated boundary integral matrix.