Nationaal Lucht- en Ruimtevaartlaboratorium
National Aerospace Laboratory NLR

NLR-TR-2008-282

Multigrid acceleration of a preconditioned Krylov method for the solution of the discretized vector wave equation

Master Thesis for the degree of Master of Science in Applied Mathematics: Literature report

S.M.F. Abdoel

.

Nationaal Lucht- en Ruimtevaartlaboratorium
National Aerospace Laboratory NLR

Executive Summary

Multigrid acceleration of a preconditioned Krylov method for the solution of the discretized vector wave equation

Master Thesis for the Degree of Master of Science in Applied Mathematics: Literature Report

Problem area
Radar cross section prediction methods are used to analyze the radar signature of military platforms when the radar signature cannot be determined experimentally because:

- The platform is in the design, development or procurement phase
- 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-side 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 simple 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 incorpo-

Report no.
NLR-TR-2008-282

Author(s)
S.M.F. Abdoel

Classification report
Unclassified

Date
June 2008

Knowledge area(s)
Numerical Mathematics

Descriptor(s)
Radar
RCS
Large sparse systems
Iterative methods
Algebraic multigrid
Preconditioning

rate so called *multigrid acceleration* in the existing algorithm.

Results and conclusions
Multigrid methods are successfully applied to efficiently solve large linear systems arising from discretization of partial differential equations. Due to the properties of the governing equations, multigrid must be applied *indirectly* to accelerate the solution of the current application. Furthermore, propositions are made for the incorporation of a multigrid black box solver into the existing algorithm.

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.

Nationaal Lucht- en Ruimtevaartlaboratorium, National Aerospace Laboratory NLR

Anthony Fokkerweg 2, 1059 CM Amsterdam,
P.O. Box 90502, 1006 BM Amsterdam, The Netherlands
Telephone +31 20 511 31 13, Fax +31 20 511 32 10, Web site: www.nlr.nl

Nationaal Lucht- en Ruimtevaartlaboratorium
National Aerospace Laboratory NLR

NLR-TR-2008-282

Multigrid acceleration of a preconditioned Krylov method for the solution of the discretized vector wave equation

Master Thesis for the degree of Master of Science in Applied Mathematics: Literature report

S.M.F. Abdoel

.

Customer	NLR, TU Delft
Contract number	----
Owner	NLR
Division NLR	Aerospace Vehicles
Distribution	Limited
Classification of title	Unclassified
	June 2008

Approved by:

Author	Reviewer	Managing department
27-6-2008	27-6-2008	27/6/2008

Summary

RADAR (**Ra**dio **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 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 (MG) methods can be used in many applications where large linear systems arise from the discretization of partial differential equations. However, when the system matrix is indefinite, multigrid cannot be directly applied. To get a definite system matrix, the system must be preconditioned (e.g. in the current application the shifted Laplace preconditioner is used). Hence multigrid is said to have an *indirect* application in the current system.

Furthermore, multigrid methods are independent of the mesh size h and they can solve a linear system with N unknowns, in cN arithmetic operations (c is a constant). To explain the multigrid idea in this thesis, the classical *geometric multigrid* methods will be introduced together with all the multigrid components. This will be achieved using a simple one dimensional Laplace problem. After this introduction it will be discussed that geometric multigrid has several favorable aspects but unfortunately also some crucial disadvantages for application in the current problem. To overcome these disadvantages *Algebraic Multigrid* (AMG) will be used instead.
The power of AMG lies in the possibility of its 'operator-independent' formulation and its applicability in general domains (e.g.: not only structured grids). Algebraic multigrid can also be extended for complex valued systems as will be seen in this thesis. Furthermore, it will be shown that AMG can be effectively used as a solution method for the current application, combined with the shifted Laplace preconditioner. Since the AMG preconditioner is constant in every iter-

ation, GCR can be replaced by a short recurrence method e.g. Bi-CGSTAB, which will lead to a considerable reduction of memory requirements.

As AMG (and MG) are widely used to solve large linear systems, there are several black box solvers available for many (specific) problems. The challenge is to choose a black box solver which can handle the properties in the current system (e.g. high frequencies). Based on the information about available multigrid solvers, one of the most suitable black box solvers for the current application is a Multilevel (ML) Preconditioning Package developed by Sandia National Laboratories (see ref. 12).

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's 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, dr. ir. Duncan van der Heul for all the guidance and patience he had during the first period of my thesis.

For the next period of my thesis, it is likely that I will cooperate with dr. ir. Johan Kok and dr. Harmen van der Ven when the implementation phase will start in the Fortran 90 programming language. In addition I would like to thank Johan Kok for the useful tips for the layout of my report. I am looking forward to work with them.

I also would like to thank my (provisional) roommate ir. Bart Eussen for some useful comments on the text of my report. Finally I would like to thank my supervisor at the Delft University of Technology, prof. dr. ir. Kees Vuik for his guidance and useful comments on my work.

Contents

3 Tables

27 Figures

(104 pages in total)

List of figures

List of tables

Preface

This is the literature report which is part one of 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.

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. Duncan van der Heul and in Delft by prof. dr. C. Vuik.

Amsterdam, June 27, 2008 Shiraz Abdoel

This page is intentionally left blank.

1 Introduction

RADAR (**Ra**dio **D**etection and **R**anging) is technology to detect aircraft and ships by using electromagnetic waves. It is very important to be able to predict the detectability of a platform by radar in the development stage. A direct consequence of working in this development stage is that it is not possible to determine the radar signature experimentally. Therefore, 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 typical 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 the Maxwell's equations with the appropriate boundary conditions inside this cavity. This equation is discretized by using the *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 preconditioned by the shifted Laplace preconditioner and the preconditioned system is solved using a Krylov subspace method, namely GCR. The greatest disadvantages from the GCR method are the storage requirements and the dependence of the total work of GCR on the number of degrees of freedom. The current implementation gives satisfactory results for intermediate system sizes up to $\#$(degrees of
freedom) $= 3.0 \cdot 10^5$. In the current application, however, the $\#$(degrees of freedom) $= 5.19 \cdot 10^5$ for a wavenumber equal to 10 and $\#$(degrees of freedom) $= 2.03 \cdot 10^7$ for a wavenumber equal to 100. Therefore more investigation is needed to advance towards these large numbers of degrees of freedom.

The purpose of this thesis is to investigate another solver for the preconditioner system, namely a multigrid solution method. Multigrid can be classified as geometric or algebraic, depending on the availability of the underlying grid. As in the current application there is no specified grid available beforehand, geometric multigrid cannot be used. When the grid locations are known, but are allowed to be unstructured or irregular, algebraic multigrid will be used as solver. In this case the coefficients in the system matrix will be used to specify the connections within the grid. In fact, when the system matrix is known, the grid can be derived, and algebraic multigrid can be used. In the current application, the system matrix is known and therefore algebraic multigrid can be used.

Since it is not likely that a full algebraic multigrid implementation can be realized in the time available, and because there are several black box implementations available, the most suitable algebraic multigrid black box solver will be chosen to be incorporated in the existing algorithm. The specific requirements that have to be met by this black box solver will be discussed.

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. In Chapter 3 the (iterative) Krylov subspace methods are discussed combined with several preconditioning techniques. Hereafter in Chapter 4 multigrid methods will be the subject. The basic principles of (classical) geometric multigrid methods will be discussed after which algebraic multigrid methods for complex valued linear systems will be treated. At the end an outline for the most promising direction of further research is given, included with the choice of the algebraic multigrid black box solver that will be used in the current application.

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. After the Maxwell equations are discretized, they form a linear system of equations which must be solved numerically.

In Section 2.2 the Maxwell equations are introduced and the necessary tools and assumptions made to arrive at the dimensionless form of these equations are presented in Subsection 2.2.2. There are applications of the FEM using zeroth order basis functions, which unfortunately lead to a large number of unknowns for a large scatterer and have 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. Finally 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 jet engine air intake of a modern fighter aircraft accounts for the main part of the scattered field for an *electromagnetic* wave that excites the platform. This air intake is a deep open cavity and is characterized by a large **L**ength/**d**iameter 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. 5).

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 form[1]:

$$\nabla^* \times \mathcal{E}^* = -\frac{\partial^* \mathcal{B}^*}{\partial^* t^*} \tag{2.2.1}$$

$$\nabla^* \times \mathcal{H}^* = \frac{\partial^* \mathcal{D}^*}{\partial^* t^*} + \mathcal{J}^* \tag{2.2.2}$$

$$\nabla^* \cdot \mathcal{D}^* = \mathcal{Q}^* \tag{2.2.3}$$

$$\nabla^* \cdot \mathcal{B}^* = 0 \tag{2.2.4}$$

$$\nabla^* \cdot \mathcal{J}^* = -\frac{\partial^* \mathcal{Q}^*}{\partial^* t^*} \tag{2.2.5}$$

[1] In this thesis dimensionfull variables are denoted with a *

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

$\mathcal{E}^*$ = electric field intensity $[\frac{V}{m}]$
$\mathcal{D}^*$ = electric flux density $[\frac{C}{m^2}]$
$\mathcal{H}^*$ = magnetic field intensity $[\frac{A}{m}]$
$\mathcal{B}^*$ = magnetic flux density $[\frac{Wb}{m^2}]$
$\mathcal{J}^*$ = electric current density $[\frac{A}{m^2}]$
$\mathcal{Q}^*$ = electric charge density $[\frac{C}{m^3}]$
t^* = time $[s]$

From this point on the assumption is made that the field quantities above are harmonic oscillating functions with a angular frequency ω^*, called *time-harmonic functions.*
ω^* is defined as $\omega^* = 2\pi f^*$, with f^* the frequency measured in hertz. In the current application discussed in this thesis, $f^* = 10$ GHz.
Let $\mathcal{F}^*(\mathbf{x}^*, t^*)$ denote a time-harmonic function denoted by:

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

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

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

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:

$$\bigtriangledown^* \times \mathbf{E}^* = j\omega^* \mathbf{B}^* \tag{2.2.8}$$

$$\bigtriangledown^* \times \mathbf{H}^* = j\omega^* \mathbf{D}^* + \mathbf{J}^* \tag{2.2.9}$$

$$\bigtriangledown^* \cdot \mathbf{J}^* = -j\omega^* q^* \tag{2.2.10}$$

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.

Note that the introduction of the general equations concerns five equations (2.2.1) - (2.2.5), but there are six variables. Since equations (2.2.8) - (2.2.10) cover three equations, additional relations are needed to close the problem. These so called *constitutive relations* are used which describe the macroscopic properties of the medium of interest. They are given by:

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

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

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

Where the parameter:

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

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 constants and it is also possible that they are complex valued (e.g. application of radar absorbing materials). 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 used, the vector wave equation in the presence of a source
$\mathbf{J}^* \neq 0$ becomes:

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

When the following two definitions are used:

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

equation (2.2.14) can be rewritten as:

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

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

The last thing to discuss, is transforming the system of equations into a so called *well-posed* problem. 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:

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

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

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. $\mathbf{N}^*(\mathbf{r}) = \mathbf{K}^*(\mathbf{r}) * G(r, r') = \int\int_{S_a} \mathbf{K}^*(\mathbf{r}')G(\mathbf{r}, \mathbf{r}')d\mathbf{r}' = \int\int_{S_a}[\hat{n} \times \mathbf{E}^*(\mathbf{r}')d\mathbf{r}']$
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

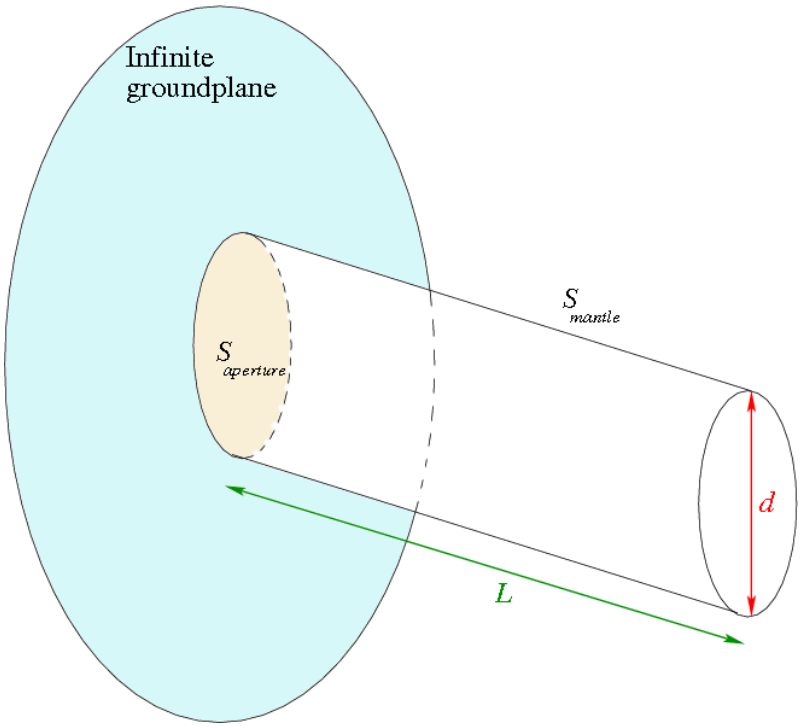

Fig. 2.2.1 Schematic view of cylindrical cavity with length L and cross section diameter d

2.2.2 Dimensional analysis

Dimensional analysis is a conceptual tool that can be used 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
- electric current: I

Quantities

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

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

Position Variables

Define $x, y, z = \dfrac{x^*}{R}, \dfrac{y^*}{R}, \dfrac{z^*}{R}$ respectively

Parameters

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

Operator: gradient $\bigtriangledown^* \rightarrow \dfrac{1}{R} \bigtriangledown$

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

$$\bigtriangledown \times \bigtriangledown \times \mathbf{E} - k_0^2 \mathbf{E} = -j k_0 Z_0 \mathbf{J} \tag{2.2.18}$$

From this dimensionless form it is clear that the most important parameter in the left hand side of this equation is the parameter 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. Therefore, k_0 is defined as $k_0 := dk_0^*$.

2.2.3 Dependence of RCS on the dimensionless wavenumber

According to Knott et al. (ref. 6), the RCS of a scattering body has a strong relationship with the non-dimensional wavenumber k_0. 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.

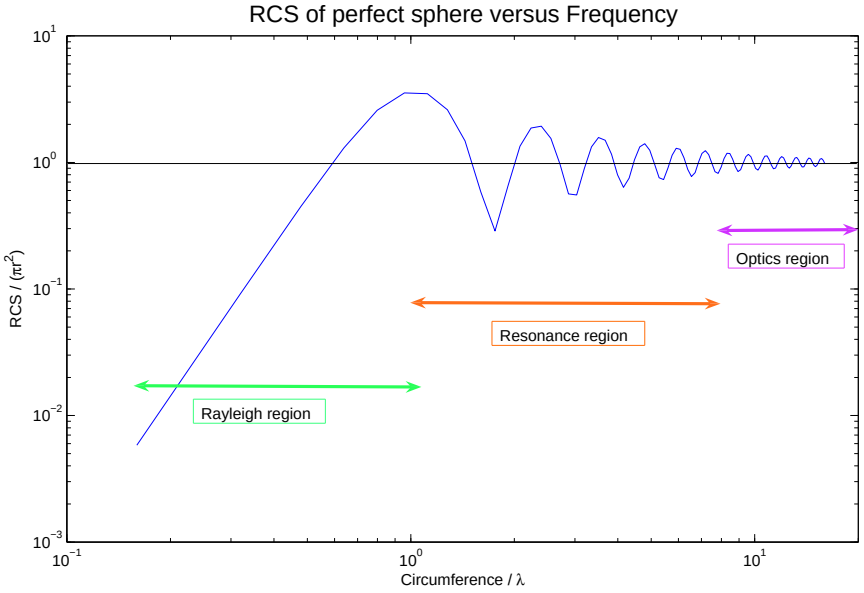

Fig. 2.2.2 The RCS σ of a metallic sphere with radius a illustrates the three scattering regions

From a computational point of view, the following discussion about a comparison in the dimensionless wavenumber is included. Suppose that there are two dimensionless wavenumbers k_1 and k_2 with $k_1 > k_2$ and recall the relations $k_i = \frac{2\pi d_i}{\lambda_i}$ and the ratios $\frac{L_i}{d_i}, i = 1, 2$ (all other parameters are fixed).

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 indefiniteness[2] of the system matrix

Ad. 1 In Hooghiemstra (ref. 8, Section 6.2) it is explained that there is a close relationship between k_i, for $i = 1, 2$ and the number of unknowns per wavelength, hence there is also a relationship between k_i, for $i = 1, 2$ and the total number of unknowns N. Therefore the assumption $k_1 > k_2$ results in $N_1 > N_2$.

Ad. 2 The discretization matrix of the systems with $k_1 > k_2$, will also show a very important difference to mention. The k_i, for $i = 1, 2$ appears on the main diagonal of the discretization matrix and therefore can be the originator of the matrix to become indefinite. 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 (linear) system of equations.
- some iterative methods cannot be used at all, as they rely of the definiteness of the system in order to converge.

In the next section the finite element discretization method will be discussed.

2.3 Finite element discretization method

From the ongoing research on computational electromagnetics it is known that the following numerical methods can be used to numerically solve systems derived from electromagnetic problems: the method of moments (MOM), the finite difference method (FDM) and the finite element discretization method (FEM). 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 these 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 they 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 overcome these discontinuities, the so called 'weak formulation' is a natural method to use.

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 geometrical modeling* and the

[2] see for definition of definiteness Section 2.4

higher order representation of the unknown quantities. For the FEM these unknown quantities can be the electric or magnetic field as seen in the Maxwell equations.
Also important to mention is the usage of vector basis functions. The most useful functions in this class are the *curl or divergence-conforming* bases functions because of their ability to have continuous tangential or normal components. In the discretization of the Maxwell equations in this thesis, the continuity in the tangential components will be used. The last remark concerning the justification of these vector basis functions is that they do not allow so called spurious solutions, as opposed to node based finite element discretization.

In the context of the FEM used here, the higher order vector basis functions proposed by Graglia, Wilton and Peterson (ref. 18) are used. The following setup is stated:

- consider a curvilinear tetrahedral element in the xyz-space[3]
- 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:

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

  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 the table below 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 $\bigtriangledown\xi_3 \cdot (\bigtriangledown\xi_1 \times \bigtriangledown\xi_2)$ is strictly positive
- the vector basis function associated with the edge shared by faces γ and β is given by:

$$\mathbf{N}_{\gamma\beta}(\mathbf{r}) = \xi_n \bigtriangledown \xi_m - \xi_m \bigtriangledown \xi_n \tag{2.3.1}$$

[3]Tetrahedral elements are easily extended from triangular elements in two dimensions

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

- 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 to be discontinuous, as occurs at the interface between two media with different permeability

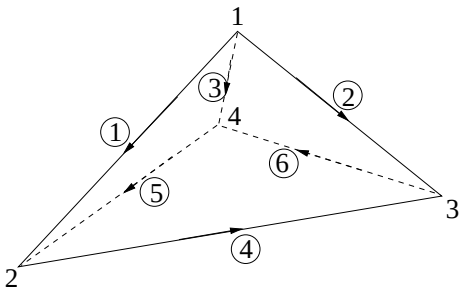

Fig. 2.3.1 The ordering of Table 1 is used here to number the edges of the tetrahedron

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. 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:

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

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

$$\hat{R}_i(p,\xi) = R_{i-1}\big(p, \xi - \frac{1}{p}\big) \tag{2.3.3}$$

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

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

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

- the value $\ell_{\gamma\beta}^{(ijkl)} = |\boldsymbol{\ell}_{\gamma\beta}|$ 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 as[4]:

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

  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 bases functions are given by:

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

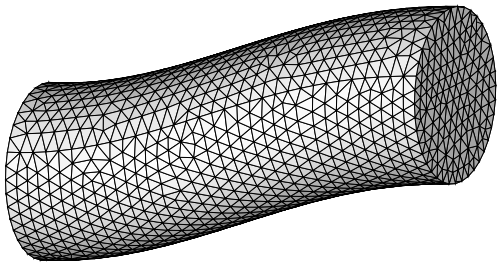

Fig. 2.3.2 Tetrahedral elements for domain Ω

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

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

(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 freedom. In the current application $p = 2$: there are 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. 18), curvilinear elements are considered. In the current application however, rectilinear 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. Therefore there are two relations needed:
- one for describing the vector basis functions for the approximation within an element (e): equation (2.3.6)
- one for describing the surface (s) field expansion: equation (2.3.7)

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

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

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

Substituting[5] these relations in the functional and applying Ritz's method, results in the following *functional*:

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

[5]Note that $\{.\}$ is used to denote a vector with elements that are a vector itself

Here the following is used:

$\star\ M =$ total number of volume elements in the cavity

$\star\ M_s =$ total number of surface elements on the mantle

$$\star \text{ matrix } [K^e] = \iiint_{V^e} \left[\frac{1}{\mu_r} \{\bigtriangledown \times \mathbf{N}^e\} \cdot \{\bigtriangledown \times \mathbf{N}^e\} - k_0^2 \varepsilon_r \{\mathbf{N}^e\} \cdot \{\mathbf{N}^e\} \right] dV \tag{2.3.9}$$

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

$\star$ matrix $[P^{st}]$ is obtained from the boundary integral and is defined as

$$[P^{st}] = 2 \iint_{S^s} \{\bigtriangledown \cdot \mathbf{S}^s\} \left\{ \iint_{S^t} \{\bigtriangledown' \cdot \mathbf{S}^t\}^T G_0 dS' \right\} dS - 2k_0^2 \iint_{S^s} \{\mathbf{S}^s\} \cdot \left\{ \iint_{S^t} \{\mathbf{S}^t\}^T G_0 dS' \right\} dS \tag{2.3.11}$$

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.

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 by[6]:

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

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 oppose 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 the 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 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.

[6] For a complete analysis of this subject, the reader is referred to [8, Chapter 6]

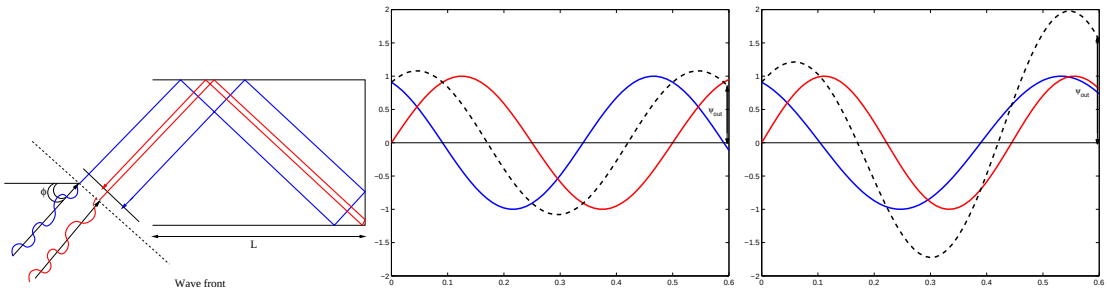

Fig. 2.3.3 ***Left:*** *The wave front enters the cavity with incidence angle ϕ. Two waves with initial phase difference $\psi_{in} = \frac{\lambda}{4}$ are followed. After reflection through the cavity there is an accumulated phase error ε –* ***Middle:*** *The exact phase difference ψ_{out} –* ***Right:*** *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. In this case the dispersion error accumulates and leads to inaccurate results. One way to achieve this is 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. 10), the maximum phase error *per wavelength* is an important quantity in this analysis. It is defined as:

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

Here the following is used:

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

According to Hooghiemstra (ref. 8) 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.

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 the following 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

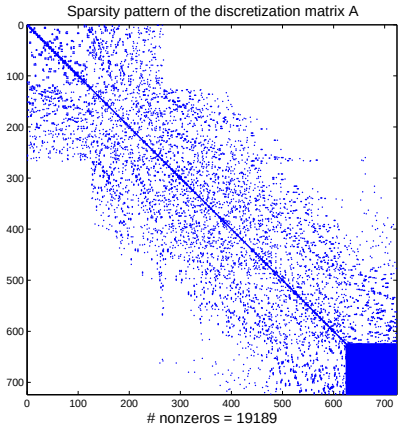

Fig. 2.4.1 $A \in \mathbb{C}^{n\times n}, n = 723, h = 0.25$. *Dimensions rectangular cavity:* $1.5\lambda \times 1.5\lambda \times 0.6\lambda$. *Fully populated block consists of 9801 nonzeros. The total number of nonzeros is 19.189. The complex valued part of the matrix consists of the unknowns on 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 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 complex, it is not a Hermitian (or self-adjoint) matrix. 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 $||.||$ is defined as:

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

  Here $A \in \mathbb{C}^{n\times n}$ and A nonsingular. Of course, this definition depends on the choice of norm.

- An $(n \times n)$ 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 definition is that for symmetric or Hermitian matrices M,the eigenvalues of M are greater than (or equal to) zero or the eigenvalues of M are less than (or equal to) zero.
  The matrix A in the current application has positive and negative eigenvalues and hence the matrix 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 linear solution methods that can be used.

In the next chapter iterative solution methods for the linear system will be outlined.

3 Iterative solutions of the linear system

3.1 Introduction

In this chapter the solution of linear system

$$Au = f \tag{3.1.1}$$

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. First direct solution methods will be compared to iterative methods, followed by some important properties of *basic* iterative methods. Hereafter, the so called *Krylov subspace methods* will be discussed in Section 3.2. The subject of Section 3.3 is *preconditioning* and in Subsection 3.3.2 a special class of preconditioners for the Maxwell equations will we introduced, namely the *shifted Laplace preconditioner*. Finally in Subsection 3.3.3 the *block preconditioner* will be considered, that is necessary to handle the partially fully and partially sparsely populated matrix in the current application (recall Figure 2.4.1).

3.2 Direct methods versus iterative methods

One way of solving the linear system (3.1.1) is by using direct solution methods. Direct methods solve the problem by a finite sequence of operations and deliver an exact solution (e.g. Gaussian elimination). Direct methods work well when the system matrix A is dense but unfortunately, when the total number of unknowns N becomes very large, the time to solve the system becomes unacceptably high. In addition, direct methods are sensitive to rounding errors.
Alternatively iterative methods try to solve the linear system by finding successive approximations to the solution starting from an initial guess. In Table 2[1] these two types of methods are compared to each other where after a classification of iterative methods will be given.

In general the following classification of iterative methods can be made:

- basic iterative methods
- Krylov subspace methods
  - short recurrence methods
  - long recurrence methods
  - induced dimension reduction methods

3.2.1 Basic iterative methods

It is already stated that the linear system considered here has the form $Au = f$. Consider the following splitting of matrix A:

$$A = B + (A - B)$$

[1] Error smoothing will be discussed in Chapter 4

Direct methods	**Iterative methods**
- Computational work increases with $O(N^3)$	- Computational work depends on the method chosen and for single grid basic iterative methods this will be of order $O(N^\alpha), \alpha > 1$
- The storage requirements increase with $O(N^2)$	- The storage requirements increase with $O(N)$
- Sensitive to rounding errors	- Also sensitive to rounding errors
- Perform well when A is dense	- Are very effective if A is sparse
- Performance does not depend on the spectrum of the eigenvalues of A	- Performance strongly depends on the eigenvalue spectrum of A
- Can be effectively used as *error smoothers*	

Table 2 Direct methods versus iterative methods

such that matrix B is an "easily invertible" matrix (B is nonsingular). Substituting this splitting in the original system, results in

$$Bu + (A - B)u = f$$

When this is used to derive a iterative procedure for u^{i+1} using u^i, an obvious choice would be:

$$Bu^{i+1} + (A - B)u^i = f$$

and rewritten for u^{i+1} results in:

$$\begin{aligned} u^{i+1} &= u^i - B^{-1}(Au^i - f) \\ &= u^i + B^{-1}(f - Au^i) := u^i + B^{-1}r^i \end{aligned} \tag{3.2.1}$$

Here u^i denotes the approximation of u before iteration i and u^{i+1} denotes the approximation of u after iteration i. I denotes the identity, B^{-1} denotes the inverse of matrix B and r^i is called the *residual* (or defect).
Equation (3.2.1) is called a *basic iterative* (or stationary) method and different methods can be distinguished depending on the choice of the splitting.
The *Jacobi iteration* is obtained when the following splitting is chosen: $A = D - E$, with D the diagonal elements of A : $D = \text{diag}(A)$. In terms of matrix B: choose $B = D$.
The *Gauss-Seidel iteration* is obtained using the splitting $A = L - U$, with L lower and U strictly upper triangular matrices respectively. ($B = L$)

When equation (3.2.1) is rewritten as:

$$u^{i+1} = Qu^i + s, (i = 0, 1, 2, \ldots)$$

Q is called the *iteration matrix* and the spectral radius $\rho(Q)$ of a $(N \times N)$ matrix Q is defined as:

$$\rho(Q) = \max\{|\lambda| : \lambda \text{ eigenvalue of } Q\}$$

The iterative method will converge if the spectral radius of Q is smaller than one:

$$\rho(Q) < 1$$

Furthermore, convergence of a iterative method implies that the sequence $u^0, u^1, u^2, \ldots$ converges to a limit u, independent of the choice of u^0.
The spectral radius can also be seen as the *asymptotic convergence rate* (in some norm) of the iteration:

$$||u - u^{i+1}|| \leq \rho(Q)||u - u^i||, \quad \text{for } i \to \infty$$

This means that the closer $\rho(Q)$ is to one, the slower the convergence.
For the convergence of the Jacobi and Gauss-Seidel method for the two dimensional Poisson equation on a Cartesian unit square grid with Dirichlet boundary conditions, the following spectral radii can be derived (see Vuik and Oosterlee, ref.2):

- $\rho(Q_{JAC}) = cos(\pi h) = 1 - \frac{1}{2}(\pi h)^2 + O(h^4)$
- $\rho(Q_{GS}) = cos(\pi h)^2 = 1 - (\pi h)^2 + O(h^4)$

Note that these convergence factors are close to one and hence the result is a slowly converging method. Another disadvantage is that the total number of unknowns N is related to h as $N = (\frac{1}{h})^{dim}$ and therefore a decreasing h will lead to an increase of N[2]. Unfortunately, this is typical for basic iterative methods (e.g.: for the methods above. If $h = \frac{1}{64}$ then $\rho(Q_{JAC}) = 0.998$ and $\rho(Q_{GS}) = 0.9976$).
To speed up the convergence compared to basic iterative methods, *Krylov subspace methods* can be used and will be discussed in the following subsection.

3.2.2 Krylov subspace methods

Consider once again equation (3.2.1). When starting with an initial guess u^0, initial *residual* $r^0 := f - Au^0$ and proceeding with the first two steps of the iteration process, the following is obtained:

$$\begin{aligned}
&u^0 \\
&u^1 = u^0 + (B^{-1}r^0) \\
&u^2 = u^1 + (B^{-1}r^1) = u^0 + B^{-1}r^0 + B^{-1}(f - Au^0 - AB^{-1}r^0) \\
&\quad = u^0 + 2B^{-1}r^0 - B^{-1}AB^{-1}r^0 \\
&\vdots
\end{aligned}$$

[2] here dim denotes the dimension of the problem

This implies that

$$u^i \in u^0 + \text{span}\{B^{-1}r^0, B^{-1}A(B^{-1}r^0), \ldots, (B^{-1}A)^{i-1}(B^{-1}r^0)\}$$

and therefore the following definition is made.

Definition (Krylov space)
The subspace $\mathcal{K}^i(A; r^0) := \text{span}(r^0, Ar^0, A^2r^0, \ldots, A^{i-1}r^0)$ is called the *Krylov space* of dimension i corresponding to matrix A and initial residual r^0

Remark
u^i calculated by a basic iterative method is an element of $u^0 + \mathcal{K}^i(B^{-1}A; B^{-1}r^0)$

Krylov subspace methods work in the following way: they start with an initial (residual) vector r^0, use r^0 to compute vector Ar^0, use Ar^0 to compute vector A^2r^0, and so on. Proceeding in this way, Krylov subspace methods form an orthogonal basis[3] of the sequence of successive matrix powers times the initial (residual) vector. In the favorable case, the approximation to the solution is calculated by *minimizing the residual* over the subspace formed, in some norm to be specified. The conjugate gradient method (CG) and the generalized minimal residual method (GMRES) are examples of Krylov subspace methods that minimize the residual. Other examples of Krylov subspace methods are the biconjugate gradient method (Bi-CG), the Bi-CGSTAB (stabilized biconjugate gradient) method, the Lanczos method using the Lanczos iteration scheme for Hermitian matrices and the Arnoldi method using the Arnoldi iteration scheme for more general matrices. In the next two subsections the properties of *short* and *long recurrence* methods will be discussed. Short recurrence methods are characterized by their efficiency in storage requirements while long recurrence methods result in an optimal minimization of the residual. In theory, CG and full GMRES are finite methods and will converge in N iterations. In practice however, performing N iterations is not favorable. Therefore these methods are only useful if the number of iterations is significantly smaller than N.

3.2.2.1 Short recurrence methods

It is already remarked that the basis of the Krylov subspace $\mathcal{K}^i(A; r^0)$, formed by the vectors $r^0, Ar^0, A^2r^0, \ldots, A^{i-1}r^0$ is generally not useful in numerical computations. i.e. the basis is ill conditioned. A well conditioned, at best *orthogonal* basis is needed in order to prevent loss of information due to the repeated matrix-vector multiplications performed in an algorithm. Another

[3] Unfortunately the basis formed by this procedure is generally not usable for use in numerical computations

important aspect is the *efficiency* of a method. In order to be efficient, it is desirable to generate the orthogonal basis with a *short recurrence*. This means that in each iteration step only a few of the latest basis vectors are required to generate the new basis vector. One of the commonly discussed methods in the literature is the *Conjugate Gradient* (CG) method. When applying the CG method, the assumption must be made that the system matrix A is symmetric and positive definite (SPD). The CG method has the following nice properties:

1. the approximation of the CG method is an element of $\mathcal{K}^i(A; r^0)$
2. the error is minimized with respect to a certain norm: so there is a optimality property
3. CG methods use short recurrences

It can be shown that it is not possible to obtain other Krylov subspace methods which have all these properties for general matrices, as is the case in the current application. However, the idea behind CG methods forms the basis for deriving other Krylov methods. In this thesis the *Bi-Conjugate Gradient* (Bi-CG) method will be discussed as an example of a short recurrence method. Bi-CG is based on the non-symmetric Lanczos algorithm and relaxes the condition on the matrix to be positive definite.

This non-symmetric algorithm builds a pair of *bi-orthogonal* bases for the following two subspaces:

1. $\mathcal{K}^i(A, v^1) = \text{span}\{v^1, Av^1, \ldots, A^{i-1}v^1\} := \mathcal{K}^i$
2. $\mathcal{K}^i(A^T, w^1) = \text{span}\{w^1, A^Tw^1, \ldots, (A^T)^{i-1}w^1\} := \mathcal{L}^i$

Here v^1 and w^1 are vectors and i is as before the dimension of the Krylov subspace. To see the algorithm that achieves the two bases, the reader is referred to Saad (ref. 26, Chapter 7).
The Bi-CG method can be derived from the Lanczos algorithm and the Bi-CG algorithm is a projection process onto $\mathcal{K}^i$ orthogonal to $\mathcal{L}^i$.
Here $v^1 = \dfrac{r^0}{||r^0||_2}$ and w^1 is arbitrary provided that $(v^1, w^1) \neq 0$, but is often chosen to be equal to v^1. In the following setup the necessary steps to derive the Bi-CG algorithm will be outlined:

- let the matrix V_i be the $(N \times i)$ matrix with columns $v^1 \ldots, v^i$
- write the LU decomposition of a tri-diagonal matrix $T_i := V_i^T A V_i$ as $T_i = L_i U_i$. Here the matrix L_i is unit lower bi-diagonal and U_i is upper bi-diagonal
- define $P_i := V_i U_i^{-1}$
- β_i are scalars obtained from the Lanczos algorithm
- let e_i be the i-th column of the $(N \times N)$ identity matrix

- express the solution u^i as:

$$\begin{aligned} u^i &= u^0 + V_i T_i^{-1}(\beta e_1) \\ &= u^0 + V_i U_i^{-1} L_i^{-1}(\beta e_1) \\ &= u^0 + P_i L_i^{-1}(\beta e_1) \end{aligned} \tag{3.2.2}$$

- when the *dual system* $A^T u^* = f^*$ is solved, then w^1 is obtained by scaling the initial residual $r^{*0} := f^* - A^T u^{*0}$. With this construction the vectors r^j and r^{*j} are in the same direction as v^{j+1} and w^{j+1} respectively and hence form a bi-orthogonal sequence
- in the same way as P_i define the matrix $P_i^* := W_i L_i^{-T}$
- with these definitions the column vectors p^{*i} of P_i^* and p^i of P_i are A-conjugate:

$$(P_i^*)^T A P_i = L_i^{-1} W_i^T A V_i U_i^{-1} = L_i^{-1} T_i U_i^{-1} = I$$

Finally, see Saad (ref. 26) the Bi-CG algorithm is given by:

Biconjugate Gradient Method

- Compute $r^0 := f - Au^0$ for some initial guess u^0 and
  choose r^{*0} such that $(r^0, r^{*0}) \neq 0$ (e.g. $r^{*0} = r^0$)
- Set $p^0 := r^0$ and $p^{*0} := r^{*0}$
- For $j = 0, 1, \ldots,$ until convergence, Do

$$\begin{aligned} \alpha^j &:= \frac{(r^j, r^{*j})}{(Ap^j, p^{*j})} \\ u^{j+1} &:= u^j + \alpha^j p^j \\ r^{j+1} &:= r^j - \alpha^j A p^j \\ r^{*j+1} &:= r^{*j} - \alpha^j A^T p^{*j} \\ \beta^j &:= \frac{(r^{j+1}, r^{*j+1})}{(r^j, r^{*j})} \\ p^{j+1} &:= r^{j+1} + \beta^j p^j \\ p^{*j+1} &:= r^{*j+1} + \beta^j p^{*j} \end{aligned}$$

- EndDo

Remark

When one is interested in the dual system with A^T, then instead of defining $r^0 := f - Au^0$, one should define $r^{*0} := f^* - A^T u^{*0}$ and the update $u^{*j+1} := u^{*j} + \alpha^j p^{*j}$ should be inserted after the update of u^{j+1}.

Proposition (Saad, ref. 26)
The vectors produced by the Bi-CG algorithm satisfy the following orthogonality properties:

$$(r^j, r^{*j}) = 0, \text{ for } i \neq j$$
$$(Ap^j, p^{*j}) = 0, \text{ for } i \neq j$$

This subsection will be concluded with some remarks concerning the Bi-CG method and in the next section long recurrence methods will be discussed.

1. In general it cannot be guaranteed that there exists an LU decomposition of the tri-diagonal matrix T_i. In that case, the Bi-CG algorithm will breakdown. This breakdown can be avoided in the Bi-Lanczos formulation of this Bi-CG scheme. The algorithm is not reproduced here and the reader is referred to Barrett et al. (ref. 17, page 22).
2. There is a recursive update of the solution vector, which avoids storage of the intermediate vectors r^0 and r^{*0}. Hence this method is indeed a short recurrence method.
3. The Bi-CG method also solves the dual system, which is usually not required. Therefore this dual system is often ignored in the formulations of the algorithms.
4. Bi-CG has short recurrences, but does not minimizes the residual.
5. Two other commonly used methods that can be derived from the Bi-CG method are the *CGS* () and the *Bi-CGSTAB* method. For more about these methods the reader is referred to Vuik and Oosterlee (ref. **?** , p. 81-82).

3.2.2.2 Long recurrence methods

To explain the idea of *long recurrence methods*, the so called *GMRES* type methods will be used in this thesis. GMRES stands for Generalized Minimal Residual Method. The long recurrences imply that the amount of work and the required memory *per iteration* grow for an increasing number of iterations. Therefore, in practice, it is not possible to use the full algorithm (full GMRES), and instead two variations on this method, namely *restarted* or *truncated* GMRES are used. These two variations will be shortly discussed after the *basic* GMRES algorithm is explained.

The GMRES method is a projection method and therefore there are two subspaces needed, say $\mathcal{K}$ and $\mathcal{L}$, with $\dim(\mathcal{L}) < \dim(\mathcal{K})$. More specifically: the residual vector $r := f - Au$ is constrained to be orthogonal to m linearly independent vectors. For more details about projection methods the reader is encouraged to read Chapter 5 from Saad (ref. 26).

The GMRES method characterizes itself by the following choices for the two subspaces mentioned before: $\mathcal{K} = \mathcal{K}_m$ and $\mathcal{L} = A\mathcal{K}_m$ where $\mathcal{K}_m$ is the Krylov subspace with dimension m and

v^1 as in the Bi-CG method defined as: $v^i = \frac{r^{i-1}}{||r^{i-1}||_2}$. With these choices, the GMRES method will minimize the residual norm over all vectors in $\{u^0 + \mathcal{K}_m\}$. As in the previous subsection, first the setup will be given and then the template of the GMRES method:

- let the matrix V_m be the $(N \times m)$ matrix with columns $v^1 \ldots, v^m$. The columns are required to be independent and orthonormal
- let y be a vector of length m. Than for any vector $u \in \{u^0 + \mathcal{K}_m\}$ it holds that it can be written as: $u = u^0 + V_m y$ $(\star_1)$
- define $J(y) := ||f - Au||_2 = ||f - A(u^0 + V_m y)||_2$ $(\star_2)$
- using the definition of $J(y)$ it holds that:

$$\begin{aligned} f - Au &= f - A(u^0 + V_m y \\ &= r^0 - AV_m y \\ &= \beta v^1 - V_{m+1}\bar{H}_m y \\ &= V_{m+1}(\beta e_1 - \bar{H}_m y) \end{aligned}$$

- since the columns of V_m are independent, they span the Krylov subspace. As they are also orthonormal, $J(y)$ can be rewritten as:

$$J(y) := ||f - A(u^0 + V_m y)||_2 = ||\beta e_1 - \bar{H}_m y||_2 \qquad (\star_3)$$

- the GMRES approximation is the unique vector $u \in \{u^0 + \mathcal{K}_m\}$ that minimizes $(\star_2)$
- using $(\star_1)$ and $(\star_3)$ the GMRES approximation can be obtained as:

$$\begin{aligned} &u^m = u^0 + V_m y_m \\ &y_m = \text{argmin}_y ||\beta e_1 - \bar{H}_m y||_2 \qquad (y_m \text{ minimizes the function } J(y)) \end{aligned}$$

Finally, the GMRES method is given as:

Generalized Minimal Residual Method

- Compute $r^0 := f - Au^0$ for some initial guess u^0, $\beta := ||r^0||_2$ and $v^1 := \frac{r^0}{\beta}$
- For $j = 1, 2, \ldots, m$, Do
  - Compute $w^j := Av^j$
  - For $i = 1, 2, \ldots, j$ Do
    - $h_{ij} := (w^j, v^i)$
    - $w^j := w^j - h_{ij}v^i$
  - EndDo
  - $h_{j+1,j} = ||w^j||_2$. If $h_{j+1,j} = 0$ set $m := j$ and proceed with defining the Hessenberg matrix
  - $v^{j+1} = \frac{w^j}{h_{j+1,j}}$
- EndDo
- Define the $(m+1) \times m$ Hessenberg matrix $\bar{H}_m = \{h_{ij}\}_{1 \leq i \leq m+1, 1 \leq j \leq m}$
- Compute minimizer of $||\beta e_1 - \bar{H}_m y||_2 : y_m$ and $u^m = u^0 + V_m y_m$

An important difference in contrast to the Bi-CG method is that the GMRES method is a stable method and breakdowns as in the case of Bi-CG can not occur. There are no denominators to give rise to problems. The case $\beta = 0$ means that the residual equals zero, hence the problem would be solved. Therefore it can be assumed that $\beta \neq 0$. If $h_{j+1,j} = 0, v^{j+1}$ will not be calculated, hence also in this case there will be no problem with a denominator being zero. Note that if $h_{j+1,j} = 0$, then $u^j = u$, and in this case there is a "lucky" breakdown (also see Proposition 6.10 in Saad (ref. 26)). In the following, a theorem will be reproduced from Vuik and Oosterlee (ref. 2), and the proof can be found in Saad and Schultz (ref. 25, 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. 25)
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 $\sigma = \{\lambda_1, \ldots, \lambda_N\}$ represent the spectrum of A. Define:

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

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

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*

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

This subsection will be concluded with the restarted GMRES and the truncated version (these processes concern the Arnoldi orthogonalization procedure). Recall that a long recurrence method becomes impractical when the number of iterations becomes very large: the memory and computational requirements increases with the number of iterations (see Table 3).

Restarted GMRES	**Truncated GMRES**
1. Compute $r^0 := f - Au^0, u^0, \beta := \|\|r^0\|\|_2$ and $v^1 := \frac{r^0}{\beta}$ 2. Generate the Arnoldi basis and the matrix $\bar{H}_m$ using the Arnoldi algorithm starting with v^1 3. Compute y_m which minimizes $\|\|\beta e_1 - \bar{H}_m y\|\|_2$ and $u^m = u^0 + V_m y_m$ 4. If satisfied then STOP, else set $u^0 := u^m$ and go to step 1	Run a modification of the GMRES algorithm in which the Arnoldi process is replaced by the incomplete orthogonalization process and all other computations remain unchanged

Table 3 Restarted GMRES and truncated GMRES

A great disadvantage of the restarted GMRES is that this method fails when the matrix is not positive definite, as is the case in the current application (see Saad, ref. 26, p. 172). The truncated version may save computational work, but will not optimize the storage: also in this version all the vectors v^i must be stored. Hence, GMRES type methods are no option to solve the linear system obtained in the current application. For a more elaborate discussion of these methods, the reader is referred to Saad (ref. 26).

3.2.2.3 Induced dimension reduction methods

In this subsection the recent work of Sonneveld and Van Gijzen (ref. 19) will be briefly discussed. Their work concerns the *Induced Dimension Reduction methods* (IDR) and these methods combine the efficiency of the short recurrence methods with the fact that they can compute the exact solution of a linear system of dimension N, in at most $2N$ steps (matrix-vector multiplications) in exact arithmetic.

The IDR methods are based on the Induced Dimension Reduction theorem, which is stated below:

Theorem (IDR) *Let A be any matrix in $\mathbb{C}^{N\times N}$, let $\boldsymbol{v}^0$ be any nonzero vector in $\mathbb{C}^N$ and let $\mathcal{G}_0$ be the complete Krylov space $\mathcal{K}^N(A,\boldsymbol{v}^0)$. Let $\mathcal{S}$ denote any proper subspace of $\mathbb{C}^N$ such that $\mathcal{S}$ and $\mathcal{G}_0$ does not share a nontrivial invariant subspace of A. Define the sequence $\mathcal{G}_j, j = 1, 2, \ldots$ as*

$$\mathcal{G}_j = (I - \omega_j A)(\mathcal{G}_{j-1} \cap \mathcal{S})$$

where the $\omega_j's$ are nonzero scalars. Then:
(i) $\mathcal{G}_j \subset \mathcal{G}_{j-1}$, for all $j > 0$
(ii) $\mathcal{G}_j = 0$, for some $j \leq N$

Proof. Uses induction: see Sonneveld and Van Gijzen (ref. 19, page 3). □

In fact, this theorem states that it is possible to generate a sequence of nested subspaces of *decreasing dimension* and that, under mild conditions, the smallest possible subspace is $\{0\}$. As the template of the algorithm can be seen in Sonneveld et al. (ref. 19, page 6), it will not be copied here. Instead some remarks will be listed:

- The IDR algorithm as stated by Sonneveld et al. may break down in two ways. For both types of breakdowns suggestions are made in their work to repair this.
- The IDR algorithm can be used to make an extension to the so called IDR(s) algorithms. In these type of methods s is a parameter greater than zero, used to make a distinction between several types of IDR methods (e.g. IDR(1) and Bi-CGSTAB are mathematically equivalent).
- The maximum number of matrix-vector products to reach the exact solution for Bi-CG type methods is $2N$. For IDR(s) the maximum number is $(N + \frac{N}{s})$ and increasing s will lead to a faster convergence.
- Unfortunately, like other Krylov subspace methods, also IDR(s) methods are sensitive to round-off errors.
- The resulting family of IDR(s) uses short recurrences and hence a limited amount of memory.
- For problems with a non-symmetric or indefinite system matrix, IDR(s) is an efficient method. For illustration: for a 3D-Helmholtz type problem, Sonneveld et al. (ref. 19) compared Bi-CGSTAB and IDR(6). It turned out that IDR(6) outperformed Bi-CGSTAB by a factor of six.

3.3 Preconditioning

In general Krylov subspace methods will show slower convergence behavior, when directly applied to linear systems of equations derived from applications in fluid dynamics or electromagnetics. This is also the case in the current application with the Maxwell equations. This convergence behavior strongly depends on the eigenvalue distribution of the coefficient matrix A. The great disadvantage is that the system matrix is ill-conditioned and the eigenvalue spectrum is not favorable for the convergence of the Krylov subspace methods. To improve the efficiency, convergence and robustness of the iterative methods, *preconditioning* can be used. In this subsection the general idea of preconditioning will be discussed using GMRES instead of Bi-CG[4] where after some classical preconditioning techniques will be discussed.

A general description of preconditioning is a transformation of the original linear system into one with the same solution, but easier to solve with an iterative method. This transformation will be achieved by using a nonsingular preconditioning matrix $M \in \mathbb{C}^{N \times N}$, to which the following requirements are made:

- the eigenvalues of $M^{-1}A$ should be clustered around 1 (in order to obtain a more favorable eigenvalue spectrum)
- it should be possible to obtain $M^{-1}y$ at low cost (for a vector $y \in \mathbb{C}^N$).

Here the original system will be denoted by $Au = f$ and for the preconditioned system there are three known possibilities:

1. *left preconditioning*: the preconditioner can be applied from the left of the original matrix A, resulting in
$$M^{-1}Au = M^{-1}f$$
2. *right preconditioning*: the preconditioner can be applied from the right of the original matrix A, resulting in
$$AM^{-1}x = f, x := Mu$$
3. *split preconditioning*: when the preconditioner can be factored as $M = M_L M_R$, with M_L and M_R triangular matrices, the preconditioned system looks like:
$$M_L^{-1}AM_R^{-1}x, x := Mu$$

Also for GMRES (or other non-symmetric iterative solvers), the above three options can be applied. An important observation is that the right preconditioned version gives rise to a so called *flexible variant*: the preconditioner is allowed to change at each step. This can be very useful in

[4]Note that for the current application there is a high interest in short recurrence methods because of the efficiency in the storage requirements. For the long recurrence methods all the basis vectors must be stored.

some applications. This is not generally used and for more on left, right and flexible GMRES, the reader is referred to Langou et al (ref. 11 and 21).
In GMRES the Arnoldi loop is used, and for e.g. the left preconditioned system, the Krylov subspace is constructed such that the approximate solution is found by constructing the iterants u^j in such a way that:

$$u^j \in u^0 + K^i(M^{-1}A, r^0), \text{ with } i > 1 \text{ the dimension of the subspace}$$

For a detailed description of the preconditioned GMRES, the reader is referred to Saad (ref. 26, Subsection 9.3).

3.3.1 Classical preconditioning techniques

In this subsection some remarks about the choices of the preconditioning matrix M will be outlined. One of the simplest ways is to perform an *incomplete LU factorization* of the original system matrix A. This decomposition can be written as $A = LU - R$, where L and U have the same nonzero structure as the lower and upper parts of A respectively and R is the residual or error matrix of the factorization. In the literature this incomplete factorization is referred to as ILU(0). The advantage of this method is that it is easy and cheap to compute; the disadvantage is that generally it is not very effective for the current application. Also for e.g. ILU(0.01) (Erlangga, ref. 7, p. 38), the storage requirements can become unacceptably high and the convergence is sensitive to the gridsize. Other preconditioners use successive overrelaxation (SOR), symmetric SOR (SSOR), the separation of variables preconditioner (Erlangga, ref. 7) and the approximate inverse preconditioner (Hooghiemstra, ref. 8, p. 94).

It has been shown by Erlangga (ref. 7) and Hooghiemstra (ref. 8) that using the ILU or the approximate inverse as preconditioner does not significantly improve the convergence of the iterative method used, to solve the system of equations derived from Helmholtz or Maxwell equation(s). Therefore these classical preconditioning techniques will not be used in the current application. In the next section a special class of preconditioners for the Maxwell equations will we discussed, namely the *shifted Laplace preconditioners*. To be able to solve the shifted Laplace preconditioner efficiently, multigrid solution methods are used. These will be discussed in the next chapter.

3.3.2 The shifted Laplace preconditioner

Due to the disadvantages mentioned in the previous section, a lot of research has been done for preconditioners applied to indefinite linear systems. In the work of Erlangga (ref. 7), the shifted Laplace operator has been discussed as preconditioner for the Helmholtz equation. As the Maxwell equations are the vector form of the Helmholtz equation, it is expected that the shifted

Laplace preconditioning technique will also be an effective preconditioner for the discretized Maxwell equations. This is supported by the results obtained by Hooghiemstra (ref. 8 and 9). Recall the dimensionless vector wave equation:

$$\bigtriangledown \times \bigtriangledown \times \mathbf{E} - k_0^2 \mathbf{E} = -\imath k_0 Z_0 \mathbf{J} \tag{3.3.1}$$

The class of preconditioners in this section is constructed by discretization of the following *shifted Laplace* operator:

$$\mathcal{M}_{(\beta_1,\beta_2)} := -\bigtriangleup -(\beta_1 - \imath\beta_2)k_0^2, \qquad \beta_1, \beta_2 \in \mathbb{R} \text{ and } \imath^2 = -1. \tag{3.3.2}$$

By using various combinations of (β_1, β_2), several preconditioners can be constructed in order to improve the convergence of an iterative method. In Erlangga (ref. 7, Section 4.1), the one dimensional Helmholtz equation is considered with a real $(\beta_2 = 0)$ and a complex shift. In Section 4.2 of ref. 7 the spectral properties of the shifted Laplace preconditioner are analyzed. The analysis shows that, in the one dimensional case with real shift, the leading part of the Helmholtz equation (the Laplacian term) is advisable for small wavenumbers[5]. When the wavenumber becomes very large, the condition number of the preconditioned system becomes very large which is not desirable. Setting $\beta_1 = -1$ leads to a fast converging preconditioned iterative method. When the complex shift is analyzed, it appeared that for $\beta_1 \leq 0, \beta_2 \in \mathbb{R}, \mathcal{M}_{0,1}$ is an optimal preconditioning operator for the Helmholtz equation. For both the real and complex shift, the preconditioner is used in combination with the CG method, applied to the normal equations.
Erlangga (ref. 7) also performed GMRES convergence analysis for the discrete two dimensional Helmholtz equation, preconditioned by the shifted Laplace operator and several types of boundary conditions are considered.
One final remark from the work of Erlangga (ref. 7) concerns the h-dependence of the convergence behavior for the shifted Laplace preconditioner. It has been shown that this is mainly determined by the smallest eigenvalue of the operator being considered. In Section 4.4 in Erlangga (ref. 7) the convergence behavior is considered by using GMRES as iterative solver on different grid sizes. It appears that different values of h do not affect the number of iterations needed to get satisfying convergence.
In extension of the spectral analysis performed in Section 4.2 of Erlangga (ref. 7), Van Gijzen, Erlangga and Vuik (ref. 16) introduced a different approach to the spectral analysis of the discrete Helmholtz operator preconditioned by the shifted Laplace operator. They showed that the eigenvalues of the preconditioned matrix are located either *in* or *on* a circle, or in a half-plane. This location of the eigenvalues of course depends on the value of the shift.

[5] this one dimensional case is considered for analytical purpose which motivates the development of the shifted Laplace preconditioner

As in the current application the system matrix also consists of a fully populated block, hence not fully sparse, direct application of the shifted Laplace preconditioner is not efficient. To handle the fully populated block originating from the discretization of the boundary conditions, another type of preconditioner will be used. These so called *block preconditioner* will be discussed in the next subsection.

3.3.3 Block preconditioner

The system matrix A of the current application has a special block structure (recall Figure 2.4.1). However, before the idea of block preconditioning can be explained, the concept of the *Schur complement* is defined first:

Definition (Schur complement)
Let A_{11}, A_{12}, A_{21} and A_{22} be $(n \times n), (m \times n), (n \times m)$ and $(m \times m)$ matrices respectively and assume that A_{11} is invertible (note that $n + m = N$). Write

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \in \mathbb{R}^{(n+m)\times(n+m)} \tag{3.3.3}$$

Then the *Schur complement of block matrix* A_{11} is defined as the $(m \times m)$ matrix S where

$$S = A_{22} - A_{21}A_{11}^{-1}A_{12} \tag{3.3.4}$$

Schur complements originate from the Gaussian elimination procedure applied to block matrices. Note that when the matrix A is defined as in equation (3.3.3), the block LU decomposition can be written as

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} I_n & 0 \\ A_{21}A_{11}^{-1} & I_m \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ 0 & S \end{bmatrix} \tag{3.3.5}$$

The block preconditioner that can be used looks like

$$P_T = \begin{bmatrix} A_{11} & A_{12} \\ 0 & S \end{bmatrix} \tag{3.3.6}$$

The block preconditioner P_T that can be used, results in a favorable spectrum for AP_T^{-1} (see Benzi, ref. 1). He has shown that the spectrum of the matrix AP_T^{-1} is equal to

$$\sigma(AP_T^{-1}) = \sigma(P_T^{-1}A) = \{1\}.$$

Hence the only eigenvalue is $\lambda = 1$ with multiplicity $n + m$. So the requirement on a preconditioning matrix to cluster the eigenvalues around 1 has been fulfilled. However, it should be remarked that the matrix AP_T^{-1} is not normal (for a normal matrix B it holds that $B^T B = BB^T$). Therefore there is no basis of eigenvectors and it is very difficult to apply spectral/convergence analysis (for e.g. GMRES) on AP_T^{-1}. The matrix AP_T^{-1} is also not symmetric and therefore there will be no convergence in one iteration, although the eigenvalues are clustered around 1 (for a symmetric matrix B it holds that $B = B^T$).

In the remainder of this section the system matrix of the current application will be considered, preconditioned by the shifted Laplace operator combined with the block preconditioner P_T. This is done analogous to Hooghiemstra (ref. 9, chapter 4).

Note that A has a block structure similar as equation (3.3.3) where the blocks A_{11}, A_{12} and A_{21} are sparsely populated and the block A_{22} is fully populated. As AP_T^{-1} uses the inverse from A_{11} which is not desirable, Hooghiemstra (ref. 9) tested several block preconditioners to approximate P_T^{-1}, which are included below.
The first three preconditioners have two diagonal blocks:

$$M_I = \begin{bmatrix} A_{11} & 0 \\ 0 & I_{22} \end{bmatrix}, M_{II} = \begin{bmatrix} I_{11} & 0 \\ 0 & A_{22} \end{bmatrix}, M_{III} \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix} \tag{3.3.7}$$

The last three preconditioners use an extra off-diagonal block:

$$M_{IV} = \begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix}, M_V = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}, M_{VI} = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} - A_{11}(\text{diag}(A_{11})^{-1})A_{12} \end{bmatrix} \tag{3.3.8}$$

Hooghiemstra (ref. 9) tested the first five preconditioners in Matlab and concluded that preconditioner M_V yields a fast converging method. Therefore the preconditioner that is chosen by Hooghiemstra is M_V. For simplicity this will be denoted by:

$$M_1 := M_V = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \tag{3.3.9}$$

Note that M_1 is exactly the Schur complement when the last term in (3.3.4) is rejected. Preconditioner M_{VI} is proposed as another preconditioner to approximate the Schur complement.

Since M_1 is an upper triangular block matrix, M_1 is also called a triangular preconditioner. To explain more about these block preconditioned system(s), the following notation is used:

- original system: $Ax = b$
- initial guess: x^0
- initial residual: $r^0 = b - Ax^0$
- residual i : r^i
- (first) preconditioned system: $M_1 s^k = r^{k-1} \qquad (\star_M)$

Note that $(\star_M)$ can be written as:

$$\begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \begin{pmatrix} s_1^k \\ s_2^k \end{pmatrix} \begin{pmatrix} r_1^{k-1} \\ r_2^{k-1} \end{pmatrix} \tag{3.3.10}$$

The first system to solve would be: $A_{22} s_2^k = r_2^{k-1}$ by applying an LU decomposition to the matrix A_{22}. As block A_{22} is very small compared to the other blocks, this decomposition is cheap to perform and has to be computed only once. Once s_2^k is solved, the next system to solve is:

$$A_{11} s_1^k = r_1^{k-1} - A_{12} s_2^k \tag{3.3.11}$$

The system in equation (3.3.11) is a very large system and is very expensive to compute with a direct solver. Therefore this system has to be solved using a second preconditioned Krylov subspace method. Two remarks about this system:

1. Since the search direction s_1^k in (3.3.11) is approximated, the number of outer iterations will increase. This increase can be bounded if the system is solved with a high accuracy (see Hooghiemstra, ref. 9, Chapter 5).
2. The system matrix A_{11} in (3.3.11) corresponds to the discretization of the vector wave equation *inside* the cavity. Therefore A_{11} is sparsely populated and is very similar to the matrix originating from the discretization of the Helmholtz equation obtained by Erlangga in reference 7.

As Erlangga (ref. 7) showed that the shifted Laplace preconditioner is an efficient preconditioner for the Helmholtz equation, Hooghiemstra (ref. 9) used the following preconditioner for the vector wave equation:

$$\mathcal{M}_{(\beta_1,\beta_2)} = -\frac{1}{\mu_r} \bigtriangledown^2 - \hat{k}_0^2 \varepsilon_r, \qquad \text{where } \beta_1, \beta_2 \in \mathbb{R}, \hat{k}_0 = (\beta_1 - i\beta_2) k_0 \text{ and } i^2 = -1. \tag{3.3.12}$$

The matrix obtained from the finite element discretization of (3.3.12), denoted by M_2, is used as preconditioner for system (3.3.11). It has already been discussed in Section 3.3.2 that the choice of (β_1, β_2) strongly influences the convergence behavior of the solution method applied to the

second preconditioner system (3.3.11). Choosing $(\beta_1, \beta_2) = (1, 0.5)$ yields a fast converging iterative method for the Helmholtz equation, if this system is solved using a multigrid method: see Chapter 6 from ref. 7 (Erlangga used multigrid as a very efficient solver for this second preconditioned system).

Summarizing

A (triangular) block preconditioner M_1 is constructed from the blocks of the original matrix A followed by a second preconditioner M_2 constructed to improve the convergence of the iterative method to solve the sparse matrix A_{11}. M_2 is obtained from the finite element discretization of (3.3.12).

The idea proposed by Hooghiemstra (ref. 9, p. 41) is to combine the triangular preconditioner with the shifted Laplace preconditioner, resulting in:

$$M_{new} = \begin{bmatrix} M_2 & A_{12} \\ 0 & A_{22} \end{bmatrix} \tag{3.3.13}$$

and the new preconditioned system to be solved is given by:

$$M_{new} s^k = r^{k-1} \tag{3.3.14}$$

This is also done in two steps:

1. solve $A_{22} s_2^k = r_2^{k-1}$ (3.3.15)

   with a precomputed LU decomposition of A_{22}(note that this decomposition has to be computed only once)

2. solve $M_2 s_1^k = r_1^{k-1} - A_{12} s_2^k$ (3.3.16)

   Hooghiemstra (ref. 9) used GCR to solve (3.3.16). In the current application multigrid will be used as solver for this preconditioner system.

This chapter will be concluded with the following discussion about the type of boundary condition used in the discretization. In Section 2.3.1 it has been explained that the matrix $[P^{st}]$ from equation (2.3.11) is obtained from the boundary integral and gives rise to the fully populated part in the system matrix A. Therefore these integral equations depend on the boundary conditions. In the current formulation a *global* radiation boundary condition is imposed on the aperture. The latter boundary condition will, within the accuracy of the discretization, make the electric field comply exactly with the theoretically correct far-field behavior and be completely transparent. Global refers here to the tight coupling of the degrees of freedom on the aperture, the discretization of which leads to a fully populated matrix.

Alternatively, a *local* boundary condition can be imposed. Local radiation boundary conditions of arbitrary order of accuracy can be derived, where the locality of the operator decreases with increasing order. Discretization of the local boundary condition leads to a sparsely populated matrix. Local boundary conditions are only accurate when imposed sufficiently removed from the domain of interest, most often by extending the computational domain.
However, an operator based on a local radiation boundary condition could be a viable alternative preconditioner to replace the currently used block preconditioning approach. It is recommended to further investigate the advantages of a local radiation boundary condition for the preconditioner matrix.
In the next chapter, multigrid methods to solve the preconditioner system (3.3.16) will be treated.

4 Multigrid methods

4.1 Introduction

As already seen in Chapter 3, Krylov subspace methods (KSMs) can be used to solve (linear) systems arising from the discretization of partial differential equations (PDEs). In most applications these methods perform better than basic iterative methods. One drawback, however, is that the KSMs show an unsatisfying convergence behavior as the number of unknowns N increases: Krylov subspace methods are said to be *dependent* on the mesh size h. The class of methods described in this chapter, the multigrid (MG) methods, are *independent* of the mesh size. Another advantage of these MG methods is that they can solve a linear system with N unknowns, in cN arithmetic operations (c is a constant).
A natural way to explain the basics of multigrid methods, is by analyzing *Geometric Multigrid*. Hence this will be discussed in this thesis, combined with the one dimensional Laplace equation, in Section 4.2 (referred to as Model Problem1). These results will be compared to the one dimensional Helmholtz equation (Model Problem2) and the conclusion will be made that multigrid can not be applied directly to the indefinite matrices which may arise in discretizing this equation, for large values of the wave number. As the Maxwell equations show very similar behavior, this will also be the case in the current application. Furthermore, in the examples discussed in this chapter, structured rectangular grids and finite differences are used, but in the current application however, an unstructured grid will be used. To handle unstructured grids, a specific multigrid type method will be used namely *Algebraic Multigrid*, combined with the finite element discretization method. It will also be explained how AMG can be incorporated in the current algorithm for the solution of an indefinite system, by means of a definite preconditioner. Since it is not likely that, due to lack of time, algebraic multigrid can be implemented in the existing algorithm, other solutions has to be sought. It is known that there are several AMG black box solvers available. Therefore Subsection 4.6 considers some requirements on an AMG black box solver for the discretized vector wave equation.

Before proceeding with multigrid methods in the next section, Model Problem1 and 2 are stated below. These model problems will be used to explain the components of (geometric) multigrid.

Model Problem1

The discrete Poisson equation in one dimension with Dirichlet boundary conditions:

$$-\triangle_h u_h(x) = -u_{xx} = f_h^{\Omega}(x), x \in \Omega_h \tag{4.1.1}$$
$$u_h(x) = f_h^{\Gamma}(x), x \in \Gamma_h = \partial\Omega_h, \text{ where}$$

- $\Omega = (0,1) \subset \mathbb{R}$
- $\Gamma = \{0\} \cup \{1\}$
- $n \in \mathbb{N} \rightarrow h = \dfrac{1}{n}$
- finite differences approximation of the partial differential operator L: $L_h = -\triangle_h$
- $L_h = \dfrac{1}{h^2}\begin{bmatrix} -1 & 2 & -1 \end{bmatrix}$

Model Problem2

The discrete Helmholtz equation in one dimension with Dirichlet boundary conditions:

$$-(\triangle_h + k_0^2)u_h(x) = -u_{xx} - k_0^2 u = f_h^{\Omega}(x), x \in \Omega_h \tag{4.1.2}$$
$$u_h(x) = f_h^{\Gamma}(x), x \in \Gamma_h = \partial\Omega_h, \text{ where}$$

- $\Omega = (0,1) \subset \mathbb{R}$
- $\Gamma = \{0\} \cup \{1\}$
- $n \in \mathbb{N} \rightarrow h = \dfrac{1}{n}$
- k_0 is the dimensionless wavenumber
- finite differences approximation of the partial differential operator $\tilde{L}$: $\tilde{L}_h = -\tilde{\triangle}_h$
- $\tilde{L}_h = \dfrac{1}{h^2}\begin{bmatrix} -1 & 2 - k_0^2 h^2 & -1 \end{bmatrix}$

4.2 Motivation of multigrid methods

From the theory of basic iterative methods, it is known that the success of these methods lies in the *relaxation* of one or more coordinates in the residual (error)vector and that they are terminated when satisfactory convergence is obtained. MG techniques take advantage of *faster convergence* of the components of the residuals in a specific direction i.e. the direction of those eigenvectors of the iteration matrix corresponding to the largest eigenvalues in the spectrum of the iteration matrix. These eigenvectors are known as the oscillatory or high[1] frequency modes. The low frequency (or smooth) modes can not be efficiently damped on the fine grid, because they converge much slower than the high frequency modes. This is the main reason for basic iterative methods to converge slow when the systems to solve become larger and larger.
To overcome this difficulty, MG methods use several representations of the error (or residual) vector on different meshes. Starting with a fine grid and corresponding mesh size h, MG methods can recursively repeat this process called *error-smoothing* on a *coarser grid,* with e.g. mesh

[1] For the definition of high and low frequency modes, the reader is referred to (4.2.7)

size $H = 2h$. The consequence of this coarsening is that the low frequency modes on the fine grid are naturally mapped into high frequency modes on the coarser mesh and then the process can be repeated. This will be outlined in Subsection 4.2.1. Mappings between representations of the error vector on different grids by means of so called *Transfer Operators* will be the subject of Subsection 4.2.3. The remarks above concerning the convergence behavior of the MG methods will be the subject of Section 4.4.

4.2.1 Error smoothing analysis

The multigrid approach is based on two main aspects:

1. error smoothing
2. coarse grid correction

In this subsection some useful properties of MG methods will be discussed and later in this chapter these two aspects are defined more precisely. To begin, the following two principles are stated (from Trottenberg et al. (ref. 20, page 16):

Smoothing principle Many classical iterative methods (Gauss-Seidel etc.), if appropriately applied to discrete elliptic problems, have a strong smoothing effect on the error of any approximation

Coarse grid principle A smooth error term is well approximated on a coarse grid. A coarse grid procedure is substantially less expensive (substantially fewer grid points) than a fine grid problem

To explain the idea of error smoothing, two classical iteration methods known as *Gauss-Seidel* and *Jacobi* type methods will be discussed (used in this context these methods are also referred to as relaxation or smoothing methods).

The first is the *lexicographical* (see Figure 4.2.1) Gauss-Seidel method (GS-LEX) for the one dimensional Poisson equation. The grid function oriented notation is used and some notational issues are given:

- the differential operator: L
- mesh size h with corresponding grid Ω_h, e.g. $\Omega_h = (0, h, 2h, \ldots, Nh)$, with N the total number of unknowns
- the approximation of L: L_h with the assumption that L_h^{-1} exists
- $x_i \in \Omega_h$
- discrete (linear) boundary value problem: $L_h u_h = f_h$ in Ω_h
- approximation of $u_h(x_i)$ *before* an iteration: u_h^m
- approximation of $u_h(x_i)$ *after* an iteration: u_h^{m+1}

- the *error*: $v_h^m(x_i) = u_h(x_i) - u_h^m(x_i)$
- the identity operator on Ω_h : I_h
- relaxation parameter θ

Fig. 4.2.1 Left: lexicographic ordering of grid points – Right: red-black ordering of grid points

Using the notation above, the following iteration formula for GS-LEX is obtained:

$$z_h^{m+1}(x_i) = \frac{1}{2}[h^2 f_h(x_i) + u_h^{m+1}(x_i - h) + u_h^m(x_i + h)]$$
$$u_h^{m+1} = z_h^{m+1} \tag{4.2.1}$$

The iteration can also be formulated in terms of the error v in the following way:

$$v_h^{m+1}(x_i) = \frac{1}{2}[v_h^{m+1}(x_i - h) + v_h^m(x_i + h)] \tag{4.2.2}$$

Note that this is an averaging of the errors over the neighbors of point x_i.
The iteration formula for the Jacobi iteration looks like:

$$z_h^{m+1}(x_i) = \frac{1}{2}[h^2 f_h(x_i) + u_h^m(x_i - h) + u_h^m(x_i + h)]$$
$$u_h^{m+1} = z_h^{m+1} \tag{4.2.3}$$

When equation (4.2.3) is applied to Model Problem1, the error of the iteration becomes *smooth* after a few iteration steps, but not necessarily *smaller* (see Figures 4.2.2 and 4.2.3).

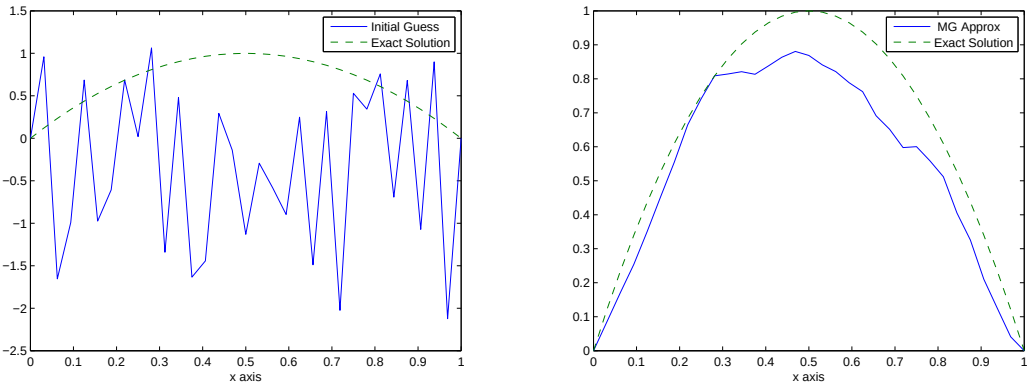

Fig. 4.2.2 The initial guess and the first iteration with Jacobi on a randomly chosen initial error. Iteration $\#1 : ||error||_\infty = 3.3222.10^{-01}$

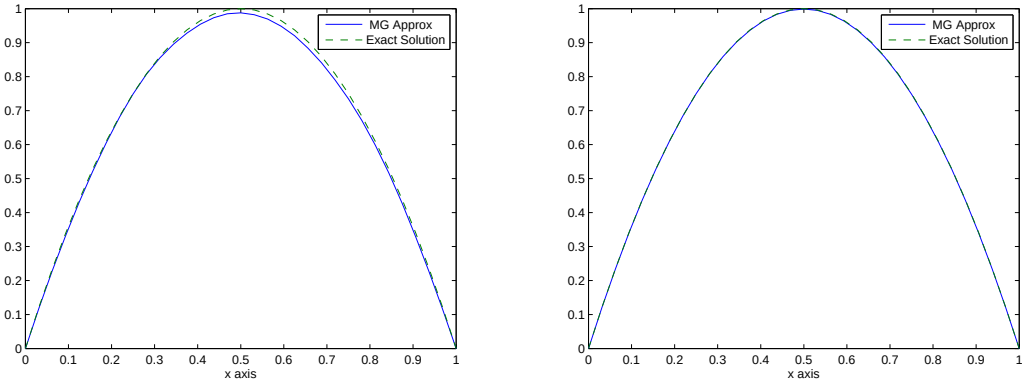

Fig. 4.2.3 The last two iterations: iteration $\#3 : ||error||_\infty = 1.1288.10^{-02}$ *and iteration* $\#4 : ||error||_\infty = 2.1015.10^{-03}$

To explain the further details of the error smoothing process, Model Problem1 and a Jacobi type iteration method will be used:

The Jacobi iteration can be rewritten as follows:

$$u_h^{m+1} = S_h u_h^m + \frac{h^2}{2} f_h, \text{ where the iteration operator } S_h = I_h - \frac{h^2}{2} L_h.$$

This iteration can be generalized by including a *relaxation parameter* ω: the so called ω*-(damped) Jacobi relaxation*, abbreviated by ω-JAC:

$$u_h^{m+1} = u_h^m + \omega(z_h^{m+1} - u_h^m) \tag{4.2.4}$$

The iteration operator for ω-JAC reads:

$$S_h(\omega) = I_h - \frac{\omega h^2}{2} L_h = \frac{\omega}{2} \begin{bmatrix} 1 & 2(\frac{2}{\omega} - 1) & 1 \end{bmatrix}_h$$

Recall that the subject of this section is not the convergence behavior of the error components, but the smoothing properties. Therefore the eigenfunctions[2] and eigenvalues of the iteration operator S_h are listed first and then the explanation of the smoothing properties follows:

$$\text{eigenfunctions of } S_h : \varphi_h^k = sin(k\pi x), x \in \Omega_h : k = 1, \ldots, n-1 \tag{4.2.5}$$

$$\text{corresponding eigenvalues of } S_h : \lambda_h^k = \lambda_h^k(\omega) = 1 - 2\omega(sin^2(k\pi h)) \tag{4.2.6}$$

Notes

1. The eigenvalues and the eigenfunctions of the iteration operator for ω-JAC are the same as the eigenvalues and eigenfunctions of L_h. This simplifies the analysis and therefore the

[2] actually: the j-th component of the k-th eigenfunction should be denoted by $\varphi_h^{k,j}, 1 \le k \le n-1, 0 \le j \le n$

analysis for this method will be included in this thesis. To see results for the Gauss-Seidel iteration matrix, the reader is referred to Briggs et al. (ref. 24, page 22).

2. When multigrid uses the ω-JAC as smoothing operator, S_h depends on the relaxation parameter ω.

In Section 4.2, some general remarks about high and low frequency modes were made. Important in the error smoothing process is the smoothing of highly oscillating eigenfunctions, i.e. the high frequency modes. To give an idea of how to define high and low frequencies, Model Problem1 will be used with the choice $H = 2h$. Consider the following two specific eigenfunctions:

$$\varphi^k, \varphi^{n-k}$$

It turns out that when these eigenfunctions are represented on the coarse grid Ω_H, they are related in the following sense:

$$\varphi^k(x) = -\varphi^{n-k}(x) \text{ for } x \in \Omega_H.$$

So for $k = \frac{n}{2}$, the eigenfunctions φ^k vanish on Ω_H. See Figures 4.2.4 and 4.2.5: here $n = 8$ and the eigenfunctions illustrated here are in context of Model Problem1. The coarse grid is denoted by '*'. Consider the left picture first: when the values of φ^7 are calculated on the coarse grid, the resulting function would be exactly $-\varphi^1$. The same holds for the right picture when the values of φ^6 are considered on the coarse grid points: the resulting line would be $-\varphi^2$.

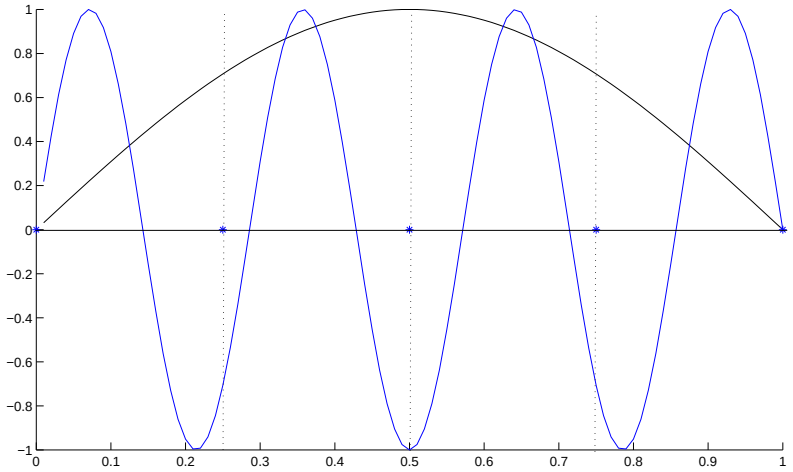

Fig. 4.2.4 φ^1 (black) and φ^7 (blue)

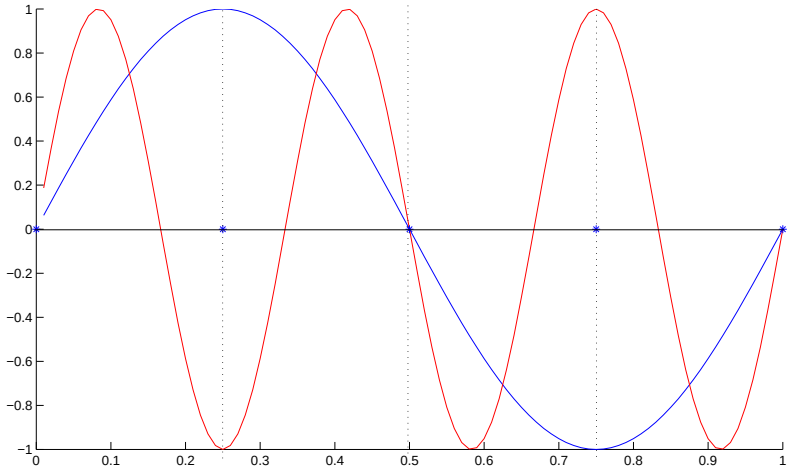

Fig. 4.2.5 φ^2 (blue) and φ^6 (red)

Remark The k-th mode consists of $\frac{k}{2}$ full sine waves and has a wavelength of $\ell = \frac{24h}{k} = \frac{2}{k}$ (when an interval of length 1 is considered); mode $k = \frac{n}{2}$ has wavelength $\ell = 4$ and mode $k = n - 1$ has wavelength $\ell = 2h$. Therefore waves with a wavenumber greater than $n (\rightarrow \ell < n)$, cannot be represented on the grid. Actually, waves with $\ell < 2h$ can be seen on a grid with a wavelength greater than $2h$.

Definition (in context of Model Problem1)
For $k \in \{1, \ldots, n-1\}$, φ^k denotes an eigenfunction (or component) of

- low frequency if $k < \dfrac{n}{2}$
- high frequency if $\dfrac{n}{2} \leq k < n$ (4.2.7)

Above definition is reason to the next classification for the eigenfunctions:
Let Ψ denote the collection of all the eigenfunctions. Let Ψ_h denote the collection of all high frequency and Ψ_l the collection of all low frequency eigenfunctions. Corresponding to Ψ_h and Ψ_l let the eigenvalues be elements of Θ_h and Θ_l. [3] Note that in defining the *(error) smoothing factor*, the largest eigenvalue λ_h^k corresponding to a eigenfunction with high frequency modes is important.

[3] $\Psi_h \cup \Psi_l = \Psi$. If Θ is the collection of all eigenvalues, then also $\Theta_h \cup \Theta_l = \Theta$

Therefore the following definition is made:

Definition (Smoothing factor of θ-JAC for Model Problem1)
The *smoothing factor* $\mu(h;\theta)$ of $S_h(\theta)$ is defined as:

$$\mu(h;\theta) := max\Big\{|\lambda_h^k(\theta)| \in \Theta_h\Big\} \tag{4.2.8}$$
$$\mu^*(\theta) := \sup_{h\in\mathcal{H}} \mu(h;\theta), \qquad \text{where } \mathcal{H} \text{ denotes the set of allowed mesh sizes}$$

The smoothing factor represents the worst factor by which the *high frequency* error components are reduced per relaxation step. So after ν smoothing steps, the amplitude of the high frequency components are reduced by a factor $(\mu^*(\theta))^\nu$ or smaller. So in order to guarantee damping (or smoothing) of the high frequency error components it must hold that $\mu^*(\theta) < 1$. [4]

To illustrate these definitions, consider the eigenvalues of S_h in equation (4.2.6). From the definition of the smoothing factor it follows that:

$$\Psi_h = \{\varphi_h^k = sin(k\pi x) : x \in \Omega_h \wedge \frac{n}{2} \leq k \leq n-1\}$$
$$\Theta_h = \{\lambda_h^k(\theta) = 1 - 2\theta(sin^2(k\pi h)) \wedge \frac{n}{2} \leq k \leq n-1\}$$
$$\mu(h;\theta) = \max\Big\{|1 - 2\theta(sin^2(k\pi h))| : \frac{n}{2} \leq k \leq n-1\Big\}$$
$$\mu^*(\theta) = \max\{|1-\theta|, |1-2\theta|\}$$

Note
When $\theta = 0$ or $\theta = 1$ then $\mu^*(\theta) = 1$

Recall that it is required to have $0 < \mu^*(\theta) < 1$, in order to have smoothing of the error.Therefore, if $\mu^*(\theta) \geq 1$ there is no smoothing and the following can be concluded in order to have *no* smoothing:

- $\mu^*(\theta) = |1-\theta| \geq 1 \Rightarrow \theta \leq 0 \vee \theta \geq 2$
- $\mu^*(\theta) = |1-2\theta| \geq 1 \Rightarrow \theta \leq 0 \vee \theta \geq 1$

In other words: to obtain a smoothing factor smaller than one for θ-JAC, choose $\theta \in (0,1)$. Also note that for λ_1 it holds that $\forall\theta \in (0,1)$:

$$\lambda_1 = 1 - 2\theta sin^2(\frac{\pi h}{2}) \approx 1 - \frac{\theta\pi^2 h^2}{2} \Rightarrow \lambda_1 \text{ close to } 1$$

[4] note that it is not necessary to use the absolute value of $\mu^*(\theta)$ because it is defined as the supremum of positive values

In fact, the discussion above stated that there is no value of θ that will reduce the smooth components of the error effectively. Also, the smaller the mesh size h, the closer λ_1 is to 1 $\Rightarrow$ improving the accuracy of the solution by decreasing the mesh size, will worsen the convergence of the smooth components of the error. On the other hand, when the mesh size is increased, as is the case on a coarser grid, λ_1 is bounded away from 1. Hence the convergence will improve on a coarser grid. This so called *coarse grid correction* will be the subject of Subsection 4.2.2

The discussion about the smoothing principle is ended with one remark:

Remark If e.g. $\mu^*(\frac{4}{5}) = \frac{3}{5}$, it follows that in one iteration step of θ-JAC with this choice of θ, a reduction of all high frequency error components (eigenfunctions) is obtained by at least a factor of $\frac{3}{5}$, *independent of the mesh size* h. This is an important property in the setup of the multigrid methods.

4.2.2 Coarse grid correction

In this subsection the coarse grid correction scheme will be explained. Before doing that, first some basic notation is given. The mesh size will be denoted by h, the corresponding grid by Ω_h and the discrete (linear) system is of the form:

$$L_h u_h = f_h, \text{ on grid } \Omega_h \tag{4.2.9}$$

In the following subsections the following notation will be used:

- the operator L_H on a coarser grid Ω_H (e.g.: mesh size $H = 2h$), with the assumptions that $L_H : \mathcal{G}(\Omega_H) \rightarrow \mathcal{G}(\Omega_H)$ with dim $\mathcal{G}(\Omega_H) <$ dim $\mathcal{G}(\Omega_h)$ and that L_H^{-1} exists
- the (linear) *restriction* operator $I_h^H : \mathcal{G}(\Omega_h) \rightarrow \mathcal{G}(\Omega_H)$
- the (linear) *prolongation* (or interpolation) operator $I_H^h : \mathcal{G}(\Omega_H) \rightarrow \mathcal{G}(\Omega_h)$

The basic idea in MG methods is that the (linear) system is (repeatedly) solved on several grids: start on a fine grid Ω_h and proceed on the coarser grid. See Figure 4.2.6.
Below the coarse grid correction steps are included (see Trottenberg et al., ref. 20, page 37):

Coarse Grid Correction $u_h^m \rightarrow u_h^{m+1}$	
- 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	$u_h^{m+1} = u_h^m + \hat{v}_h^m$

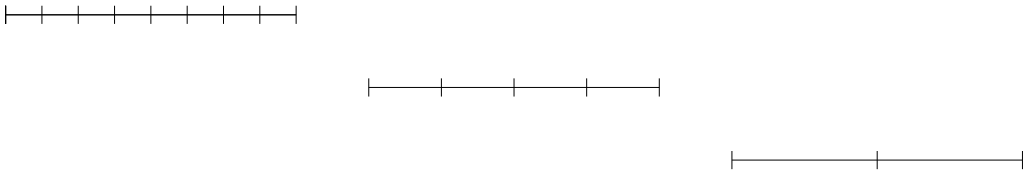

Fig. 4.2.6 A sequence of grids starting with $h = \frac{1}{8}$

This subsection is concluded with some remarks about the choice of the coarse grid operator L_H. One obvious choice is to use the operator L_h as basis and use that on grid Ω_H. In terms of Model Problem1:

$$L_H = \frac{1}{H^2}\begin{bmatrix} -1 & 2 & -1 \end{bmatrix}_H$$

This is a natural choice and for most basic MG methods a good operator. For other MG methods however, different coarse grid operators perform much better. One of them is the *Galerkin coarse grid operator* and is defined as:

$$L_H := I_h^H L_h I_H^h \tag{4.2.10}$$

A discussion of the Galerkin operator will be postponed until Section 4.6.
The coarse grid correction steps include an restriction and an interpolation operator. The next subsection considers these so called transfer operators.

4.2.3 Transfer operators

As seen in the introduction, one of the main aspects in the success of MG methods requires special mappings referred to as *transfer operators*. Because it turns out that this process can be defined recursively, an analysis for transferring between Ω_H (coarse grid) and Ω_h (fine grid) will be sufficient for $H = 2h$. The operators needed will define mappings from the fine grid to the coarse grid and vice-versa[5]. The next subsection starts with the operator mapping from the coarse to the fine grid.

4.2.3.1 The prolongation operator

As can be seen in the coarse grid correction scheme in Section 4.2.2, a *restriction* and a *prolongation* operator are necessary. It will turn out that, in a special case, the restriction operator can be 'derived' from the prolongation operator. Therefore, the prolongation operator is discussed first. Another word for prolongation is *interpolation*, in other words: an approximation of the

[5]It should be noted that handling the boundaries 'sometimes' requires special adjustments

error on the fine grid is needed, using information of the error on the coarse grid (recall the definition of the error: $v_h^m = u_h - u_h^m$). For most multigrid methods, the simple linear interpolation will perform very well. It will be denoted I_{2h}^h, and is defined as:

$$\begin{aligned} v_{2j}^h &= v_j^{2h} \\ v_{2j+1}^h &= \frac{1}{2}(v_j^{2h} + v_{j+1}^{2h}), \qquad 0 \leq j \leq \frac{n}{2} - 1 \end{aligned} \tag{4.2.11}$$

Or written in vector notation: $I_{2h}^h \mathbf{v}^{2h} = \mathbf{v}^h$ (see Figure 4.2.7). Note that from this definition it follows that $I_{2h}^h : \mathbb{R}^{\frac{n}{2}-1} \rightarrow \mathbb{R}^{n-1}$.
Below is an example for the case $n = 8$:

$$I_{2h}^h \mathbf{v}^{2h} = \frac{1}{2} \begin{bmatrix} 1 & & \\ 2 & & \\ 1 & 1 & \\ & 2 & \\ & 1 & 1 \\ & & 2 \\ & & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}_{2h} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{bmatrix}_h = \mathbf{v}^h \tag{4.2.12}$$

It is important to mention that if the assumption is made that the error on the fine grid is smooth (in practice that is not known of course), the *interpolation* of the error on the coarse grid to the fine grid, also produces a smooth error. In this case, the interpolation operator gives a good approximation of the error. However, when a oscillatory error is assumed on the fine grid, the interpolation operator gives a very bad approximation to the error. So prolongation is very effective when the error is smooth. Fortunately, for the restriction operator, it is the other way around: an oscillatory error on the fine grid results in a smoother error on the coarse grid.

4.2.3.2 The restriction operator

This section is started with the restriction operator which is related to the prolongation operator in the previous subsection: the *full weighting* (FW) operator. This operator uses the values on the fine grid to make a weighted average to obtain the values for the coarse grid vector (see Figure 4.2.8). It is defined as $I_h^{2h} \mathbf{v}^h = \mathbf{v}^{2h}$, where:

$$v_j^{2h} = \frac{1}{4}(v_{2j-1}^h + 2v_{2j}^h + v_{2j+1}^h), 1 \leq j \leq \frac{n}{2} - 1 \tag{4.2.13}$$

Another important fact of this operator is the following relation:

$$I_{2h}^h = c(I_h^{2h})^T, c \in \mathbb{R} \tag{4.2.14}$$

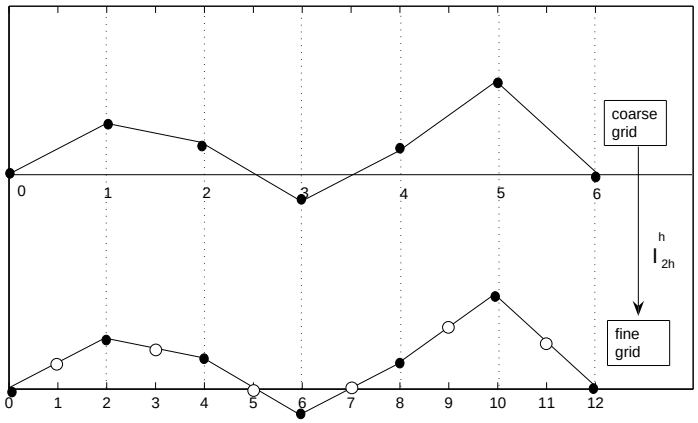

Fig. 4.2.7 Interpolation of a vector on coarse grid Ω_H to fine grid Ω_h

as illustrated in the following example (also for $n = 8$):

$$I_h^{2h}\mathbf{v}^h = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 & & & & & \\ & & 1 & 2 & 1 & & & \\ & & & & 1 & 2 & 1 & \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{bmatrix}_h = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}_{2h} = \mathbf{v}^{2h} \tag{4.2.15}$$

Another, somewhat natural, restriction operator is the *injection* operator and is defined as: $I_h^{2h}\mathbf{v}^h = \mathbf{v}^{2h}$, where $v_j^{2h} = v_{2j}^h$. So the value in the coarse grid point is simply the value of the corresponding fine grid point.

4.3 From the two-grid Cycle to multigrid

As all the necessary ingredients to obtain a simple iterative method are discussed, the following scheme of the so called *two-grid* method is given below. This is a combination of the already mentioned smoothing process and the coarse grid correction (CGC) (see Trottenberg et al.: ref. 20, page 39). It is possible to perform some smoothing processes *before* the CGC process and also *after* the CGC process. These are referred to as *presmoothing* and *postsmoothing* respectively. Presmoothing can be used to eliminate oscillatory error components on the fine grid and postsmoothing can additionally be applied when the error components after the CGC steps are not satisfactory smoothed.

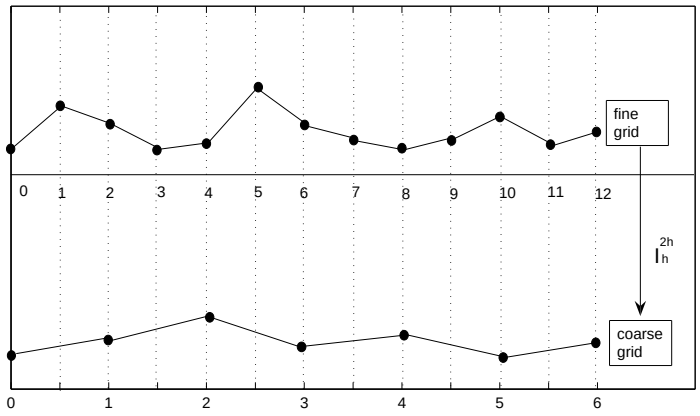

Fig. 4.2.8 Restriction by full weighting of a fine-grid vector to the coarse grid

Note that the two-grid cycle involves two grids where the necessary operations are performed. In complex problems, however the coarse grid (with for example mesh size $H = 2h$) may still contain too many unknowns to use e.g. a direct method or some other solution method to obtain the desired solution. To overcome this, a coarser grid with e.g. mesh size $\hat{H} = 2H = 4h$ can be used. This process can be repeatedly used, until a reasonably coarse grid is obtained where it is possible to get a good approximation of the solution. This repeated usage of several coarse grids leads to the Multigrid cycle. As before, first some notation is given to understand the scheme below. The coarsest grid has mesh size h_0 and the finest grid has mesh size h_ℓ and using this the following sequence of grids can be defined:

$$\Omega_{h_\ell}, \Omega_{h_{\ell-1}}, \ldots, \Omega_{h_0} \tag{4.3.1}$$

For simplicity h_k will be denoted by the index k.
The multigrid cycle described here is an $(\ell + 1)$-grid cycle to solve $L_k u_k = f_k$ (on Ω_k) for a fixed $\ell \geq 1$.

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 $u_h^{m,\text{after CGC}}$:

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

Note

In (4.3.3) 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 4.3.1 and 4.3.2.

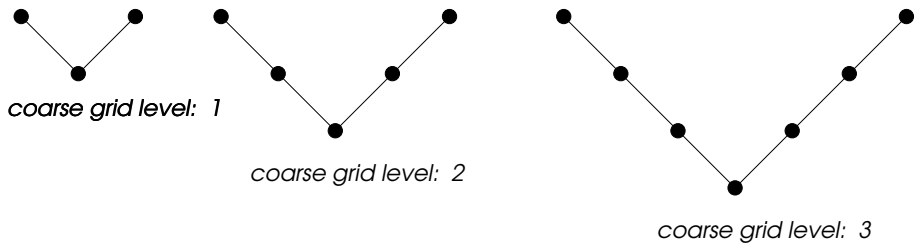

Fig. 4.3.1 V-cycles for different coarse grid levels and $\gamma = 1$

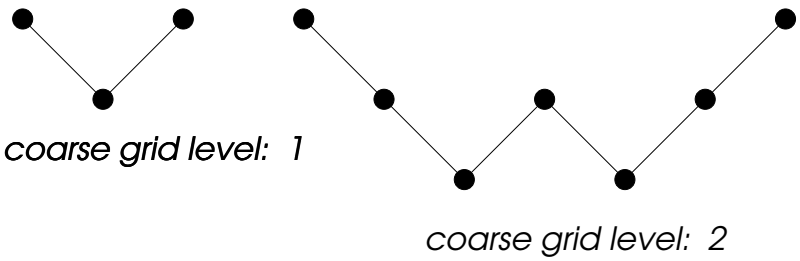

Fig. 4.3.2 W-cycles for different coarse grid levels and $\gamma = 2$

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}:

   $$L_{k-1}\hat{v}_{k-1}^m = \bar{d}_{k-1}^m, \text{ using the following} \tag{4.3.2}$$

   ► If $k = 1$, use a direct or fast iterative solver for (4.3.2)

   ► If $k > 1$, solve (4.3.2) approximately by performing $\gamma(\geq 1)$ $k-$grid cycles using the zero grid function as a first approximation:

   $$\hat{v}_{k-1}^m = \text{MGCYCLE}^{\gamma}(k-1, \gamma, 0, L_{k-1}, \hat{d}_{k-1}^m, \nu_1, \nu_2) \tag{4.3.3}$$

   - 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 $u_k^{m,\text{after CGC}}$:

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

In the following pictures some results are included, obtained from a multigrid V-cycle program in Matlab. Model Problem1 is considered discretized with the standard finite difference discretization. In all figures there are four pictures corresponding to four iterations and started in a similar initial pattern as in Figure 4.2.2.

In the figures here, the effect of pre- and post-smoothing are demonstrated by using different values of ν_1 and ν_2. In Figure 4.3.3, $\nu_1 = \nu_2 = 0$, so there is no smoothing: the MG approximation does not converge to the exact solution.

In Figures 4.3.4 and 4.3.5, there is one pre and one post smoothing respectively. The convergence is almost similar in both cases.

The smoothing method used is again of Jacobi type. Note the very fast smoothing for $\nu_1 = 2$ and $\nu_2 = 2$ in Figure 4.3.7.

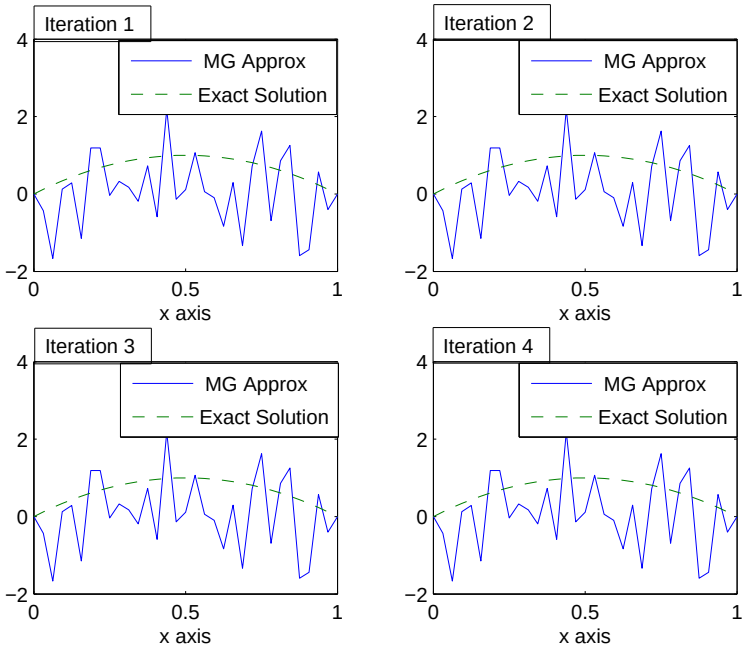

Fig. 4.3.3 $\nu_1 = 0$ *and* $\nu_2 = 0$

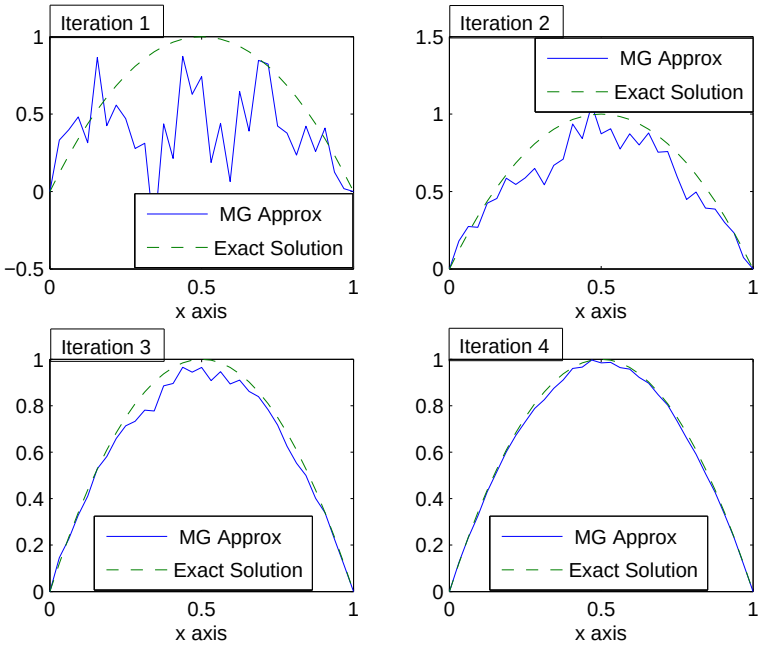

Fig. 4.3.4 $\nu_1 = 1$ *and* $\nu_2 = 0$

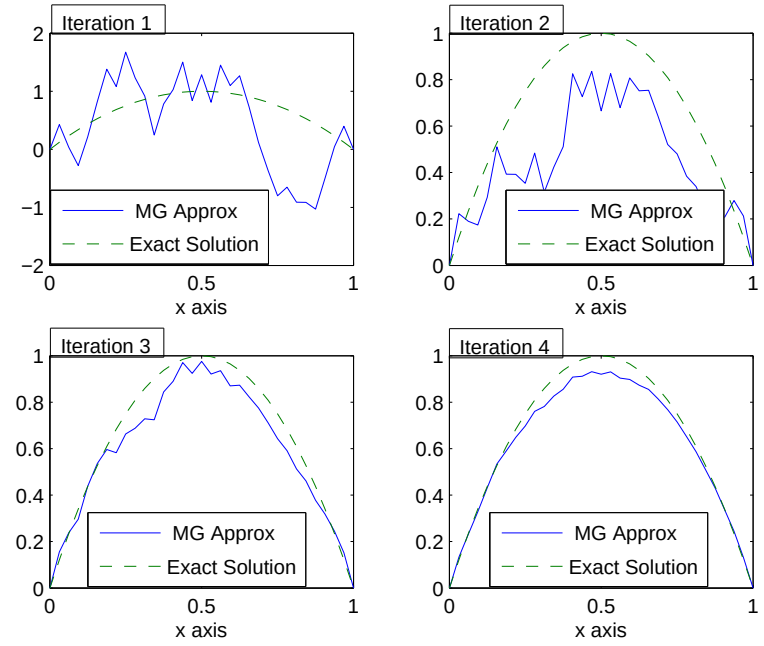

Fig. 4.3.5 $\nu_1 = 0$ *and* $\nu_2 = 1$

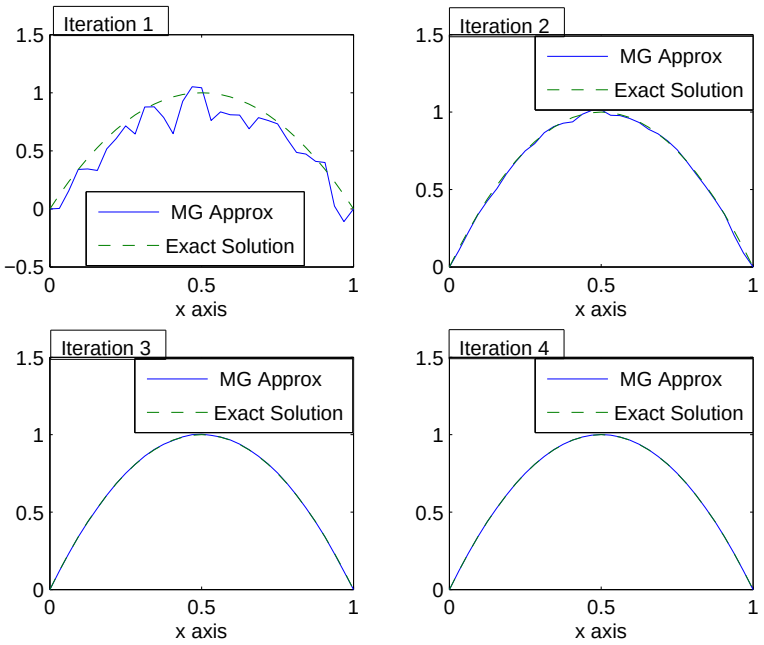

Fig. 4.3.6 $\nu_1 = 1$ *and* $\nu_2 = 1$

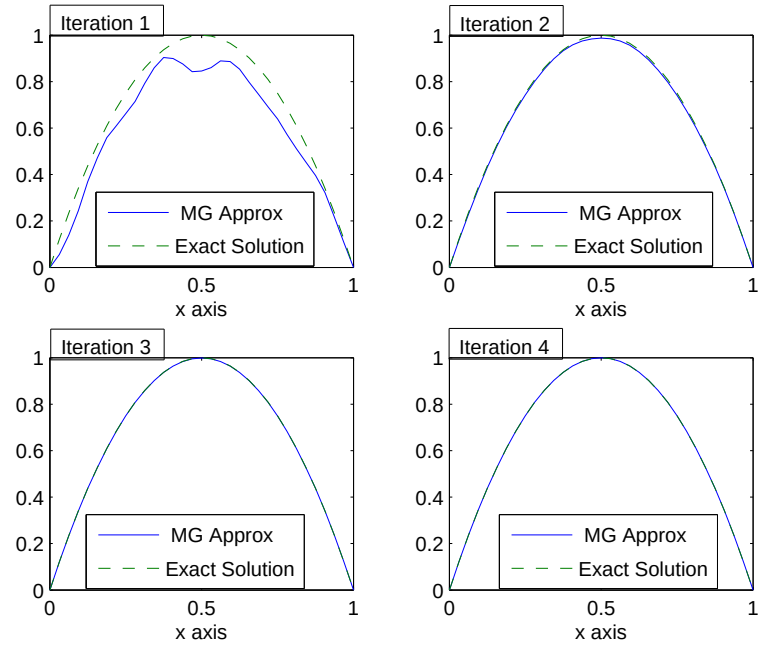

Fig. 4.3.7 $\nu_1 = 2$ *and* $\nu_2 = 2$

So far Model Problem2 has not been used. This model problem will now be used to illustrate that the convergence of multigrid strongly depend on the definiteness of the system matrix. The system matrix of the discrete Helmholtz operator will become indefinite if the wave number k_0^2 becomes larger than the smallest eigenvalue $\lambda^1_{L_h}$ of the discrete Laplace operator L_h. In all pictures below $\nu_1 = 2, \nu_2 = 2$, $\lambda^1_{L_h} = 9.8617$ and the first four MG iterations are presented. So when $k_0^2 > 9.8617$, multigrid will diverge (when $k_0 = 0$ the pictures look the same as the pictures in Figure 4.3.7).
In Figure 4.3.8 $k_0 < \lambda^1_{L_h}$, so the system matrix is still definite. Therefore there is no divergence, but the convergence is slower compared to the convergence in e.g. Figure 4.3.7. In Figure 4.3.9 $k_0 = \lambda^1_{L_h}$ resulting in no divergence but also no reasonable approximation. In Figures 4.3.10, 4.3.11 and 4.3.12, $k_0 > \lambda^1_{L_h}$. Hence the system matrices become indefinite and there is no convergence at all. Note the 'explosive' character (10^9 on the y-axis) when $k_0 = 49$.

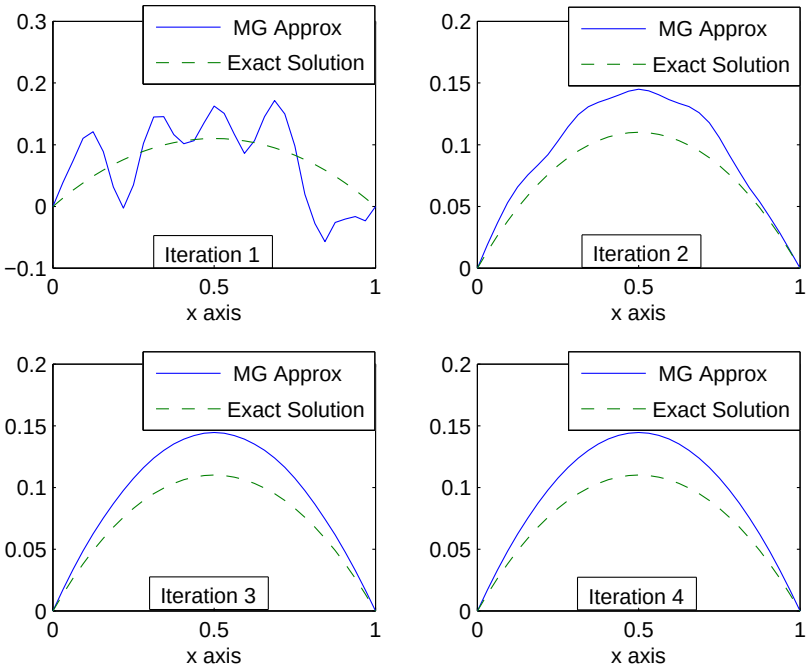

Fig. 4.3.8 wavenumber $k_0 = \sqrt{9.8617} - 2$

As the Maxwell equations in the current application have similar properties as the one dimensional Helmholtz equation, it is likely that a direct application of a multigrid solution method on the discretized linear system will result in a diverging method. Therefore multigrid will be applied on a preconditioner version of the current application.

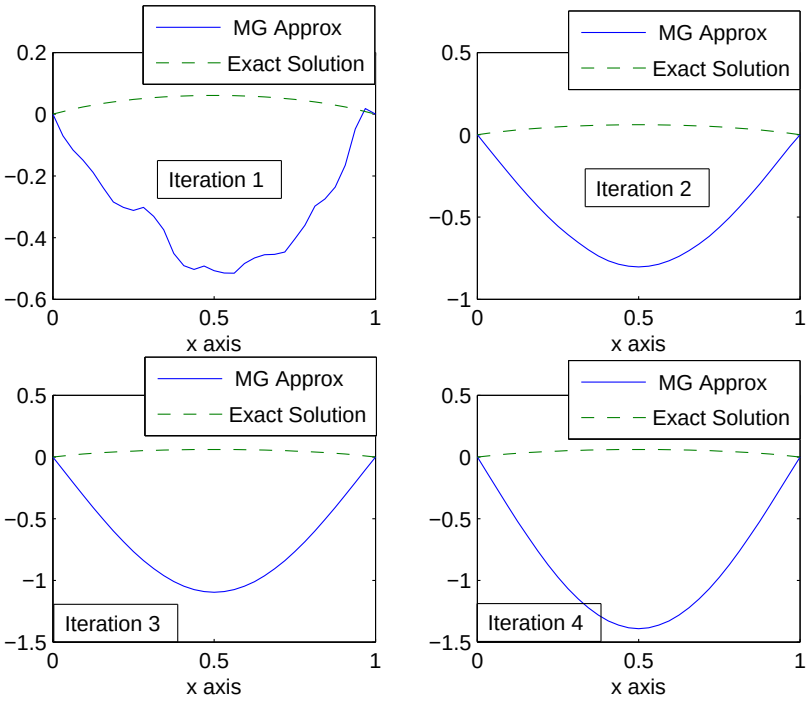

Fig. 4.3.9 wavenumber $k_0 = \sqrt{9.8617}$

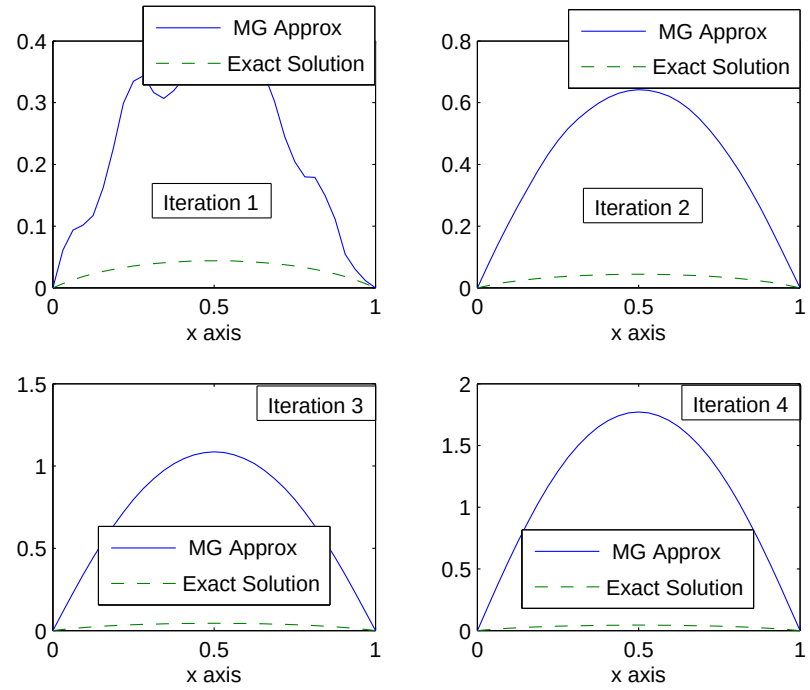

Fig. 4.3.10 wavenumber $k_0 = \sqrt{9.8617} + 1$

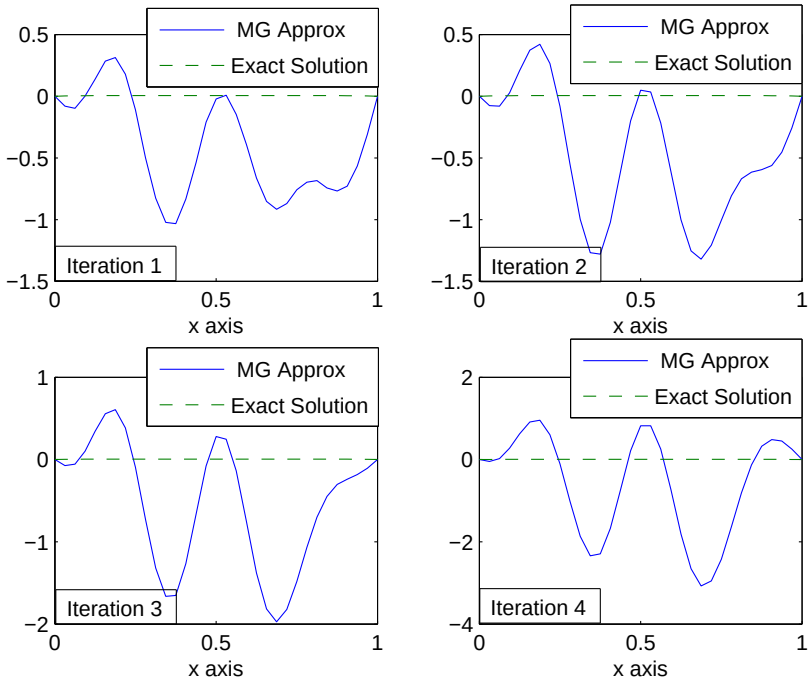

Fig. 4.3.11 wavenumber $k_0 = \sqrt{9.8617} + 10$

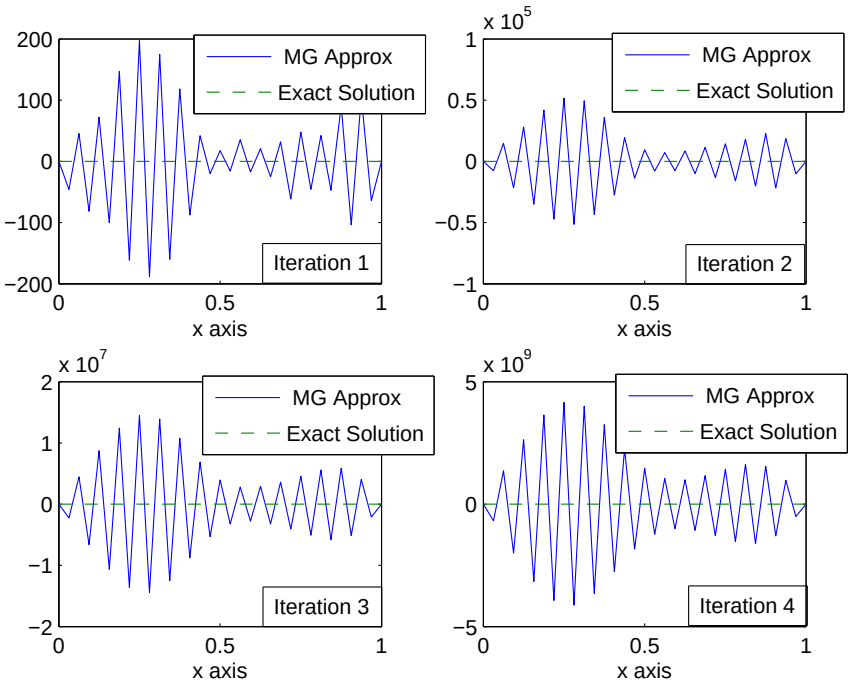

Fig. 4.3.12 wavenumber $k_0 = 49$

In the next section it will be explained why the two-grid cycle converges and because this is the basis for the (full) multigrid cycle, the analysis for the two-grid cycle will be sufficient.

4.4 Convergence analysis

As already mentioned in the previous section, the two-grid cycle forms the basis of the MG cycles. For this reason, the two grid cycle will be analyzed first. Note the following statement (Trottenberg et al., ref. 20, Section 3.2):

> ***Statement*** *If a given two-grid method converges sufficiently well, i.e.*
>
> $$||T_h^H||_{A_h} \leq \sigma,$$
>
> *with σ small enough and independent of h, then the corresponding multigrid method will have similar convergence properties, under natural assumptions*

Here T_h^H denotes the two-grid operator (see equation (4.4.1)). For a detailed description about convergence factors for the two-grid and multigrid methods, the reader is referred to Trottenberg et al. (ref. 20).

4.4.1 Analysis of the two-grid cycle – Two important subspaces

Recall the necessary processes in the two-grid cycle in Section 4.3:

- Presmoothing
- Coarse grid correction
- Postsmoothing

Each of these processes lead to an operation on the operator A_h on the finest grid with mesh size h. When there are no pre- and postsmoothing processes, only the coarse grid correction is of importance. In Section 4.2.3 the coarse grid correction steps are outlined and the resulting operator can be denoted by:

$$T_h^H = I - \underbrace{I_H^h A_H^{-1} I_h^H}_{:=B_h} A_h \tag{4.4.1}$$

Note that B_h is the preconditioning matrix associated with this iteration.

It can be shown (Saad, ref. 26, p. 425) that when $A_H := I_h^H A_h I_H^h$, then the coarse grid correction operator T_h^H is a *projector*[6] that is orthogonal with respect to the A_h-inner product. Furthermore, the range of T_h^H is A_h-orthogonal to the range of I_h^H. This information will be useful in

[6] For the definition of a projector, the reader is referred to appendix B

defining two subspaces to finally arrive at an important theorem which explains why MG methods converge.

Before the two important subspaces can be defined, some preparation is necessary. When Q_h is defined as follows:

$$Q_h = I_H^h A_H^{-1} I_h^H A_h \tag{4.4.2}$$

the following holds:

1. Q_h is an A_h-orthogonal projector onto the subspace $\Omega_h \Rightarrow I - Q_h$ is also an A_h-orthogonal projector. Using fundamental relation (FR I) combined with the properties in Appendix B, it can be shown that:

$$\Omega_h = \text{Ran}(Q_h) \oplus \text{Ker}(Q_h) := \text{Ran}(Q_h) \oplus \text{Ran}(I - Q_h) \tag{4.4.3}$$

2. $\text{Ran}(Q_h) \subset \text{Ran}(I_H^h)$
3. It can also be proved that the inclusion in point 2. above also holds the other way around: take a vector z in the range of $I_H^h \Rightarrow z = I_H^h y$ for a certain $y \in \Omega_h$. With the definition of A_H it then follows that:

$$Q_h z = I_H^h A_H^{-1} I_h^H A_h I_H^h y = I_H^h y = z$$

   This shows that $z \in \text{Ran}(Q_h)$. Hence: $\text{Ran}(Q_h) \supset \text{Ran}(I_H^h)$
4. Combining point 2. and 3. results in: $\text{Ran}(Q_h) = \text{Ran}(I_H^h)$.

Summarizing:

- Q_h is the A_h-orthogonal projector onto the space $\text{Ran}(I_H^h)$
- T_h^H is the A_h-orthogonal projector onto the orthogonal complement: the range of $(I - Q_h)$ which is also the null space of Q_h according to fundamental relation (FR II)
- Using fundamental relation (FR I) again results in:

$$\Omega_h = \text{Ran}(I_H^h) \oplus \text{Ker}((I_H^h)^T) = \text{Ran}(Q_h) \oplus \underbrace{\text{Ker}((I_H^h)^T)}_{\text{Ker}(I_h^H)} \tag{4.4.4}$$

- Using equation (4.4.3) results in:

$$\text{Ker}(Q_h) = \text{Ker}(I_h^H)$$

Finally the two fundamental subspaces can be defined as follows:

- $\mathcal{S}_h := \text{Ran}(Q_h)$: consisting of the smooth components
- $\mathcal{T}_h := \text{Ran}(T_h^H)$: consisting of the oscillatory components

with the following relations:

1. $\Omega_h = \mathcal{S}_h \oplus \mathcal{T}_h$
2. $\mathcal{S}_h = \text{Ran}(Q_h) = \text{Ker}(T_h^H) = \text{Ran}(I_H^h)$
3. $\mathcal{T}_h = \text{Ker}(Q_h) = \text{Ran}(T_h^H) = \text{Ker}(I_h^H)$

Note that relation 1. here states that the subspace Ω_h can be decomposed into two other subspaces $\mathcal{S}_h$ and $\mathcal{T}_h$. It is also true that Ω_h can be decomposed into a subspace consisting of the smooth components ($\mathcal{S}_h$) and a subspace consisting of the oscillatory components ($\mathcal{T}_h$). Let s be a smooth mode and t a oscillatory one. Then the action of the two-grid operator roughly translates into:

$$T_h^H s \approx 0 \qquad\qquad T_h^H t \approx t$$

In contrary, for Q_h the opposite is true:

$$Q_h s \approx s \qquad\qquad Q_h t \approx 0$$

To illustrate the properties above, the following example is given:

In the context of Model Problem1, consider the prolongation operator I_h^H corresponding to the one dimensional full weighting (FW) case. Let w_k be an eigenmode with components $\sin(j\theta_k)$ for $j = 1, \ldots, n$ and $\theta_k = \frac{k\pi}{n+1}$. When FW is applied onto eigenmode w_k, the following is obtained:

$$\begin{aligned}
(I_h^H w_k)_j &= \frac{1}{4}[\sin((2j-1)\theta_k) + 2\sin(2j\theta_k) + \sin((2j+1)\theta_k)] \\
&= \frac{1}{4}[2\sin(2j\theta_k)cos(\theta_k) + 2\sin(2j\theta_k)] \\
&= \frac{1}{2}(1+\cos(\theta_k))\sin(2j\theta_k) \\
&= \cos^2\left(\frac{\theta_k}{2}\right)\sin(2j\theta_k)
\end{aligned}$$

Now distinguish between the following two cases:

1. Suppose $k \approx n$ (i.e. k is large). Then $\theta_k \approx \pi \rightarrow \cos^2(\frac{\theta_k}{2}) \approx 0 \Rightarrow I_h^H w_k \approx 0$. This shows that w_k is near the null space of I_h^H and with k being large w_k is a oscillatory node $\rightarrow$ oscillatory modes are close to being in the null space of I_h^H, or equivalently, the range of T_h^H. In terms of fine and coarse grids: the restriction operator will transform mode w_k into a constant ($\cos(\frac{\theta_k}{2})$) times the same mode on the coarser grid.

2. When k is small, then w_k is a smooth mode and the constant $\cos(\frac{\theta_k}{2}) \approx 1$. Therefore the interpolation operator produces the equivalent smooth mode in subspace Ω_H without damping it. ◁◁

In the next subsection first some notation will be introduced where after two important properties will be stated. These will be used to derive the important convergence theorem for the two-level iteration.

4.4.2 Convergence of multigrid

In this subsection the convergence for the Galerkin case is analyzed in which the A_h-norm[7] is used. The 2-norms weighted by $D^{\frac{1}{2}}$ or $D^{-\frac{1}{2}}$ are convenient, where $D = \text{diag}(A)$.

The following notation will be used for a arbitrary vector x and (error)vector e:

$||x||_D = (Dx, x)^{\frac{1}{2}} := ||D^{\frac{1}{2}}x||_2$

$||e||_{A_h D^{-1} A_h} = (D^{-1}A_h e, A_h e)^{\frac{1}{2}} := ||A_h e||_{D^{-1}}$

Using the notation introduced above, the next two properties can be shown:

⋆ The *smoothing property*:

$$||S_h e^h||^2_{A_h} \leq ||e^h||^2_{A_h} - \alpha||Ae^h||^2_{D^{-1}} \qquad \forall e^h \in \omega_h \tag{SP}$$

In this equation α is a positive constant. Furthermore, this property holds independent of the choice of h and characterizes the smoother.

⋆ The *approximation property*:

$$\min_{u_H \in \Omega_H} ||e^h - I_H^h e^H||^2_D \leq \beta||e^h||^2_{A_h} \tag{AP}$$

In this equation β is independent of h and this property characterizes the discretization.

Theorem (Convergence of the two-level iteration)

In this theorem the following assumptions are made:

- matrix A is symmetric and positive definite (SPD)
- the restriction and prolongation operator are linked as follows:

$$I_H^h = 2^d (I_h^H)^T$$

where d is the dimension of the space and I_H^h is of full rank

[7] For the definition of a norm, the reader is referred to Appendix B

- inequalities (SP) and (AP) are satisfied for a certain smoother and $\alpha, \beta > 0$

When all these assumptions are fulfilled, the following three statements hold:

1. $\alpha < \beta$
2. the two-level cycle converges
3. the norm of the smoother and two-cycle operator $S_h T_h^H$ is bounded as follows:

$$||S_h T_h^H||_{A_h} \leq \sqrt{1 - \frac{\alpha}{\beta}} \tag{4.4.5}$$

Proof. Recall from the previous subsection that $\text{Ran}(T_h^H) = \mathcal{T}_h$ is A_h-orthogonal to $\text{Ran}(I_H^h) = \mathcal{S}_h$. Therefore:

$$(e^h, I_H^h e^H)_{A_h} = 0 \quad \forall e^h \in \text{Ran}(T_h^H) \Rightarrow ||e^h||_{A_h}^2 = (A_h e^h, e^h - I_H^h e^H) \quad \forall e^h \in \text{Ran}(T_h^H)$$

Using the Cauchy-Schwarz inequality for any $e^h \in \text{Ran}(T_h^H)$ gives:

$$\begin{aligned} ||e^h||_{A_h}^2 &= (D^{-\frac{1}{2}} A_h e^h, D^{\frac{1}{2}}(e^h - I_H^h e^H)) \\ &\leq ||D^{-\frac{1}{2}} A_h e^h||_2 ||D^{\frac{1}{2}}(e^h - I_H^h e^H)||_2 \\ &= ||A_h e^h||_{D^{-1}} ||e^h - I_H^h e^H||_D \end{aligned} \tag{4.4.6}$$

Using (AP) on (4.4.6) implies that:

$$||e^h||_{A_h} \leq \sqrt{\beta} ||A_h e^h||_{D^{-1}} \quad \forall e^h \in \text{Ran}(T_H^h) \equiv ||T_h^H e^h||_{A_h}^2 \leq \beta ||A_h T_H^h e^h||_{D^{-1}}^2 \quad \forall e^h \in \Omega_h$$

Finally using (SP) results in:

$$\begin{aligned} 0 \leq ||S_h T_h^H e^h||_{A_h}^2 &\leq ||T_h^H e^h||_{A_h}^2 - \alpha ||A_h T_h^H e^h||_{D^{-1}}^2 \\ &\leq ||T_h^H e^h||_{A_h}^2 - \frac{\alpha}{\beta} ||T_h^H e^h)||_{A_h}^2 \\ &= \Big(1 - \frac{\alpha}{\beta}\Big) ||T_h^H e^h||_{A_h}^2 \\ &\underset{\leq}{(\star)} \Big(1 - \frac{\alpha}{\beta}\Big) ||e^h||_{A_h}^2 \end{aligned}$$

$(\star)$ is allowed because of the fact that T_h^H is an A_h-orthogonal projector. □

Before completing this section, one must note the following important remark: The convergence theorem in this subsection is based on the assumption that the matrix A is SPD. In the current application the system matrix is 'nearly' symmetric and indefinite. Hence direct application of multigrid on the discretized version of the Maxwell equations will probably not result in a converging method. Therefore multigrid will be used as solution method to the linear system stemming from the discretization of the shifted Laplace preconditioner. In the next section the shortcomings of geometric multigrid will be mentioned when applied to the current problem.

4.5 Algebraic multigrid

So far geometric multigrid has been explained and it can be concluded that these methods are very effective solvers for large (linear) systems arising from discretization of several types of PDEs and when the system matrix is SPD. Unfortunately there are some drawbacks when classical MG methods are used to solve systems arising from e.g. physical problems with anisotropic[8], strongly varying or discontinuous coefficients. Another drawback of geometric multigrid is that they require structured grids for their successful application. Also the definition of the interpolation (and thus restriction) operator are dependent on the operator being considered in the original equation: *operator-dependent interpolation*. In the current application there is no structured grid and to overcome the other shortcomings and at the same time take advantage of all the good properties of the classical MG methods, an extension has to be made to *Algebraic Multigrid*. It will turn out that the Galerkin approach that will be used in algebraic multigrid will deal with the operator-dependent interpolation. It is also favorable that the Galerkin operator can be constructed pure algebraically because no grid specification is needed.

In this section Algebraic multigrid (AMG) will be discussed. Actually, the AMG approach is opposite to the geometric approach. Geometric multigrid first fixes the coarse grids and then defines suitable operators and smoothers, while the AMG approach can roughly be written as:

- fix smoother (e.g. Gauss-Seidel or Jacobi)
- choose coarse grids and prolongation operator I_H^h such that the error not reduced by relaxation is in Ran(I_H^h)
- define other MG components so that coarse-grid correction eliminates error in Ran(I_H^h) (e.g. use Galerkin principle)

For classical AMG methods, the system matrix is assumed to be an M-matrix. For these type of matrices, the convergence of Jacobi and Gauss-Seidel is well understood.
In Subsection 4.5.1 the basic steps of algebraic multigrid will be outlined and after this short introduction, AMG for complex valued systems will be introduced in Subsection 4.5.2. Finally in Subsection 4.5.3, AMG for the finite element discretization will be discussed.

4.5.1 Basic tools of algebraic multigrid

Another property of geometric multigrid methods is that they require a given problem to be defined on a grid known a priori. So the coarsening process itself is fixed and kept as simple as possible. Algebraic multigrid does not require these but operates directly on sparse linear systems:

[8] if certain material properties are not equal in all directions

$$A_h u^h = f^h \quad \text{or} \sum_{j=1}^{N} a_{ij}^h u_j^h = f_i^h \quad (i = 1, 2, \ldots, N) \qquad N \text{ denotes the total number of unknowns} \tag{4.5.1}$$

In the discussion about geometric multigrid the following terms were used: *grids, subgrids* and *grid points.* When these terms are replaced by *sets of variables, subsets of variables* and *single variables* respectively, AMG can be formally described in the same way as geometric multigrid. Before the basics are discussed, the following remark is stated where after the two necessary ingredients of AMG are stated:

Remark:

It is important to realize that although the discussion in this thesis is written in terms of a (given) matrix A, the power of AMG also covers a *class of matrices* $\mathcal{A}$; e.g.: the class consisting of all M-matrices.

AMG needs two basic ingredients for the setup:

1. a way of defining the coarse subspace X_H from a fine subspace X_h. Note that h no longer denotes a mesh size but an index to a certain level and H is used to index a coarser level. Also note that Ω_h is now replaced with a subspace X_h of $\mathbb{R}^n$ at a certain level h.
2. a way to define the interpolation operator I_h^H from X_h to X_H. In AMG, the coarse-level problem$^{(*_1)}$ is defined using the Galerkin approach which has already been seen in Subsection 4.2.3:

   $$A_H = I_h^H A_h I_H^h, \qquad f^H = I_h^H f^h \tag{4.5.2}$$

   The restriction (I_h^H) and prolongation (I_H^h) operator are both defined pure algebraically and are related by:

   $$I_h^H = (I_H^h)^T \tag{4.5.3}$$

   A minimal assumption that must be made on the prolongation operator is that it must be of full rank. The direct consequence of this assumption is that the restriction operator is also of full rank.

$(*_1)$ Note that the coarse-level problem is denoted by:

$$A_H u^H = f^H \quad \text{or} \sum_{j=1}^{N} a_{ij}^H u_j^H = f_i^H \quad (i \in X_H) \tag{4.5.4}$$

Before it is possible to define a scheme for coarsening, a distinction must be made analogous to the distinction of smooth and oscillatory modes in the geometric case. This extension of smooth and oscillatory modes is the subject of the next subsection.

4.5.1.1 Smoothness in AMG

In AMG an error is also decomposed into smooth and oscillatory components. The difference with geometric MG is that in the AMG case this decomposition will now be defined with respect to the ability or the inability of the chosen smoother to reduce these components. This can be formulated as follows: *an error s is smooth when its convergence with respect to the chosen smoother S_h is slow.* Mathematically written this looks like:

$$||S_h s||_A \approx ||s||_A$$

Here the energy norm is used[9].

As AMG employs simple smoothing processes, a typical smoother uses e.g.:

- Gauss-Seidel relaxation: $S_h = I_h - Q_h^{-1} A_h$, with Q_h being the lower triangular part of A_h, including the diagonal
- ω-Jacobi relaxation: $S_h = I_h - \omega D_h^{-1} A_h$, with $D_h = \text{diag}(A_h)$

If s satisfies the smoothing property (SP), it holds that:

$$||As||_{D^{-1}} \ll ||s||_{A_h}$$

Using the definition of $||.||_{A_h}$ and the Cauchy-Schwarz inequality gives:

$$\begin{aligned} ||s||_{A_h}^2 &= (D^{-\frac{1}{2}} A_h s, D^{\frac{1}{2}} s) \\ &\leq ||D^{-\frac{1}{2}} A_h s||_2 ||D^{\frac{1}{2}} s||_2 \\ &= ||A_h s||_{D^{-1}} ||s||_D \end{aligned}$$

Since $||As||_{D^{-1}} \ll ||s||_{A_h}$, this means that $||s||_{A_h} \ll ||s||_D$

$$\Rightarrow (As, s) \ll (Ds, s) \tag{4.5.5}$$

When $v := D^{\frac{1}{2}} s$ then

$$(D^{-\frac{1}{2}} A D^{-\frac{1}{2}} v, v) \ll (v, v)$$

These requirements demand the *Rayleigh quotient*[10] of $D^{\frac{1}{2}} s$ to be small, which in turn implies that the vector v is a linear combination of the eigenvectors of A with smallest eigenvalues. In particular, $(As, s) \approx 0$ also implies that $As \approx 0$:

$$a_{ii} s_i \approx -\sum_{j \neq i} a_{ij} s_j \tag{4.5.6}$$

[9] For the definition of the energy norm, the reader is referred to Appendix B

[10] For the definition of Rayleigh quotient, the reader is referred to Appendix B

Equation (4.5.6) above heuristically characterizes a smooth error. This is true because interpolation should average the error out → eliminate highly oscillating components in X_h and produce a function that becomes smooth in X_H.
When (As, s) above is expanded, the notation in terms of matrix coefficients looks like:

$$\begin{aligned}(As, s) &= \sum_{i,j} a_{ij} s_i s_j \\ &= \frac{1}{2} \sum_{i,j} -a_{ij}\big((s_j - s_i)^2 - s_i^2 - s_j^2\big) \\ &= \frac{1}{2} \sum_{i,j} -a_{ij}(s_j - s_i)^2 + \sum_i \left(\sum_j a_{ij} \right) s_i^2\end{aligned}$$

If equation (4.5.5) is rewritten in terms of ϵ with $0 < \epsilon \ll 1$ and the assumption is made that the row sums of the matrix are zero and the off-diagonal elements are negative, the following fundamental relation (4.5.7) can be derived:

$$\sum_{j \neq i} \frac{|a_{ij}|}{a_{ii}} \Big(\frac{s_i - s_j}{s_i} \Big)^2 \ll 1 \tag{4.5.7}$$

For (4.5.7) to hold, $\frac{|s_i - s_j|}{s_i}$ must be small when $|\frac{a_{ji}}{a_{ii}}|$ is large. In other words: the components of s vary slowly in the direction of the strong connections. This observation is at the basis of many AMG techniques and will be used in the next subsection when interpolation operators will be defined.

4.5.1.2 Interpolation in AMG

In the last paragraph of the previous subsection it was remarked that the components of s vary slowly in the direction of the *strong* connections. So before the interpolation operator can be defined, it is necessary to distinguish between different *couplings* between nodes. To achieve this, (4.5.7) will be used as follows:

- let i be a coarse node; its adjacent node with index j such that $a_{ij} \neq 0$
- when $|\frac{a_{ij}}{a_{ii}}|$ is smaller then a certain threshold σ, i and j are said to be *weakly coupled*
- when $|\frac{a_{ij}}{a_{ii}}|$ is greater then a certain threshold σ, i and j are said to be *strongly coupled*

To understand "smaller" in the context above, a *splitting* in coarse and fine nodes is needed, a so called ***CF-splitting*** (see Figure 4.5.1). The smaller filled black circles represent the fine nodes and the thin dashed lines represent weak connections. The dash-dot lines represent the strong connections.
In the context here, the definition of strong connections is given as:

$$S_i = \Big\{ j : -a_{ij} \geq \theta \max_{k \neq i} \{a_{ik}\} \Big\}, \qquad \text{for some fixed } \theta \in (0, 1) \tag{4.5.8}$$

(a typical choice for parameter θ is $\theta = 0.25$)

Using the CF-splitting, the following notation[11] of X_h is introduced for three types of nodes among the nearest neighbors of a fine node i:

- C_i: denotes the set of coarse nodes ($\Rightarrow F_i$ is the set of fine nodes)
- F_i^s: set of fine nodes strongly connected with i ($\Rightarrow F_i^s = F_i \cap S_i$)
- F_i^w: set of fine nodes weakly connected with i ($\Rightarrow F_i^w = F_i \cap F_i^s$)

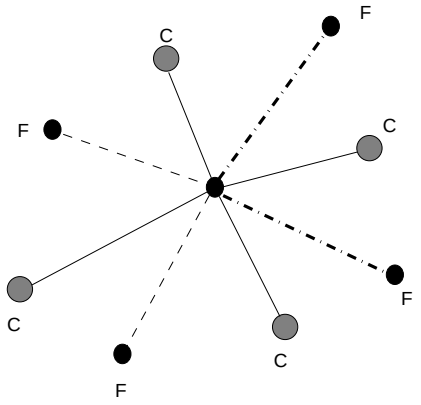

Fig. 4.5.1 Example of nodes adjacent to a fine node i (center). Fine mesh nodes are labeled with F, coarse nodes with C.

As already mentioned, equation (4.5.6) heuristically characterizes the smooth error, and will now be used to produce an interpolation function. Using the CF-splitting mentioned above, equation (4.5.6) can be rewritten as:

$$a_{ii}s_i \approx -\sum_{j\in C_i} a_{ij}s_j - \sum_{j\in F_i^s} a_{ij}s_j - \sum_{j\in F_i^w} a_{ij}s_j \tag{4.5.9}$$

Starting from equation (4.5.9), first the weak connections will be handled by adding their result into the diagonal term a_{ii} and secondly the 'combine and write technique' should be used to express the right hand side of the formula in terms of coarse grid points only. The end result is a formula that only depends on the coarse grid points, which will be included in this thesis. For the complete derivation of this formula, the reader is referred to Saad (ref. 26, p. 439). The resulting

[11] Note that $X_h = C_i \cup F_i^s \cup F_i^w$

formula looks like:

$$s_i = \sum_{j \in C_i} w_{ij} s_j, \qquad \text{with} \qquad w_{ij} := -\frac{a_{ij} + \sum_{k \in F_i^s} \frac{a_{ik} a_{kj}}{\delta_k}}{a_{ii} + \sum_{k \in F_i^w} a_{ik}} \text{ and } \delta_k := \sum_{l \in C_i} a_{kl} \tag{4.5.10}$$

When the weights w_{ij} are determined, the resulting interpolation formula generalizes the formulas seen for geometric multigrid:

$$(I_H^h x)_i = \begin{cases} x_i, & \text{if } i \in X_H \\ \sum_{j \in C_i} w_{ij} x_j, & \text{otherwise} \end{cases} \tag{4.5.11}$$

This subsection will be ended with some remarks about the interpolation and the CF-splitting.

Remarks

1. In order to achieve fast convergence, the algebraically smooth error needs to approximated well by the interpolation.
2. As the size of the coarse-level operator strongly depends on the total number of C-variables, it is desirable to limit the number of C-variables, while still guaranteeing that all F-variables are sufficiently strongly connected to the C-variables. The goal however, is not to minimize the total number of C-points. It is important to create CF-splittings which are as uniform as possible with F-variables being "surrounded" by C-variables to interpolate from.
3. Recall equation (4.5.1) rewritten as:

   $$A_h u^h = f^h \qquad \text{or} \qquad \sum_{j \in X_h} a_{ij}^h u_j^h = f_i^h \quad (i \in X_h) \tag{4.5.12}$$

   where X_h denotes the indexing set $\{1, 2, 3, \ldots, n\}$.
   For theoretical investigations, it is convenient to write a given CF-splitting of (4.5.12) in block form as follows:

   $$A_h u = \begin{bmatrix} A_{FF} & A_{FC} \\ A_{CF} & A_{CC} \end{bmatrix} \begin{pmatrix} u_F \\ u_C \end{pmatrix} = \begin{pmatrix} f_F \\ f_C \end{pmatrix} = f \tag{4.5.13}$$

   For more about the usefulness of this block notation and all the theorems that can be derived, the reader is referred to Trottenberg et al. (ref. 20, Appendix A).
   This block notation will also be used in the next subsection where the coarse spaces in AMG are defined, using multilevel ILU.

4.5.1.3 Coarse spaces in AMG

To define a coarse space X_H from a fine subspace X_h, the mechanism that will be used is called *coarsening*. In this thesis one way of getting to a two-level cycle will be discussed. For other derivations of two-level methods the reader is encouraged to read e.g. Trottenberg et al. (ref. 20,

p.427) and fore more details about the coarsening strategies in AMG, the reader is referred to Trottenberg et al. (ref. 20, p.472).
The simplest way to achieve a coarsening scheme uses e.g. the idea of independent set orderings (ISOs)[12]. ISOs transform the original system (4.5.12) into the following form:

$$\begin{bmatrix} B & F \\ E & C \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}, \text{ where block } B \text{ is a diagonal matrix.} \tag{4.5.14}$$

A block LU factorization of the previous system looks like:

$$\begin{bmatrix} B & F \\ E & C \end{bmatrix} = \begin{bmatrix} I & 0 \\ EB^{-1} & I \end{bmatrix} \begin{bmatrix} B & F \\ 0 & S \end{bmatrix}, \text{ where } S \text{ is the Schur complement } S := C - EB^{-1}F \tag{4.5.15}$$

To understand the link with AMG-type methods, the demand on B being diagonal, may be dropped. It is then possible to derive a generalization of the factorization shown above. As ISOs work with independent set orderings, this generalization will use block or group independent sets[13].

When the unknowns in a independent group are permuted such that those associated with the group independent set are listed first, followed by the other unknowns, the original coefficient system will take the form (4.5.14), where now matrix B is a block diagonal matrix. When an exact or complete LU factorization of B is performed: $B = LU + R$, the original system can be factorized as follows:

$$A = \begin{bmatrix} B & F \\ E & C \end{bmatrix} \approx \begin{bmatrix} L & 0 \\ EU^{-1} & I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} U & L^{-1}F \\ 0 & I \end{bmatrix} := \mathcal{LDU} \tag{4.5.16}$$

The factorization in (4.5.16) is analogue to the two-grid cycle seen in geometric multigrid. Using this factorization, the following setup can be stated:

- solve with the $\mathcal{L}$-matrix: take a vector with components u, y in the fine and coarse spaces respectively, to produce the vector $y_H = y - EU^{-1}u$ in the coarse space
- the Schur complement S can now be solved in some unspecified manner
- solve back with the $\mathcal{U}$-matrix: take a vector from the coarse space and produce the u variable from the fine space as $u := u - L^{-1}Fy$

[12] For more about ISOs, the reader is referred to Appendix C
[13] See Appendix C

This setup can be used to derive the following ***two-level block algorithm***:

1. $f := L^{-1}f$
2. $g := g - EU^{-1}f$
3. *Solve* $Sy = g$
4. $f := f - L^{-1}Fy$
5. $x = U^{-1}f$

When this two-level algorithm is extended using a recursive definition, a possible variation of an algebraic multigrid method is achieved. See for an example of an algebraic recursive multilevel solver (ARMS) Saad (ref. 26, Chapter 13, page 444).
One disadvantage in AMG methods is that the sparse structure of the used matrices will be lost, as the number of levels increases. To maintain sparsity, small elements can be dropped in the block factorization (as is done in ARMS).
In the next subsection AMG will be briefly discussed for complex valued systems, as the system matrix in the current application is complex valued.

4.5.2 Algebraic multigrid for complex valued systems

In this subsection it is assumed that the system matrix is complex-valued symmetric or Hermitian and the results are analogous to the results of Maclachlan and Oosterlee (ref. 15).

When the AMG algorithm is to be generalized to complex-valued matrices, it must be ensured that the relaxation performs as expected. In this thesis the (weighted) Jacobi or Gauss-Seidel were mentioned as smoothers, and when the extension is made to complex-valued systems, it must be certain that the smoothing properties are provided for a reasonable class of problems.
The complex generalization of M-matrices are the H-matrices, which is defined as follows:

Definition (Comparison matrix – H-matrix)
Let $A \in \mathbb{C}^{n\times n}$ be such that its *comparison matrix*,

$$(\mathcal{M}(A))_{ij} = \begin{cases} |a_{ii}|, & \text{if } i = j \\ -|a_{ij}|, & \text{if } i \neq j \end{cases}$$

is an M-matrix. Then A is called an H*-matrix*.

For this class of H-matrices, the following theorem is reproduced from Varga (ref. 22):

Theorem

First define the following:

- *for any nonsingular H-matrix, $A \in \mathbb{C}^{n\times n}$, let D be the diagonal of A and $-L$ be the strictly lower triangular part of A. Then $U = A - (D - L)$ is strictly upper triangular*
- *define $J_\omega(A) = I - \omega D^{-1}A$ to be the error propagation operator for the weighted Jacobi iteration with weight ω*
- *define $G_\omega(A) = I - \omega(D - \omega L)^{-1}A$ to be the error propagation operator for the weighted Gauss-Seidel (SOR) iteration with weight ω*
- *let $\rho(A)$ denote the spectral radius of matrix A*

Then the following holds:

- $\rho(J_1(A)) \leq \rho(J_1(\mathcal{M}(A))) < 1$
- *for any* $\omega \in \left(0, \dfrac{2}{1+\rho(J_1(A))}\right) : \rho(J_\omega(A)) \leq \omega\rho(J_1(A)) + |1-\omega| < 1$
- *for any* $\omega \in \left(0, \dfrac{2}{1+\rho(J_1(\mathcal{M}(A)))}\right) : \rho(G_\omega(A)) \leq \omega\rho(J_1(A)) + |1-\omega| < 1$

The first part of this theorem covers the convergence of the different types of Jacobi and Gauss-Seidel type relaxations. Maclachlan et al. (ref. 15) also performed *local Fourier analysis* to say more about the spectra of the Jacobi and Gauss-Seidel iteration matrices. This is done because of the main interest of the performance of these schemes as smoothers. The results are not copied here.

In the next subsections the components of AMG for complex systems will be discussed.

4.5.2.1 Interpolation

In this subsection a simple extension of the classical AMG (for M-matrices) is used. To get an idea of how strong connections can be defined in the complex case for H-matrices, a simple extension of the classical strong-connection measure in equation (4.5.8) can be used:

$$S_i = \left\{ j : |a_{ij}| \geq \theta \max_{k\neq i} |a_{ik}| \right\}, \qquad \text{for some fixed } \theta \in (0,1) \tag{4.5.17}$$

Using this measure in the complex case, it must also be the case that, for H-matrices, algebraically smooth errors vary slowly between strongly connected points. Furthermore, using this definition, AMG coarse grids may be selected using a maximal independent set algorithm as already discussed in this thesis (ISOs). It is worth mentioning that the choice of strong connections and AMG coarsening in general, is still an area of active research. To see another interesting relationship between multigrid approaches for non symmetric real matrices and their equivalent real form of a complex matrix, consider the following setting:

- let $A^{(R)}, A^{(I)} \in \mathbb{R}^{n\times n}$

- write $A \in \mathbb{C}^{n\times n}$ as $A = A^{(R)} + \imath A^{(I)} \qquad (\imath^2 = -1)$
- let $\mathbf{u} = \mathbf{u}^{(R)} + \imath\mathbf{u}^{(I)}, \qquad \mathbf{u}^{(R)}, \mathbf{u}^{(I)} \in \mathbb{R}^n$
- let $\mathbf{b} = \mathbf{b}^{(R)} + \imath\mathbf{b}^{(I)}, \qquad \mathbf{f}^{(R)}, \mathbf{f}^{(I)} \in \mathbb{R}^n$
- then the complex system $A\mathbf{u} = \mathbf{b}$ can be expressed in terms of its real parts as:

$$\begin{bmatrix} A^{(R)} & -A^{(I)} \\ A^{(I)} & A^{(R)} \end{bmatrix} \begin{pmatrix} \mathbf{u}^{(R)} \\ \mathbf{u}^{(I)} \end{pmatrix} = \begin{pmatrix} \mathbf{b}^{(R)} \\ \mathbf{b}^{(I)} \end{pmatrix} \tag{4.5.18}$$

In this thesis, two ways of constructing an interpolation operator are discussed. The first one is the following: it can be shown, using local Fourier analysis, that for (non-symmetric) matrices, the interpolation procedure should be constructed based on the symmetric part of the operator being considered (see Wienand and Oosterlee, ref. 23). For an Hermitian operator, the equivalent real form is symmetric and for a complex symmetric operator the symmetric part of the equivalent real form is a block diagonal matrix. So building the interpolation procedure in this way, means that one determines the information only on the real part of matrix A. It appears that this way of constructing the interpolation operator is too restrictive to apply in all interesting cases as a black box solver. To deal with this, a natural complex extension of the coefficients w_{ij} in (4.5.11) has to be made to define the interpolation operator.
For more details about this relationship between multigrid approaches for non symmetric real matrices and their equivalent real form of a complex matrix, the reader is referred to Maclachlan et al. (ref. 15, p. 1553).

In the next subsection the restriction operator will be briefly discussed.

4.5.2.2 Restriction

In the work of Maclachlan et al. (ref. 15) three ways of constructing a restriction operator are discussed. These three ways are summarized below:

1. The first way of getting to a restriction operator is to choose a 'simple' injection-type operator. However, the assumption must be made that the residuals at F-points are so small that actually, they can be neglected in the coarse grid problem. Therefore this choice is not commonly used in AMG. Furthermore, in the Hermitian-definite or complex-symmetric cases, using injection as restriction operator leads to poor convergence.
2. The second approach is suggested by Dendy (ref. 4) and considers complex non-symmetric operators. He suggests that, theoretically, the restriction operator should be determined as the adjoint of the interpolation operator (he based his suggestion on experiments with convection-diffusion problems). Note that this approach is the generalization of the setup in the case of classical AMG (in the classical AMG case the interpolation and restriction

were related by the transpose). To use this approach, it is necessary to define an appropriate norm. However in general, the matrix A itself cannot be used to define this norm. Therefore the normal form A^*A must be used, where A^* denotes the Hermitian of A. The disadvantage here can be the costs of forming A^*A in order to perform a restriction step. However, if the basic AMG-interpolation scheme is adapted to these complications, this approach can be very effective (it is not always necessary to store both the matrices A and A^*).

3. The third approach uses an appropriate subspace decomposition of $\mathbb{R}^n$and requires the system matrix A to be Hermitian and definite (as this is not the case in the current application, this approach will be briefly treated). The subspace is decomposed into the range of the AMG-interpolation operator and its A-orthogonal complement. When a two-level multigrid cycle is performed, the coarse grid correction phase, eliminates the errors which lie in the range of the AMG-interpolation operator (the algebraically smooth errors), while errors that are A-orthogonal are reduced on the fine grid by a relaxation method. For more details, the reader is referred to Maclachlan et al. (ref. 15).

As already seen, the power of AMG is strongly connected with the fact that AMG does not rely on geometric information about the problem to be solved. However, as in the current problem the finite element discretization method is used, there *is* some useful information available in the element stiffness matrices that can be used. In the next subsection, algebraic multigrid will be briefly considered for the finite element discretization method.

4.5.3 Algebraic multigrid for finite element discretization

In this subsection some remarks will be made about the work of Brezina et al. (ref. 13). In their work, algebraic multigrid is considered based on element interpolation. They refer to their approach as *AMGe*. This AMG-type method is an algebraic multigrid method for solving the discrete equations that arise in Ritz-type finite element methods for partial differential equations. As this is the case in the current application, some remarks about the work of Brezina et al. are included in this thesis.

AMGe assumes access to the element stiffness matrices in the finite element method and uses theory and remarks derived in the classical multigrid theory. Some of them are summarized below:

- The error in the direction of an eigenvector associated with a relatively large eigenvalue in the spectrum of the system matrix, is rapidly reduced by relaxation.
- The error in the direction of an eigenvector associated with a relatively small eigenvalue in the spectrum of the system matrix, is reduced by a factor that may approach 1 as the eigenvalue approaches 0.

- Smooth error varies slowest in the direction of strong connection.
- Interpolation must be able to approximate an eigenvector with an error bound proportional to the size of the associated eigenvalue.

The properties listed above provide the basis for constructing effective interpolation and coarsening operators for AMG(e). In the work of Brezina et al. (ref. 13) the interpolation process is analyzed.

This subsection is ended with some concluding remarks about the paper of Brezina et al. (ref. 13). They presented two local measures based on finite element theory and classical multigrid approximations properties. These two quantities measure how well the local coarsening process determines algebraically smooth error while they provide a basis for constructing better coarsening. Using some numerical experiments, they also confirmed that the resulting interpolation operators lead to improved AMG convergence rates for their test problems: a Poisson equation discretized on stretched quadrilaterals and a plane-stress cantilever beam.
In the next subsection, some concluding remarks are made about algebraic multigrid combined with the shifted Laplace preconditioner.

4.6 An AMG solver as part of the shifted Laplace preconditioner

It is not likely that in the time available, a full algebraic multigrid solver can be realized for incorporation in the existing algorithm, and on the other hand there are many AMG black box solvers available. The task is to formulate certain requirements for an AMG black box solver in order to incorporate it in the existing algorithm with minimal effort. In this section the following requirements on the black box solver will be discussed:

- vectorization
- matrix-free implementation
- parallel computing

4.6.1 Vectorization

As the main computing platform used at NLR is a *NEC SX-8R 8 processor shared memory vector machine*, an appropriate formulation of the black box code is necessary: it should be able to perform vector calculations. To give an intuitive feeling for vectorization, consider the following example. A matrix-multiplication $A \times B = C$ for 5 rows and 5 columns would mean that a 'traditional' machine without vector optimization would need $5^2 = 25$ steps of addition and multiplication for getting the result C.
A vector machine deals with a whole row and column at once, resulting in only 5 clockcycles. In other words: "traditional" iterative approaches require $O(N^2)$ clockcycles to perform an matrix-

multiplication, whereas vector-approaches run linear in $O(N)$, if N is the total number of unknowns.
Another property of vectorization is the need for so called *direct addressing* in loop structures: the location of the variables in memory handled in the next step of the loop, can be computed trivially.

This will be explained by the following example:

The goal is to compute the product $y = Ax$ which can be achieved by: $y(i) = \sum_k A(i,k)x(k)$, for a certain i. Note that one way of storing a sparse matrix A is by using the following arrays:

- $A(i,k)$: the value a_{ik}
- $L(i)$: the number of the row i
- $K(i,k)$: the number of the column k

Assume that matrix A has dimensions $N \times N$. To perform the product, direct addressing uses the vectorization as follows:
For a certain $i = 1..N$
    for $k = 1..N$
        $y(i) = A(i,k) * x(k)$
    end
End

This is done in $O(N)$ clockcycles.

Opposite to direct addressing is indirect addressing:
For a certain $i = 1..N$
    for $k = 1..L(i)$
        $y(i) = A(i,k) * x(K(i,k))$
    end
End

This is done in $O(N \times \text{length}(L(i)))$ clockcycles. When $\text{length}(L(i)) = N$, this indirect multiplication is performed in $O(N^2)$ clockcycles.

Finally, a direct consequence of vectorization is the need of a matrix-free implementation, which will be treated in the next subsection.

4.6.2 Matrix-free implementation

In this subsection the matrix-free implementation in the existing algorithms will be explained. The *direct solver* and the *iterative solver* proposed by Hooghiemstra (ref. 8 and 9) both use the matrix-free implementation. The direct solver avoids assembling the element matrices in the system matrix analogous to the iterative solver proposed by Hooghiemstra. In addition, this iterative solver also avoids indirect addressing. Incorporating algebraic multigrid in the existing algorithm, should preferably also avoid assemblage and use direct addressing.
In the next subsections these three approaches are discussed.

4.6.2.1 Direct solver

It is well known that greatest disadvantage of direct solvers are the large memory requirements. An example of a direct solver that can be used with the finite element method, is the *frontal solution method.* This method keeps track of a number of *active* variables and by continuously adding new variables, other variables are fully summed and will be eliminated. In general there are only a few variables in action at the same time which form a *front,* explaining the name of the method. For more about this method, the reader is referred to Hooghiemstra (ref. 8, Section 4.2.2).
In the next subsection, the solution procedure proposed by Hooghiemstra will be discussed.

4.6.2.2 Present implementation – GCR as iterative solver for the preconditioner system

The idea proposed by Hooghiemstra is already explained in Subsection 3.3.3 and is repeated here for convenience: the idea is to combine the triangular preconditioner with the shifted Laplace preconditioner, resulting in:

$$M_{new} = \begin{bmatrix} M_2 & A_{12} \\ 0 & A_{22} \end{bmatrix} \tag{4.6.1}$$

and the new preconditioned system to be solved is given by:

$$M_{new} s^k = r^{k-1} \tag{4.6.2}$$

This is done in two steps:

1. solve $A_{22} s_2^k = r_2^{k-1}$ with a precomputed LU decomposition of A_{22} (this has to be computed only once)
2. solve $M_2 s_1^k = r_1^{k-1} - A_{12} s_2^k$ using GCR

This preconditioner is constructed in an efficient way and the matrix M_2 originates from the finite element discretization of the shifted Laplace operator. The product in the second step above is performed matrix-free, which means that the matrix M_2 is not assembled from the element

matrices[14]. The advantage here is that it is not necessary to store the complete matrix and for large sparse matrices this leads to a huge saving in memory. Furthermore, the multiplication with the sparse matrix is done as a sequence multiplications with the densely populated element matrices, eliminating the need for indirect addressing. However, when the total number of unknowns increases, the total work to store the entire Krylov base for the GCR method becomes unacceptably high (recall that GCR is a long recurrence method).
In the next subsection the idea of using algebraic multigrid as solver for the preconditioner system will be discussed.

4.6.2.3 Present implementation – AMG as iterative solver for the preconditioner system

Another way of solving the preconditioner system $M_2 s_1^k$ is by using algebraic multigrid as iterative solver. It is expected that using AMG will lead to the following advantages:

- the preconditioner will remain constant
- a constant preconditioner makes it possible to use a short recurrence method
- a short recurrence method leads to an considerable reduction in the storage requirements

Note

The fact that multigrid methods can solve a system of M unknowns in cM arithmetic operations, will not lead to a great gain compared to the work of using GCR as proposed by Hooghiemstra (ref. 8). The main reason for GCR to perform poor, are the storage requirements when M increases (when $M \geq 5.0 \cdot 10^5$ the memory requirements can not be met). Here M denotes the number of degrees of freedom. This is not necessarily equal to the total number of unknowns N.

As in the current application the element matrices are used without assembling, it would be advantageous to do that also when using AMG. Furthermore, as the element matrices contain useful information for computational aspects, it is required that the AMG black box solver uses the element matrices as one of the input parameters, instead of the whole system matrix.
In the next subsection a final requirement for the black box solver is stated.

[14]Note that the size of the element matrices depends on the order of the vector basis functions used. In the current application, second order tetrahedral elements are used, which results in 45 basis functions $\Rightarrow$ size of element matrix is 45

4.6.3 Parallel computing

The present algorithm is not parallelizable and hence it runs on a single processor. When a domain decomposition method would be applied, it is possible to parallelize the algorithm. As one property of multigrid methods is the possibility of parallel application, the AMG black box should be usable for parallel computing.

4.7 Choosing the most appropriate AMG black box solver

Based on the criteria for the algebraic black box solver formulated in the previous section, an appropriate algebraic multigrid black box algorithm must be chosen. Unfortunately, at the moment, there are no black box solvers available which are suitable for vector computations nor black box solvers which have a matrix-free implementation. However, these requirements do not have to be fulfilled to be able to proof the concept of AMG acceleration for this application. According to the information about available multigrid solvers (S.P. Maclachlan and C.W. Oosterlee, ref. 14), the most suitable black box solver for the current application is a Multilevel (ML) Preconditioning Package developed by Sandia National Laboratories.

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, including the Maxwell equations (see ref. 12). When it turns out that this AMG concept significantly improves the solution procedure, alternative ways to fulfill the requirements of vector computations and matrix-free implementation will be sought.

5 Conclusion & recommendations

In this thesis a 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. The resulting system can be denoted by

$$Au = f$$

where matrix A is indefinite, ill-conditioned, 'nearly' symmetric (but not Hermitian), partly sparse and partly fully populated.

In ref. 7, Erlangga proposed an iterative method to effectively solve the discrete Helmholtz equation in two and three dimensions at very high wavenumbers. He showed that *multigrid* can be applied to a properly chosen preconditioning operator, namely the *shifted Laplace operator*. As the Helmholtz equation and the Maxwell's equations have similar properties, it is expected that multigrid will also be very effective to use in the present implementation.

Using multigrid in order to optimize the storage requirements in the current application, will result in a so called *constant preconditioner* (in the present implementation the preconditioner is changed every iteration). When the preconditioner remains constant, the GCR method used in the current algorithm can be replaced by a *short recurrence method* e.g. Bi-CGSTAB. As short recurrence methods use only a few of the latest basis vectors to generate the new basis vector, the memory requirements will be reduced significantly compared to the existing implementation.

Multigrid methods can be classified as geometric or algebraic depending on the availability of the underlying grid and the definitions of the smoothing operators. Geometric multigrid strongly depends on structured grids, which is not the case in the current application. Therefore, algebraic multigrid will be used to be incorporated in the current solution algorithm. Since it is not possible to implement a full algebraic multigrid solver in the time available, and given the fact that several algebraic black box solvers are available, it is recommended to choose an appropriate black box solver to incorporate in the existing algorithm. To achieve this efficiently, the black box solver also has to fulfill certain requirements discussed in Section 4.6:

- vectorization
- matrix-free implementation
- parallel computing

Unfortunately, at the moment, there are no black box solvers available which are suitable for vector computations nor black box solvers which have a matrix-free implementation. The most suitable black box solver for the current application is a Multilevel (ML) Preconditioning Pack-

age developed by Sandia National Laboratories. This AMG algorithm will be incorporated in the existing algorithm.

6 Future research

The chosen AMG algorithm will be incorporated in the iterative solver of Hooghiemstra (ref. 8 and 9), and will replace the solution algorithm for the preconditioner solve. Hereafter, the dependence of the AMG algorithm on different parameter settings, will be evaluated (e.g.: number of pre- and postsmoothings, cycle type, etc.).

Additionally, the GCR algorithm used by Hooghiemstra, will be replaced by an appropriate short recurrence method, e.g. Bi-CGSTAB or IDR(s).

References

1. Michele Benzi. Block Preconditioning for Markov Chain Problems. *Markov Anniversary Meeting Charleston, SC, 13 June 2006*, 2006.
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. J.E. Dendy. Black box multigrid for nonsymmetric problems. *SIAM J. Sci. Comput.*, 13:261–283, 1983.
5. 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.
6. E.F. Knott, J.F. Shaeffer and M.T. Tuley. *Radar Cross Section*. Artech House, Inc., 1985.
7. Yogi Ahmad Erlangga. A robust and efficient iterative method for the numerical solution of the Helmholtz equation. Thesis, TU DELFT, 2005.
8. 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.
9. 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 eqaution. Report NLR-TR-2007-741, NLR, 2007.
10. 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.
11. L. Giraud, S. Gratton, J. Langou. A note on relaxed and flexible GMRES. Report TR/PA/04/41, CERFACS, 2004.
12. Sandia National Laboratories. http://trilinos.sandia.gov/packages/ml/index.html.
13. M. Brezina, A.J. Cleary, R.D. Falgout, V.E. Henson, J.E. Jones, T.A. Manteuffel, S.F. Mccormick and J.W. Ruge. Algebraic multigrid based on element interpolation. *SIAM J. Sci. Comput.*, 22(5):1570–1592, 2001.
14. Scott P. Maclachlan and Cornelis W. Oosterlee. Private conversation.
15. Scott P. Maclachlan and Cornelis W. Oosterlee. Algebraic Multigrid Solvers for Complex-Valued Matrices. *SIAM J. Sci. Computing*, 30(3):1548–1571, 2008.
16. 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.

17. R. Barrett, M. Berry, T.F. Chan, J. Demmel, J. Donato, J. Dongara, V. Eijkhout, R. Pozo, C. Romine and H. van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM, Philadelphia, 1994.
18. Andrew F. Peterson Robert D. Graglia, Donald R. Wilton. Higher Order Interpolatory Vector Bases for Computational Electromagnetics. *Journal of Aircraft*, 1997.
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. Ulrich Trottenberg, Cornelis Oosterlee, Anton Schüller. *Multigrid*. Academic Press, 2001.
21. Valérie Fraussé, Luc Giraud, Serge Gratton. A Set of Flexible GMRES Routines for Real and Complex Arithmetics on High Performance Computers. Report TR/PA/06/09, CERFACS, 2006.
22. R.S. Varga. On recurring theorems on diagonal dominance. *Linear Algebra Appl*, 13:1–9, 1976.
23. R. Wienands and C. W. Oosterlee. On three-grid Fourier analysis for multigrid. *SIAM J. Sci. Comput.*, 23:651–671, 2001.
24. William L. Briggs, Van Emden Henson, Steve F. McCormick. *A Multigrid Tutotrial*. SIAM, Philadelphia, 2000.
25. Y. Saad and M.H. Schultz. *GMRES: a generalized minimal residual algorithm for solving non-symmetric linear systems*. SIAM, Philadelphia, 1986.
26. Yousef Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, 2003.

This page is intentionally left blank.

Appendix A Electromagnetic quantities

In this appendix the basic SI (International System of Units) are discussed. In Table 4 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:

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

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

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

Table 4 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: $(.,.)$

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

Hölder p-norms

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

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

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

$$||x||_1 = |x_1| + |x_2| + \ldots + |x_n|$$
$$||x||_2 = [|x_1|^2 + |x_2|^2 + \ldots + |x_n|^2]^{\frac{1}{2}}$$
$$||x||_\infty = \max_{i=1,\ldots,n} |x_i|$$

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:

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

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

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:

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

and the following two important properties:

- the two subspace Ker(P) and 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 $(.,.)_{A_h}$ and is defined as

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

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*:

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

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 Independent and group sets

In this appendix independent set orderings are treated followed by group independent sets (also see Saad, ref. 26, Chapter 3).

Independent set ordering

ISOs are effectively applied on matrices with the following structure:

$$A = \begin{bmatrix} D & E \\ F & C \end{bmatrix} \tag{C.0.3}$$

in which D is diagonal and C, E and F are sparse matrices. The upper diagonal block corresponds to unknowns from the previous levels of refinement and its presence is due to the ordering of equations in use. As new vertices are created in the refined grid, they are given new numbers and the initial numbering of the vertices in unchanged. since toe old connected vertices are "cut" by new ones, they are no longer related by equations. Such sets are called *independent sets* and they are especially useful in parallel computing for implementing direct and iterative methods.
Referring to the adjacency graph $G = (V, E)$ of the matrix and denoting by (x, y) the edge from vertex x to vertex y, an independent set S is a subset of the vertex set V such that:

$$\text{if } x \in S, \quad \text{then } \{(x, y) \in E \text{ or } (y, x) \in E\} \rightarrow y \notin S$$

So elements of S are not allowed to be connected to other elements of S either by incoming of outgoing edges.
An independent set is *maximal* if it cannot be augmented by elements in its complement to form a larger independent set (a maximal independent set is not necessarily the largest possible independent set that can be found).

Group independent set

A group independent set is a collection of subsets of unknowns such that there is no coupling between unknowns of any two different groups. Unknowns within the same group may be coupled (see Figure C.0.1).

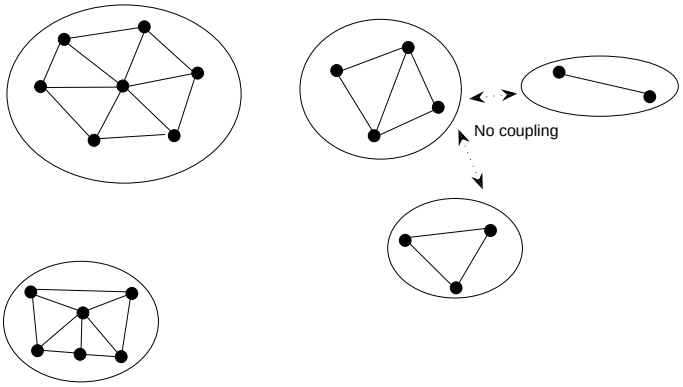

Fig. C.0.1 Group (or block) independent sets