Iterative solvers and preconditioning for electromagnetic boundary integral equations

Iteratieve oplosmethodes en preconditionering voor elektromagnetische randintegraal vergelijkingen

(met een samenvatting in het Nederlands)

Proefschrift

ter verkrijging van de graad van doctor aan de Universiteit Utrecht op gezag van de Rector Magnificus, Prof.dr. H.O. Voorma, ingevolge het besluit van het College van Promoties in het openbaar te verdedigen op maandag 5 maart 2001 des middags te 16.15 uur

door

Menno Ewout Verbeek

geboren op 13 maart 1972, te Vlaardingen

Promotor:	Prof.dr. H.A. van der Vorst
	Faculteit der Wiskunde en Informatica
	Universiteit Utrecht
Co-promotor:	Dr.J.R.M. Bergervoet
	Philips Research

Dit onderzoek maakt deel uit van het ELSIM project van het Platform HPCN en is een samenwerkingsverband tussen de Universiteit Utrecht en Philips Research.

2000 Mathematics Subject Classification: 65F10, 65F35, 65N38, 65N55, 65R20, 65Z05, 78A40, 78M05, 78M15.

Verbeek, Menno Ewout
Iterative solvers and preconditioning for electromagnetic boundary integral equations
Proefschrift Universiteit Utrecht – Met samenvatting in het Nederlands.

ISBN 90-393-2655-X

Contents

1 Introduction **1**
- 1.1 Overview . . . 4
- 1.2 Iterative solvers . . . 5
  - 1.2.1 Minimal residual Krylov subspace methods . . . 5
  - 1.2.2 GMRES . . . 5
  - 1.2.3 Convergence properties . . . 7
  - 1.2.4 Preconditioning . . . 9
  - 1.2.5 Ritz pairs . . . 10

2 Discretisation **11**
- 2.1 Some physics . . . 11
  - 2.1.1 The driving force . . . 15
- 2.2 Petrov-Galerkin approach . . . 15
  - 2.2.1 Thin boards and wires approximation . . . 16
- 2.3 Wire model . . . 17
  - 2.3.1 The choice of basis and test functions . . . 18
  - 2.3.2 Drawbacks of the wire model . . . 20
- 2.4 Surfaces model . . . 21
  - 2.4.1 The wire approximation to surfaces . . . 22
  - 2.4.2 The point approximation to surfaces . . . 24
- 2.5 Convergence results . . . 25
  - 2.5.1 The test problem . . . 26
  - 2.5.2 Results . . . 28
- 2.6 Choosing an integration variant . . . 29
- 2.7 Fast multipole method . . . 31

3 Basis transformation **35**
- 3.1 Matrix properties . . . 35
  - 3.1.1 Fourier analysis . . . 37
  - 3.1.2 Discussion of the Fourier analysis . . . 44
- 3.2 Constructing a new basis . . . 45
- 3.3 The continuous analogue . . . 50
  - 3.3.1 Specification of div^+ and $\mathbf{grad}^+$ . . . 50
  - 3.3.2 The new operator . . . 52
  - 3.3.3 Special case: the infinite plane conductor . . . 52
  - 3.3.4 Fourier analysis on the infinite plane conductor . . . 53

3.3.5 Discussion of the Fourier analysis . . . 56
3.4 Properties of the transformed matrix . . . 56
3.5 Implementation details . . . 60

4 Geometric multigrid 65
4.1 Multigrid in a nutshell . . . 65
4.2 Smoother . . . 67
4.2.1 Frobenius norm minimisation . . . 68
4.2.2 Preconditioning with truncated interaction . . . 69
4.3 Coarse grid correction . . . 73
4.4 Implementation details . . . 75
4.4.1 Computation of matrices . . . 75
4.4.2 Regularisation of an LU-decomposition . . . 76
4.5 Experimental results . . . 79
4.6 Conclusions . . . 82
4.7 Combination with the Fast Multipole Method . . . 83

5 Reuse of computational information 85
5.1 Multiple right-hand sides . . . 85
5.1.1 GMRESR . . . 86
5.1.2 GMRESR and search space injection . . . 87
5.1.3 Selecting information . . . 90
5.2 Frequency extrapolation . . . 94
5.3 Conclusions . . . 97

6 Algebraic Multigrid 99
6.1 Introduction . . . 99
6.2 Algebraic multigrid framework . . . 100
6.3 The Ruge-Stüben approach . . . 102
6.4 An alternative approach . . . 104
6.5 The interpolation . . . 105
6.5.1 Linear interpolation . . . 105
6.5.2 Adaptation . . . 106
6.5.3 Algebraically smooth vector . . . 107
6.6 The restriction . . . 108
6.7 Experimental results for transport problems . . . 108
6.8 Adaptation for A_Q . . . 111
6.9 Experimental results for electromagnetic boundary integral equations . . . 112
6.10 Discussion . . . 113

Bibliography 115

Samenvatting 119

Dankwoord 123

Curriculum Vitae 124

Chapter 1

Introduction

The research presented in this thesis, was motivated by the need for a faster solver for the electromagnetic compatibility simulation code under development at Philips Research.

Electromagnetic compatibility (EMC) deals with the interaction between an electric apparatus and its electromagnetic environment. An electromagnetically compatible apparatus should be able to function in its electromagnetic environment, and it should not generate electromagnetic interference for anything in its environment. As such, EMC has two separate requirements. First, an apparatus should have a certain degree of immunity against external electromagnetic interference. Second, the apparatus should not create electromagnetic interference for other apparatus.

An example where EMC plays a role in a domestic situation, is when the video recorder is located right next to the television. If they are not electromagnetically compatible, switching on the video recorder might distort the television image, making it impossible to watch a video tape with high quality. Absence of EMC could have much graver consequences in, for instance, the operating rooms of a hospital, where many vital electrical appliances are close together. Another part of EMC is that an apparatus should not interfere with radio communications ranging from normal FM radio to air traffic guidance beacons.

The electromagnetic disturbances can range from power voltage fluctuations to radiated high-frequency electromagnetic fields. The term EMC applies to this whole spectrum of phenomena, with frequencies ranging from 0 Hz to the GHz (10^9 Hz) range.

The International Electrotechnical Commission (IEC) publishes EMC standards for four different product categories [25] :

Component A unit that has no final function in itself, but is intended for incorporation in an apparatus. For example, a single capacitor, an integrated circuit or a power supply unit.

Apparatus A finished product with a direct function intended for final use. For example domestic appliances like a video recorder or medical equipment like a heart monitor.

System A combination of components and apparatus constituting a single functional unit. For example, a computer system with a computer, keyboard, mouse, monitor, printer, etc.

Installation A combination of components, apparatus and systems in a given area, for example an industrial plant.

The standards consist of two parts, an emission and an immunity standard, and are differentiated between domestic and industrial use. Some standards are used for legislation, such as the standards from the International Special Committee on Radio Interference (CISPR), which is now part of the IEC. The IEC standards can also serve as a recommendation, or they can be used in commercial contracts.

Due to the increased use of high frequency digital components, it is increasingly important to take the EMC requirements into account when designing an apparatus. Depending on the apparatus and its use, the design will have to comply with one or more of the EMC standards. Building a prototype and measuring its EMC properties can be an expensive and time consuming process. In order to reduce the design cost, and most importantly, in order to reduce the design time, it is desirable to be able to compute the electromagnetic behaviour of a design using fast computer simulations. This research project focused on the simulation of the high frequency radiation emission at the level of the whole apparatus.

This type of simulations can also be used to compute electromagnetic scattering behaviour (such as radar reflections) and antenna behaviour. These problems all deal with phenomena that have oscillatory behaviour with respect to time. By using Fourier transformations, this type of calculations can also be used as a basis for computations that deal with very different time dependence, so-called transient behaviour. One example is the simulation of electrostatic discharges.

In this thesis, we will use a strongly simplified model of the apparatus. The non-conductive elements of the apparatus, for example the plastic parts, have very little influence on the electromagnetic behaviour and are omitted from the model. What remains are the conducting parts, which usually consists of parts of the case, wires and printed circuit boards. The printed circuit boards are very complex components, containing all the electronics and a fine pattern of conducting lines. For the high frequency range we are interested in, the fine pattern of conducting lines combined with the ground plane that is present in the printed circuit board, can be simplified to a single conducting surface.

The electric circuit in the apparatus generates the currents that cause the emission of radiation, but the EMC simulation does not attempt to simulate the electric circuit. The effects of the electric circuit are modelled with power sources at the points where the printed circuit boards are connected to the wires. In order to know what power sources to use, the behaviour of the electric circuit during operation of the apparatus must be known. This information can be obtained using a separate circuit simulator or by measurements in an experimental setup of the circuit. In the current situation, the effects of electromagnetic fields on the functioning of the electric circuit are not in the scope of the EMC simulations. These effects, like crosstalk between lines on a printed circuit board, can be included in the electric circuit description using separate simulations.

The result of these simplifications is that the models we use consist of a number of thin, flat, rectangular conducting plates and thin wires. An example of such a model is shown in Figure 1.1. The methods we apply are not necessarily restricted to these simple shapes, but our code does not yet support more complex geometries.

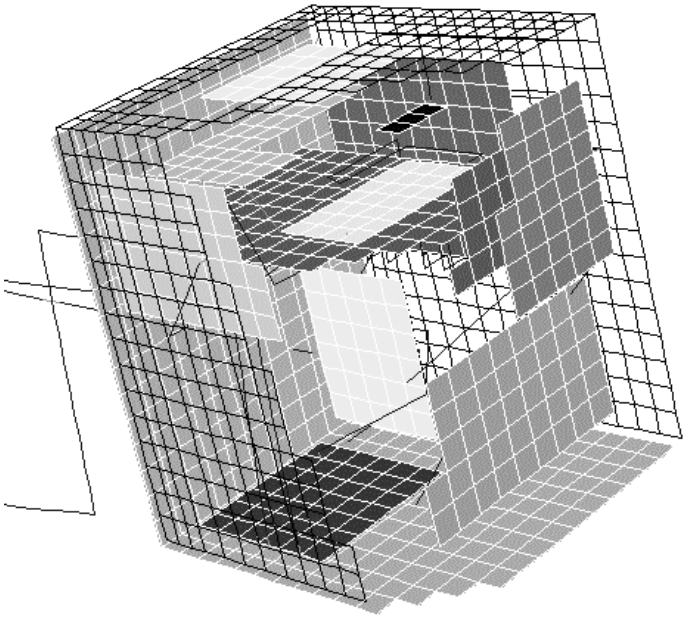


FIGURE 1.1: A model of a hi-fi set. The external wires lead to the speakers and the antenna, and are much longer than shown here. Some parts have been omitted in order to show the interior.

The EMC simulation package under development at Philips Research is called EMIR. This contains a computation kernel in the form of a separate program named BERBER, which we have been working on. The BERBER package numerically solves a boundary integral equation, and is the most time-consuming part of an EMIR run. The numerical treatment of the boundary integral equation leads to a dense matrix equation, as is further described in Chapter 2.

At the start of this project, the linear system was solved using a direct solver based on LU-decomposition. One possible way of reducing the time needed for a simulation, is by parallelising the BERBER program. We did this using the SCALAPACK package [11] and tested this on a 12 processor shared memory SGI Power Challenge. The extra overhead due to the parallelism was relatively small with respect to the large amount of computational work to be done. This, combined with a more efficient use of the cache memory, lead to parallel efficiencies of approximately 1. This means that using p processors resulted in a reduction of the execution time with a factor of approximately p. Naturally, we could only use $p \leq 12$ on this computer.

One disadvantage of this method is that, in order to reduce the computation time, the program must be run on an expensive parallel computer. Another disadvantage is, that when the problem size increases, the computation time will still increase as fast as before parallelisation. For the direct solver used, this grows proportional with the cube of the number of degrees of freedom, which is called an $\mathcal{O}(n^3)$ method.

By using a fast iterative solver, this might be reduced, but due to the dense $n \times n$ matrix, this cannot be faster than $\mathcal{O}(n^2)$. The only way to get a lower order, is by using

matrix free methods. When using a matrix free method, the linear system matrix is not computed explicitly, but only a matrix-vector multiplication with the matrix can be computed. This implies that direct solvers cannot be used, and one is forced to use iterative solvers. A well known matrix-free method is the fast multipole method (FMM) which we discuss briefly in section 2.7.

In this project, we have worked on an efficient iterative solver for the electromagnetic boundary integral equation used in Berber. We have developed this iterative solution method such that, with some extra work, it can be used in combination with a matrix-free fast multipole method. Since this fast multipole method is rather complicated to implement, and since this is still an active field of research, we have not implemented this matrix-free method, but we have concentrated on the iterative solver.

1.1 Overview

We finish this chapter with section 1.2, that contains a quick introduction to the basis of the iterative method we will use in this thesis.

In chapter 2, we start with a brief description of the physics involved in the EMC simulation, and pose the boundary integral equation that we will solve numerically. Next, we discuss the discretisation of this equation and show some experimental results for the accuracy of the discretisation.

The matrix equation we found in chapter 2 is analysed in chapter 3. This shows that, especially for the relatively low frequencies, the matrix is very badly conditioned in such a way that this is very hard to precondition. In order to make effective preconditioning possible, we propose a new basis transformation that separates the capacitive and inductive terms, and we analyse some properties of the transformed matrix.

In chapter 4, we construct a geometric multigrid preconditioner for the transformed matrix. We start with the design of a suitable smoother, followed by the construction of a coarse grid correction mechanism and experimental results. We also address some implementation complications due to the basis transformation. The chapter is finished with a discussion of a possible combination of multigrid with the matrix-free fast multipole method.

We discuss the reuse of computational information in chapter 5. The first opportunity to reuse information results from the fact that, in general, we have to solve for many right-hand sides. Using search space injection, we are able to remove the initial phase of slow convergence from the iterative solve, thereby reducing the number of iterations for the second and later right-hand sides. Another possibility to reuse information, results from the fact that the linear system must be solved for a range of frequencies. We use the fact that the solutions for nearby frequencies might not differ very much.

The beginning of chapter 6 describes research done in collaboration with J.Cullum at the Los Alamos National Laboratory and deviates from the central theme. We discuss some new ideas for the construction of a smoother dependent interpolation operator for algebraic multigrid, and apply this to transport problems. Next, we generalise this method so it can be applied to the electromagnetic boundary integral problems used before.

1.2 Iterative solvers

In this section we will give a short introduction to minimal residual Krylov subspace methods, and we will describe one often used variant for general matrices named GMRES (Generalised Minimal RESidual) [36]. For a broader introduction to iterative methods for linear systems and more references, see the Templates book [6].

1.2.1 Minimal residual Krylov subspace methods

We will consider solving the linear system

$$A\bar{x} = b \quad . \tag{1.1}$$

In some situations, we will want to use an already available approximation x_0 of the solution $\bar{x}$. This approximate solution x_0 is called the initial guess, and is subtracted from the linear system (1.1) :

$$A(\bar{x} - x_0) = b - Ax_0 \quad \Leftrightarrow \quad Ax = r_0 \quad , \tag{1.2}$$

with $x = \bar{x} - x_0$ and $r_0 = b - Ax_0$. This leads to a new right-hand side r_0 that is (hopefully) smaller than b. For the new system, we start with $x_0 = 0$.

When solving the linear system

$$Ax = r_0 \tag{1.3}$$

with a Krylov subspace method, step by step an orthogonal basis for the Krylov subspace

$$\mathcal{K}_k(A, r_0) = \langle r_0, Ar_0, A^2r_0, \ldots, A^{k-1}r_0 \rangle \tag{1.4}$$

is constructed. In each step this basis is extended to form a basis for the Krylov subspace of one dimension larger, $\mathcal{K}_{k+1}(A, r_0)$. This basis is used to find the optimal solution $x_k \in \mathcal{K}_k(A, r_0)$, and thus $\mathcal{K}_k(A, r_0)$ is appropriately called the search space. Since the error $e_k = x - x_k$ is unknown, the residual $r_k = Ae_k = r_0 - Ax_k$ is used to define what is optimal :

$$x_k = \underset{x_k \in \mathcal{K}_k(A, r_0)}{\arg\min} \; \|r_0 - Ax_k\|_2 \quad . \tag{1.5}$$

Expansion of the search space is repeated until this approximation is thought to be accurate enough. The criterion for this is called the stopping criterion, and is usually related to the relative size of the residual :

$$\frac{\|r_k\|_2}{\|b\|_2} \leq \epsilon_{\text{tol}} \quad , \tag{1.6}$$

where ϵ_{tol} is a user specified tolerance.

1.2.2 GMRES

GMRES stands for Generalised Minimal RESidual, and is one of the most popular minimal residual Krylov subspace algorithms since it can be applied to any square non-singular linear system. We will describe the algorithm here.

As mentioned before, an orthonormal basis is generated for the Krylov subspace :

$$V_{k+1} = [v_1, \ldots, v_{k+1}] \quad \text{with} \quad \langle V_{k+1} \rangle = \mathcal{K}_{k+1}(A, r_0) \quad \text{and} \quad V_{k+1}^{\mathrm{H}} V_{k+1} = I \ . \tag{1.7}$$

This basis is generated with the Arnoldi iteration, which we will explain now. One starts with $V_1 = [\, r_0/\|r_0\| \,]$, which is a basis for $\mathcal{K}_1(A, r_0)$. If the Arnoldi basis V_k for $\mathcal{K}_k(A, r_0)$ is already available, it is extended with the direction Av_k to get V_{k+1}. In order to make V_{k+1} orthonormal, this new vector is first orthogonalised with respect to V_k before it is added to the basis.

Now note the Krylov subspace property that $A\mathcal{K}_k(A, r_0) \subset \mathcal{K}_{k+1}(A, r_0)$, which implies that there is an $\underline{H}_k \in \mathbb{C}^{(k+1)\times k}$ such that

$$AV_k = V_{k+1}\underline{H}_k \ . \tag{1.8}$$

If we introduce the notation h_{ij} for the elements of $\underline{H}_k$, we can write the last column of this matrix equation as

$$Av_k = \sum_{j=1}^{k+1} h_{jk} v_j \quad \Leftrightarrow \quad v_{k+1} = \left(Av_k - \sum_{j=1}^{k} h_{jk} v_j \right) / h_{k+1,k} \ , \tag{1.9}$$

which shows that the elements of $\underline{H}_k$ are precisely the values that have already been computed in the orthogonalisation of the basis V_{k+1}. It can also be seen that $\underline{H}_k$ has an upper Hessenberg structure ($h_{ij} = 0$ for all $i > j + 1$).

Once we have a basis for the Krylov subspace, $\langle V_k \rangle = \mathcal{K}_k(A, r_0)$ with $V_k^{\mathrm{H}} V_k = I$ and $AV_k = V_{k+1}\underline{H}_k$, we can rewrite the minimisation (1.5) as

$$\min_{x_k \in \langle V_k \rangle} \|r_0 - Ax_k\|_2 = \min_{y \in \mathbb{C}^k} \|r_0 - AV_k y\|_2 \tag{1.10a}$$

$$= \min_{y \in \mathbb{C}^k} \|V_{k+1}\rho_0 e_1 - V_{k+1}\underline{H}_k y\|_2 \tag{1.10b}$$

$$= \min_{y \in \mathbb{C}^k} \|\rho_0 e_1 - \underline{H}_k y\|_2 \ , \tag{1.10c}$$

where $\rho_0 = \|r_0\|_2$. The minimisation in (1.10c) is a small linear least squares problem of size $(k + 1) \times k$, that can be solved very efficiently. Once we have computed y we can construct $x_k = V_k y$. Note that we do not need to calculate x_k in order to find its residual since this is equal to the residual of the small minimisation problem.

By putting the above together, we obtain the GMRES algorithm, see Figure 1.2. The algorithm starts with some initialisations. The main loop contains the Arnoldi iteration for the extension of the search space V_k to V_{k+1} and the corresponding extension of $\underline{H}_{k-1}$ to $\underline{H}_k$. The new search direction is found by the matrix-vector product Av_k, which is then orthogonalised using modified Gram-Schmidt in the inner loop. Next, V_k and $\underline{H}_{k-1}$ are extended and the new minimal residual norm ρ_k is computed. If convergence is achieved within the specified tolerance, the approximate solution x_k corresponding to the minimal residual is computed.

Note that the GMRES algorithm does not require the matrix A to be explicitly available, only matrix-vector multiplications with A have to be computed.

With respect to the computational costs of GMRES, we see that there is one matrix-vector multiplication with A, and there are $k + 1$ vector inner products and $k + 1$ vector

Choose x_0
$r_0 = b - Ax_0$
$\rho_0 = \|r_0\|$, $v_1 = r_0/\rho_0$
$k = 0$, $V_1 = [v_1]$, $\underline{H}_0 = [\ ]$
while $\rho_k/\|b\| > \epsilon_{\text{tol}}$ do
    $k = k + 1$
    $v_{k+1} = Av_k$
    for $i = 1 \ldots k$ do
        $h_{ik} = v_i^{\text{H}} v_{k+1}$
        $v_{k+1} = v_{k+1} - h_{ik} v_i$
    $h_{k+1,k} = \|v_{k+1}\|$
    $v_{k+1} = v_{k+1}/h_{k+1,k}$
    $V_{k+1} = [V_k, v_{k+1}]$
    $\underline{H}_k = \begin{bmatrix} \underline{H}_{k-1} & (h_{1k} \ldots h_{kk})^{\text{T}} \\ 0 \ldots 0 & h_{k+1,k} \end{bmatrix}$
    Compute $\rho_k = \min_y \|\rho_0 e_1 - \underline{H}_k y\|_2$
$x_k = x_0 + V_k y$

Main property :
$x_k = \underset{x \in \mathcal{K}_k(A,b)}{\arg\min} \|b - Ax\|_2$

FIGURE 1.2: The GMRES algorithm with an initial guess x_0.

updates per iteration. Then there also is the minimisation problem, but since this is a small $(k+1) \times k$ problem, these costs are relatively small if k is small. In our situation, the matrix A is a dense $n \times n$ matrix, and the matrix-vector multiplication will cost n^2 complex multiplications and additions, while the inner products and vector updates together involve only $2(k+1)n$ of these combined operations. One such a combined complex multiplication and addition ($a = a + b \cdot c$) is equivalent to 8 floating point operations (flops). The conclusion is that if $k \ll n$ then the major part of the cost of one GMRES iteration is due to the matrix-vector multiplication.

1.2.3 Convergence properties

First we note that it may happen that the new direction Av_k is already in the search space, $Av_k \in \langle V_k \rangle$. This would result in $h_{k+1,k} = 0$ and we have a division by zero. However, if this happens, $\underline{H}_k$ will be essentially square, and the minimum in equation (1.10c) will be zero. As a result, x_k will be the exact solution. This is therefore called a lucky breakdown. In exact arithmetic, this always happens within n steps, because if step n is reached, V_n will span the whole $\mathbb{C}^n$ and thus contain the exact solution. This implies that GMRES will always terminate (in exact arithmetic). However, if we really need n iterations, this would be very expensive. The method is most useful if it leads to

acceptable approximations, within the specified tolerance, in a relatively small number of iterations.

A very convenient way of interpreting the GMRES algorithm is by using polynomials in A. The elements of the Krylov subspace (1.4) can be written as a polynomial of degree $k-1$ in A times r_0 :

$$x_k \in \mathcal{K}_k(A, r_0) \quad \Leftrightarrow \quad x_k = p_{k-1}(A) r_0 \ , \tag{1.11}$$

with p_{k-1} a polynomial of degree $k-1$. Using this expression for the residual leads to

$$r_k = r_0 - A x_k = (I - A p_{k-1}(A)) r_0 \ , \tag{1.12}$$

showing that

$$r_k = q_k(A) r_0 \quad \text{with} \quad q_k(0) = 1 \ , \tag{1.13}$$

for some polynomial q_k. Since GMRES minimises r_k, it (implicitly) finds the polynomial q_k with $q_k(0) = 1$ that minimises r_k. A general form for this polynomial is

$$q_k(z) = \prod_{i=1}^{k} \left(1 - \frac{z}{\beta_i} \right) \ , \tag{1.14}$$

where the β_i are the zeroes of the polynomial.

Let us assume that A is diagonalisable, i.e. A has a set of eigenvectors U, such that

$$A = U \Lambda U^{-1} \ , \tag{1.15}$$

with Λ a diagonal matrix with the eigenvalues $\lambda_1, \ldots, \lambda_k$ of A on the diagonal. This allows us to rewrite equation (1.13) in the form

$$r_k = q_k(U \Lambda U^{-1}) r_0 = U q_k(\Lambda) U^{-1} r_0 \ . \tag{1.16}$$

Let us, for the moment, also assume that the matrix A is normal, i.e. $U^{\mathrm{H}} U = I$. This means that minimising $\|r_k\|$ is equivalent to minimising

$$\|q_k(\Lambda) s\|^2 = \sum_i |q_k(\lambda_i) s_i|^2 \ , \tag{1.17}$$

where we write s for $U^{-1} r_0$. This shows that, in order for the residual to be small, the polynomial $q_k(\lambda_i)$ must be small for all i for which s_i is not small. If we look at a worst case scenario in which none of the s_i is small, then all $q_k(\lambda_i)$ must be small.

Since GMRES will use the optimal polynomial q_k, it does better than any other polynomial $\widetilde{q}_k$ of degree k with $\widetilde{q}_k(0) = 1$. Using this, we see that if the spectrum $\sigma(A) = \{\lambda_i\}$ is clustered well away from the origin, then choosing a few zeroes $\widetilde{\beta}_j$ of the polynomial $\widetilde{q}_k$ in the cluster $\sigma(A)$ will already be able to give small values $\widetilde{q}_k(\lambda_i)$, and GMRES will converge quickly. However, if $\sigma(A)$ is spread out uniformly over a wide area close to the origin, or even around the origin, a much higher order polynomial is required in order to make all $\widetilde{q}_k(\lambda_i)$ small and still meet the requirement $\widetilde{q}_k(0) = 1$. As a result GMRES will converge more slowly. However, in the case where we have a clustered spectrum $\sigma(A)$ with one outlying large or small eigenvalue, we can get reasonably small

values $\widetilde{q}_k(\lambda_i)$ by taking one $\widetilde{\beta}_j$ at the outlying eigenvalue and a few in the cluster $\sigma(A)$. This indicates that the outlying eigenvalue will result in slower convergence compared with the system without the outlying eigenvalue, but the convergence is not as bad as in the case where the spectrum is evenly distributed over the region between the cluster and the outlying eigenvalue.

If we drop the assumption that the matrix A is normal, the U and U^{-1} in equation (1.16) may change this behaviour. The following estimate for the GMRES residual norm of step k can be derived [36]

$$\|r_k\| \leq \|U\widetilde{q}_k(\Lambda)U^{-1}r_0\|_2 \quad , \tag{1.18}$$

for all polynomials $\widetilde{q}_k$ of degree k with $\widetilde{q}_k(0) = 1$. This gives the following estimate for the relative residual norm

$$\frac{\|r_k\|_2}{\|r_0\|_2} \leq \kappa(U) \max_i |\widetilde{q}_k(\lambda_i)| \quad , \tag{1.19}$$

in which $\kappa(U) = \|U\|_2\|U^{-1}\|_2$ is the condition number of the eigenvector matrix. If we see the $\kappa(U)$ term as a fixed factor, the behaviour that we sketched for normal matrices would still hold, except for the extra factor. However, a strongly non-orthogonal eigenbasis U may change the actual convergence behaviour drastically. If the matrix is only mildly non-normal, then the behaviour that we sketched for normal matrices still gives a fair impression of the actual situation.

Estimate (1.19) can be used, in combination with cleverly chosen polynomials $\widetilde{q}_k$, to generate more explicit estimates for the convergence of GMRES.

1.2.4 Preconditioning

We have argued that GMRES converges relatively quickly if the matrix is not very non-normal ($\kappa(U)$ not too large) and the spectrum is nicely clustered away from the origin. However, the linear system matrix A often does not have these properties. In order to get faster convergence, the linear system has to be preconditioned, which means that it is replaced by an equivalent linear system, for instance

$$Ax = b \quad \Leftrightarrow \quad AMy = b \quad \text{and} \quad x = My \quad , \tag{1.20}$$

in which M is a non-singular matrix. This is called right preconditioning and changes the linear system matrix from A to AM. There are many different opinions on what properties a good preconditioner M should have. One (sufficient but not necessary) requirement is that AM should approximate the identity, which results in fast convergence for any Krylov subspace iterative method. This implies that we want to construct an operation $M \approx A^{-1}$. Just as for A, M does not have to be explicitly available in matrix form, we only have to apply the operator M to a vector. Alternative ways of preconditioning are left preconditioning, $MAx = Mb$, and two-sided preconditioning, $M_1AM_2y = M_1b$. We will mainly use right preconditioning, since this leads to residuals that are more easily interpreted.

At present, there are many iterative methods like GMRES for the solution of very wide classes of linear systems. However, there are no general purpose preconditioning

techniques that work for broad classes of problems, such as the positive definite matrices. Some methods are successfully applicable to smaller ranges of linear systems (e.g. ILU-decomposition [13, 28] and approximate inverses [8, 20, 7]). For many types of problems, including those coming from industrial applications, tailor made preconditioners are necessary in order to get an efficient iterative solver. Chapters 3, 4, and 6 of this thesis are about reformulating and preconditioning linear systems resulting from electromagnetic boundary integral equations.

1.2.5 Ritz pairs

The Arnoldi iteration can also be used to approximate eigenvalues and eigenvectors of A, which is usually done by calculating Ritz pairs. In this subsection we will give a short description of Ritz pairs, more information can be found in [30].

A Ritz pair of A with respect to the subspace $\langle V \rangle$ is a pair (θ, u) with $u \in \langle V \rangle$ and $\theta \in \mathbb{C}$ such that

$$Au - \theta u \perp \langle V \rangle \quad . \tag{1.21}$$

These Ritz pairs can be used as approximations to eigenpairs of A. The Arnoldi iteration can be used to calculate Ritz pairs by projecting the matrix A on the Krylov subspace $\langle V_k \rangle$

$$V_k^{\mathrm{H}} A V_k = V_k^{\mathrm{H}} V_{k+1} \underline{H}_k = H_k \quad , \tag{1.22}$$

with H_k the leading $k \times k$ block of $\underline{H}_k$. The eigenpairs of H_k correspond to Ritz pairs for A. This can be seen by

$$AV_k y - \theta V_k y \perp \langle V_k \rangle \quad \Leftrightarrow \tag{1.23a}$$

$$V_k^{\mathrm{H}}(AV_k y - \theta V_k y) = 0 \quad \Leftrightarrow \tag{1.23b}$$

$$V_k^{\mathrm{H}} A V_k y = \theta V_k^{\mathrm{H}} V_k y \quad \Leftrightarrow \tag{1.23c}$$

$$H_k y = \theta y \quad , \tag{1.23d}$$

which shows that $(V_k y, \theta)$ is a Ritz pair of A with respect to $\langle V_k \rangle$.

Alternatively, one can use harmonic Ritz pairs to approximate eigenpairs of A. A harmonic Ritz pair of A with respect to $\langle V \rangle$ is a pair (θ, u) with $u \in \langle V \rangle$ such that

$$Au - \theta u \perp A\langle V \rangle \quad . \tag{1.24}$$

This is equivalent with the requirement that $(1/\theta, Au)$ is a Ritz pair of A^{-1} with respect to $A\langle V \rangle$. To calculate harmonic Ritz vectors from the Arnoldi process is somewhat more complicated, and requires the use of the entire $\underline{H}_k$. In the context of a GMRES process, the harmonic Ritz values correspond to the roots β_i of the GMRES polynomial q_k (1.14).

Chapter 2

Discretisation

2.1 Some physics

The correct dynamical behaviour of electric charges and currents, electromagnetic fields, magnetisation and related quantities was first described in 1865 by Maxwell in, what we now know as the Maxwell equations. A thorough description of the theory can be found in, for instance, Jacksons book "Classical electrodynamics" [26]. In Gaussian units, the Maxwell equations read

$$\begin{aligned} \nabla \times \mathbf{H} &= \frac{4\pi}{c}\mathbf{J} + \frac{1}{c}\frac{\partial}{\partial t}\mathbf{D} \qquad & \nabla \cdot \mathbf{D} &= 4\pi\rho \\ \nabla \times \mathbf{E} &= -\frac{1}{c}\frac{\partial}{\partial t}\mathbf{B} \qquad & \nabla \cdot \mathbf{B} &= 0 \qquad . \end{aligned} \tag{2.1}$$

The different vector fields are the electric field $\mathbf{E}$, the displacement $\mathbf{D}$, the magnetic field $\mathbf{H}$, and the magnetic induction $\mathbf{B}$. Also appearing are the charge density ρ, the current density $\mathbf{J}$, and the light speed c. To get a complete description, these equations have to be combined with the continuity equation, describing the conservation of charge

$$\nabla \cdot \mathbf{J} + \frac{\partial}{\partial t}\rho = 0 \quad . \tag{2.2}$$

Also needed are equations that describe the medium properties, containing relations between $\mathbf{D}$ and $\mathbf{E}$, and between $\mathbf{B}$ and $\mathbf{H}$. These relations are determined by the polarisation and magnetisation of the medium.

In vacuum, where there is no medium, these relations are simply $\mathbf{D} = \mathbf{E}$ and $\mathbf{B} = \mathbf{H}$. If there are also no free charges, so $\rho = 0$ and $\mathbf{J} = 0$, the Maxwell equations reduce to

$$\begin{aligned} \nabla \times \mathbf{H} &= +\frac{1}{c}\frac{\partial}{\partial t}\mathbf{E} \qquad & \nabla \cdot \mathbf{E} &= 0 \\ \nabla \times \mathbf{E} &= -\frac{1}{c}\frac{\partial}{\partial t}\mathbf{H} \qquad & \nabla \cdot \mathbf{H} &= 0 \quad , \end{aligned} \tag{2.3}$$

which we will call the vacuum Maxwell equations. In the largest part of our computational domain, the medium will be air. However, for our application, air has a negligible polarisation and magnetisation, so we will use the simpler vacuum Maxwell equations for this part of the domain.

We can simplify this even further. Due to the linearity of the Maxwell equations, the sum of two solutions is again a solution. In this context, this is called the superposition principle, and it can be used to change from the time domain to the frequency domain by Fourier transformation. In practice this means that, instead of solving one time-dependent system, we will have to solve lots of time independent systems for a range of different frequencies. We will say more about this in section 2.1.1, but for now the result is that all quantities will behave harmonic in time with the same angular frequency ω. This means that we only have to know the amplitude and the relative phase of any quantity to know its complete time-dependent behaviour. This allows the Maxwell equations to be simplified even further by introducing implicit harmonic time dependence. Instead of working with a full time dependent quantity $A(\mathbf{x}, t)$, we perform our computations with a time independent complex quantity $A(\mathbf{x})$, of which the norm represents the amplitude of the real quantity and the argument represents the relative phase,

$$A(\mathbf{x}, t) = \mathrm{Re}\left(A(\mathbf{x})e^{i\omega t}\right) \quad . \tag{2.4}$$

From here on, we will use only the time independent complex quantities, unless a time dependence is explicitly mentioned. Using this convention in the vacuum Maxwell equations, we get a time independent set of equations

$$\begin{aligned} \nabla \times \mathbf{H} &= +ik\mathbf{E} & \nabla \cdot \mathbf{E} &= 0 \\ \nabla \times \mathbf{E} &= -ik\mathbf{H} & \nabla \cdot \mathbf{H} &= 0 \ , \end{aligned} \tag{2.5}$$

where $k = \omega/c = 2\pi/\lambda$ is called the wave number and λ is the wave length of electromagnetic waves. The continuity equation is changed correspondingly, which leads to

$$\nabla \cdot \mathbf{J} + i\omega\rho = 0 \tag{2.6}$$

for harmonic time dependence. This implies that ρ is fully determined if $\nabla \cdot \mathbf{J}$ is known, making ρ a derived variable.

Apart from the air, we also have to compute what happens with the conductors in our model. This is also described by the Maxwell equations, but now the medium properties are more complicated. For a perfect conductor, the electric field will not penetrate the interior of the conductor, and all interesting behaviour takes place at the conductor surface, which we will denote by Γ. For conductors, the electric and magnetic field will penetrate the conductor, but will decrease exponentially with the depth. The characteristic penetration depth is called the skin depth and is proportional to $(\omega\sigma)^{-1/2}$, where σ is the conductivity. For the frequency and conductivity range we are interested in, the skin depth is very small. The charge and current in the skin layer can effectively be seen as surface charge and current. From here on, we will be using the effective surface current density $\mathbf{J}$ and the effective surface charge density ρ. In fact, both quantities are the integral of the volume density over the very thin skin layer.

The continuity equation (2.6) still holds for these effective surface densities, but now we have to use the divergence restricted to the conductor surface Γ. Using the full Maxwell equations (2.1), it can be derived that this surface charge and current induce a discontinuity in the electric and magnetic fields at the conductor surface. Since the

fields are zero on the inside of the skin layer, we can write the jump condition for the parallel magnetic field $\mathbf{H}_{\|}$ as

$$\mathbf{n} \times \mathbf{H} = \mathbf{J} \qquad \text{on } \Gamma \ , \tag{2.7}$$

where $\mathbf{n}$ is the outward unit normal of Γ and $\mathbf{H}$ is evaluated at the outside limit to the conducting surface. From here on, when writing about the electric or magnetic fields at the conductor surface Γ, we implicitly mean the field evaluated at the outside limit to the conducting surface.

In the case of harmonic time dependence, condition (2.7) implies the jump condition for the orthogonal electric field $\mathbf{E}_{\perp}$,

$$\mathbf{n} \cdot \mathbf{E} = 4\pi\rho \qquad \text{on } \Gamma \ , \tag{2.8}$$

which can be seen by taking the surface divergence of (2.7) and using the continuity equation (2.6) and the vacuum Maxwell equations (2.5).

The parallel electric field $\mathbf{E}_{\|}$ is continuous across the surface layer. However, in the surface layer, there must be a parallel electric field in order to overcome the resistance. Ohms law states that the (volume) current density in a conductor is proportional to the electric field and proportional to the conductivity σ of the material. This causes a direct relation between $\mathbf{E}_{\|}$ just outside the surface, and the volume current density just inside the conductor, which is in turn directly related to the total current density $\mathbf{J}$. Using these relations, one finds that

$$\mathbf{E}_{\|} = Z\mathbf{J} \qquad \text{on } \Gamma \ , \tag{2.9}$$

where Z is called the surface impedance. Because of the depth of the skin layer, there is a phase shift between $\mathbf{E}_{\|}$ and $\mathbf{J}$, which makes the surface impedance Z a complex variable. For good conductors (i.e. when the conductivity approaches infinity, $\sigma \to \infty$),

$$Z \sim (1 - i)\, \omega^{-\frac{1}{2}} \sigma^{-\frac{3}{2}} \ . \tag{2.10}$$

When combining the conditions (2.7) and (2.9), we find a boundary condition for the fields that does not contain the current and charge,

$$\mathbf{E}_{\|} \equiv (\mathbf{n} \times \mathbf{E}) \times \mathbf{n} = Z\mathbf{n} \times \mathbf{H} \qquad \text{on } \Gamma \ . \tag{2.11}$$

This condition on the conductor surface can be augmented with conditions on the fields "at infinity", the so-called radiation conditions [15].

At this point, we have defined a set of partial differential equations (2.5) for the domain outside the conductor and a boundary condition (2.11) on the conductor surface. Together this defines the problem to solve.

One way to approach solving this system, is to eliminate the magnetic field $\mathbf{H}$ from the vacuum Maxwell equations (2.5), to get a vector Helmholtz equation augmented with a condition on the divergence

$$\Delta\mathbf{E} + k^2\mathbf{E} = 0 \qquad \nabla \cdot \mathbf{E} = 0 \ , \tag{2.12}$$

with the corresponding boundary condition

$$\mathbf{E}_{\|} = \frac{iZ}{k}\mathbf{n} \times (\nabla \times \mathbf{E}) \qquad \text{on } \Gamma \ . \tag{2.13}$$

This system has to be solved on the domain outside the conductor, which can be done using a finite element method. However, in our problems, the domain outside the conductor stretches out to infinity. There are possibilities to artificially truncate the domain, using an absorbing boundary condition, but in order to get a good approximation of the real situation, this boundary should be far from the conductor, leading to a large computational domain.

Another approach is to reformulate the problem as a boundary integral equation. From the physicist's point of view, we have to find the surface current for which the electric field it generates, combined with the external electric field, satisfy the boundary condition (2.9). The electric field that is generated by an oscillating current $\mathbf{J}$ can be separated in two parts. The capacitive part gives the electric field due to the charge accumulation ρ by the current, and is thus related to the spatial derivative of $\mathbf{J}$ via the continuity equation (2.6). The inductive part of the electric field is a consequence of the fact the current changes, and thus related to the time derivative of $\mathbf{J}$. Combining all contributions to the electric field, and using the implicit harmonic time dependence (2.4), we get

$$\mathbf{E}(\mathbf{x}) = -\int_\Gamma \nabla G(\mathbf{x},\mathbf{x}')\rho(\mathbf{x}')d^3x' - \frac{ik}{c}\int_\Gamma G(\mathbf{x},\mathbf{x}')\mathbf{J}(\mathbf{x}')d^3x' + \mathbf{E}^E(\mathbf{x}) \tag{2.14}$$

with G the Green function for this problem,

$$G(\mathbf{x},\mathbf{x}') = \frac{e^{-ik|\mathbf{x}-\mathbf{x}'|}}{|\mathbf{x}-\mathbf{x}'|} \ . \tag{2.15}$$

The contribution to $\mathbf{E}$ by the first integral of equation (2.14) is the capacitive part and the contribution by second integral is the inductive part. We added the electric field due to external sources $\mathbf{E}^E$ to get the total electric field $\mathbf{E}$. From the mathematician's point of view, this boundary formulation (2.14) can also be derived using representation formulas for the solution of the vector Helmholtz equation (2.12) [15]. More theory on integral equations can be found in [22].

When we combine the electric field in equation (2.14) with the boundary condition (2.9), we get

$$\frac{i}{\omega}\int_\Gamma \nabla_\Gamma G(\mathbf{x},\mathbf{x}')\nabla'\cdot\mathbf{J}(\mathbf{x}')d^2\mathbf{x}' + \frac{i\omega}{c^2}\int_\Gamma G(\mathbf{x},\mathbf{x}')\mathbf{J}(\mathbf{x}')d^2\mathbf{x}' + Z(\mathbf{x})\mathbf{J}(\mathbf{x}) = \mathbf{E}^E_\Gamma(\mathbf{x}) \qquad \forall_{\mathbf{x}\in\Gamma} \ . \tag{2.16}$$

This linear integral equation gives a direct relation between the external electric field $\mathbf{E}^E$ and the induced current $\mathbf{J}$, and is called the Electric Field Integral Equation (EFIE). We will use this equation to compute the induced current $\mathbf{J}$, which can in turn be used to compute the radiated electric field $\mathbf{E}$ in the whole space using equation (2.14).

Alternatively, we could have eliminated the electric field from the vacuum Maxwell equations (2.5), which would have led to the Magnetic Field Integral Equation (MFIE),

which is equivalent to the EFIE. We can even consider working with a linear combination of the EFIE and the MFIE called the Combined Electric Field Integral Equation (CFIE). The EFIE and MFIE are known to have problems with internal resonances [15], while using the CFIE can prevent these problems. In our situation, the model consists of thin boards and wires which have only a very small interior. In the thin boards and wires approximation we will use in section 2.2.1, we will effectively remove the interior. As a result, there will be no internal resonances, so we will work with the EFIE (2.16).

2.1.1 The driving force

The external electric field $\mathbf{E}^E$ is the driving force in all the equations we saw so far, but in a model description of an electrical device, the driving forces will be modelled as voltage sources, usually located at the connection between the wires and the boards. These voltage sources can be simulated by very local external electric fields at those points. By splitting these time dependent voltage sources in their Fourier modes, we get a large number of sources with different frequencies and at different locations. By solving the current and radiated electric field induced by each of these sources separately, and combining them afterwards, we can find the total resulting current and radiated electric field of the time-dependent system. The same results can also be used in different combinations, to get results for different modes of operation of the same device.

As a result, we will have to solve the EFIE (2.16) for many different frequencies and for each frequency for many different localised external electric fields.

2.2 Petrov-Galerkin approach

To solve the EFIE (2.16) we discretise it using a general Petrov-Galerkin approach, which is also called the method of moments in this context. First we write the EFIE equation in a weak form where we use the test functions $\mathbf{T}$, leading to

$$\begin{aligned} &-\frac{i}{\omega}\int\!\!\int_\Gamma \nabla\cdot\mathbf{T}(\mathbf{x})^*\; G(\mathbf{x},\mathbf{x}')\; \nabla'\cdot\mathbf{J}(\mathbf{x}')\; d^2\mathbf{x}'d^2\mathbf{x} \\ &+\frac{i\omega}{c^2}\int\!\!\int_\Gamma \mathbf{T}(\mathbf{x})^*\; G(\mathbf{x},\mathbf{x}')\; \mathbf{J}(\mathbf{x}')\; d^2\mathbf{x}'d^2\mathbf{x} \\ &+\int_\Gamma \mathbf{T}(\mathbf{x})^*\; Z(\mathbf{x})\; \mathbf{J}(\mathbf{x})\; d^2\mathbf{x} = \int_\Gamma \mathbf{T}(\mathbf{x})^*\; \mathbf{E}^E(\mathbf{x})\; d^2\mathbf{x} \qquad \forall_{\mathbf{T}\in\mathcal{T}(\Gamma)}\ , \end{aligned} \tag{2.17}$$

where we used integration by parts to see that

$$\int_\Gamma \mathbf{T}(\mathbf{x})^*\cdot\nabla_\Gamma G(\mathbf{x},\mathbf{x}')\; d^2\mathbf{x} = -\int_\Gamma \nabla\cdot\mathbf{T}(\mathbf{x})^*\; G(\mathbf{x},\mathbf{x}')\; d^2\mathbf{x}\ . \tag{2.18}$$

Rigorously constructing a correct test space $\mathcal{T}(\Gamma)$ is complicated for practical problems like these, and we will not attempt to do this.

Next we discretise by introducing a finite set of test functions $\mathbf{T}_i \in \mathcal{T}(\Gamma)$ and a finite basis for the current density, such that

$$\mathbf{J}(\mathbf{x}) = \sum_{j=1}^{n} x_j \mathbf{\Psi}_j(\mathbf{x})\ , \tag{2.19}$$

with $\mathbf{\Psi}_j$ the basis functions for the current. When we substitute this in the weak formulation (2.17) we get

$$\begin{aligned}
&-\frac{i}{\omega}\sum_{j=1}^{n}\iint_{\Gamma}\nabla\cdot\mathbf{T}_i(\mathbf{x})^*\;G(\mathbf{x},\mathbf{x}')\;\nabla'\cdot\mathbf{\Psi}_j(\mathbf{x}')\;d^2\mathbf{x}'d^2\mathbf{x}\\
&+\frac{i\omega}{c^2}\sum_{j=1}^{n}\iint_{\Gamma}\mathbf{T}_i(\mathbf{x})^*\;G(\mathbf{x},\mathbf{x}')\;\mathbf{\Psi}_j(\mathbf{x}')\;d^2\mathbf{x}'d^2\mathbf{x}\\
&+\sum_{j=1}^{n}\int_{\Gamma}\mathbf{T}_i(\mathbf{x})^*\;Z(\mathbf{x})\;\mathbf{\Psi}_j(\mathbf{x})\;d^2\mathbf{x}=\int_{\Gamma}\mathbf{T}_i(\mathbf{x})^*\;\mathbf{E}^E(\mathbf{x})\;d^2\mathbf{x}\qquad\forall_{i=1\ldots n}\;\;.
\end{aligned}\tag{2.20}$$

This is an ordinary set of linear algebraic equations which can be written as a matrix equation

$$Ax=b\tag{2.21}$$

where we used the following definitions

$$A=C+L+R\tag{2.22a}$$

$$C_{ij}=-\frac{i}{\omega}\iint_{\Gamma}\nabla_\Gamma\cdot\mathbf{T}_i(\mathbf{x})^*\;G(\mathbf{x},\mathbf{x}')\;\nabla'_\Gamma\cdot\mathbf{\Psi}_j(\mathbf{x}')\;d^2\mathbf{x}'d^2\mathbf{x}\tag{2.22b}$$

$$L_{ij}=\frac{i\omega}{c^2}\iint_{\Gamma}\mathbf{T}_i(\mathbf{x})^*\;G(\mathbf{x},\mathbf{x}')\;\mathbf{\Psi}_j(\mathbf{x}')\;d^2\mathbf{x}'d^2\mathbf{x}\tag{2.22c}$$

$$R_{ij}=\int_{\Gamma}\mathbf{T}_i(\mathbf{x})^*\;Z(\mathbf{x})\;\mathbf{\Psi}_j(\mathbf{x})\;d^2\mathbf{x}\tag{2.22d}$$

$$b_i=\int_{\Gamma}\mathbf{T}_i(\mathbf{x})^*\;\mathbf{E}^E(\mathbf{x})\;d^2\mathbf{x}\;\;.\tag{2.22e}$$

The capacitive effects are now represented by the matrix C, the inductive effects by the matrix L and the resistive effects by the matrix R. The external electric field is represented by the right-hand side b while the current $\mathbf{J}$ is represented by x.

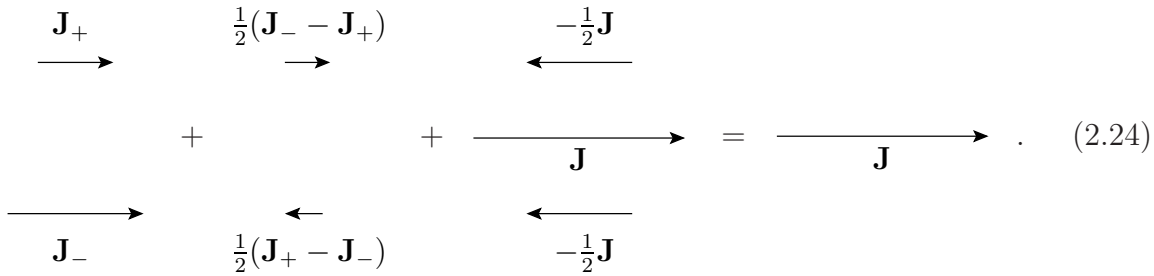
2.2.1 Thin boards and wires approximation

In the problems that we deal with, the conductors will be composed of thin wires and boards. The fact that they are thin allows us to make some simplifications.

For each board there is a front and back plane, both having their own current. Since both sides of the board are close together, we can combine these front and back currents into one current, located at the centre of the board. The error we introduce by doing this, will be negligible if the board is thin. Suppose the current at the front is $\mathbf{J}_+$, the current at the back is $\mathbf{J}_-$, and the thickness of the board is d. We combine both currents and get a current $\mathbf{J}=\mathbf{J}_++\mathbf{J}_-$ at the centre of the board. To justify this, we decompose the front and back currents as

$$\begin{aligned}
\mathbf{J}_+&=\frac{1}{2}(\mathbf{J}_++\mathbf{J}_-)+\frac{1}{2}(\mathbf{J}_+-\mathbf{J}_-)=\frac{1}{2}\mathbf{J}+\frac{1}{2}(\mathbf{J}_+-\mathbf{J}_-)\\
\mathbf{J}_-&=\frac{1}{2}(\mathbf{J}_++\mathbf{J}_-)-\frac{1}{2}(\mathbf{J}_+-\mathbf{J}_-)=\frac{1}{2}\mathbf{J}-\frac{1}{2}(\mathbf{J}_+-\mathbf{J}_-)\;\;,
\end{aligned}\tag{2.23}$$

which leads to the following graphical equation



(2.24)

This shows that the difference between the real separated currents (the first term) and the approximation (the right-hand side) consists of a dipole term of strength $d|\mathbf{J}_+ - \mathbf{J}_-|/2$ (second term) and a quadrupole term of strength $d^2|\mathbf{J}|/2$ (third term). This shows that the total error is of order of d, and will vanish as d tends to zero. In practice, the board thickness d will be much smaller than the discretisation grid size h. This means that the errors that are introduced here will be small compared to the discretisation errors.

This simplification reduces the computational domain Γ for the boards by a factor of two. Since relatively, the boards contain the largest number of discretised degrees of freedom, this results in a large reduction of computational work.

For the thin wires, we can make similar simplifications. We will only allow currents in the direction along the wire, and we will assume that the current density is constant around the wire. This makes the wire essentially one dimensional. By ignoring currents in the plane perpendicular to the wire, and ignoring variation of the current around the wire, we again make a dipole field error, that is now proportional to the wire radius. This radius is in general also small compared to discretisation grid size, which means that these errors are insignificant. Ideally, we would have liked to replace the thin wire by a conducting line at the centre of the wire, in the same way as we replaced the two sided board by a single plane at the centre of that board. Unfortunately, this is not possible, since the resulting line current would introduce infinite field strengths on that line. We thus have to account for the thickness of the wire, even if it is very thin.

2.3 Wire model

It is well known in physics, that a fine mesh of conducting wires shields radiation almost as well as a conducting plate, as long a the holes in the wire grid are much smaller than the wavelength of the radiation. A very common application of this effect can be found in the doors of microwave ovens. The fine metal mesh in the door shields the microwave radiation while you can still look inside.

We can use this principle and replace all the thin conducting plates by wire meshes. The advantage of this substitution is that the resulting model consists only of wires, which will reduce the complexity of the code since it has to deal with only one type of element. However, in section 2.3.2 we will see that there are also some disadvantages to this substitution, resulting in the fact that we are not using this substitution. But, even when we are not using the wire grid approximation for surfaces, the wire discretisation described below can still be used for the true wires in the model.

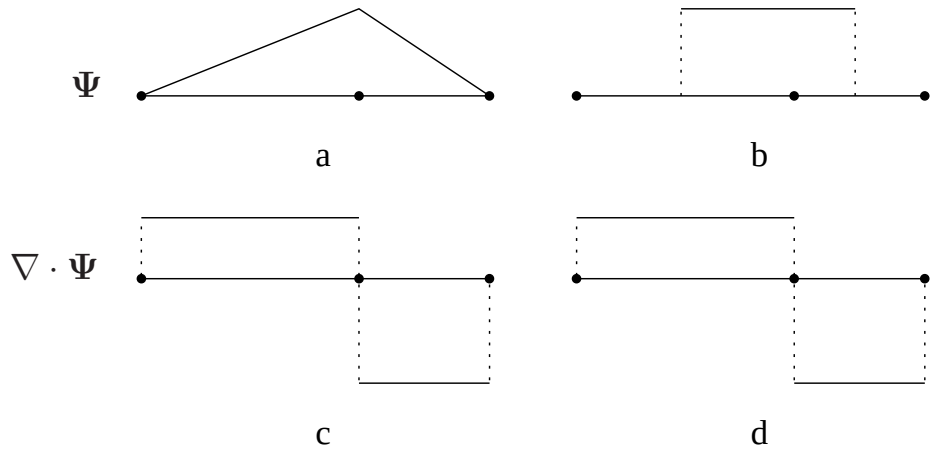


FIGURE 2.1: Current and charge basis functions on a pulse. Left the original, right the simplified version.

This model, and its discretisation as described in this section, were implemented in the Berber code by Bergervoet [10, 33]. In that sense, it was the starting point for this project.

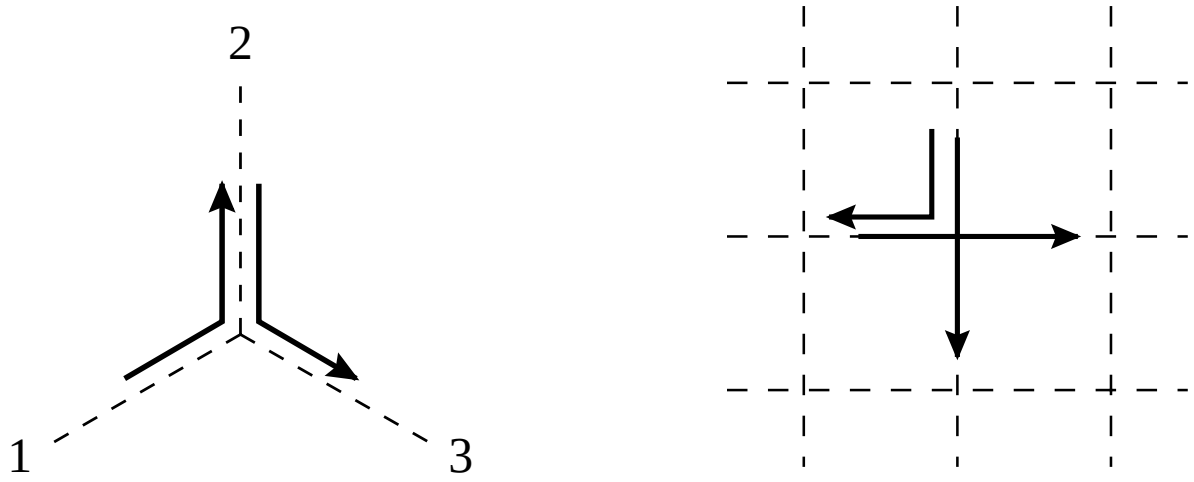
2.3.1 The choice of basis and test functions

Under the model restrictions described in section 2.2.1, the wires are essentially one dimensional, in the sense that the current on the wire surface can only change in one direction (along the wire) and the direction of the current is fixed along the wire. We can thus represent the current by a scalar function on a 1-dimensional domain, and apply standard discretisation methods for 1-dimensional scalar functions to the wire. We split the wire in segments, and define the basis functions for the current as the often used continuous and piecewise linear functions that satisfy

$$\mathbf{\Psi}_i(\mathbf{x}_j) = \begin{cases} 1 & \text{for} \quad j = i \\ 0 & \text{for} \quad j \neq i \end{cases} , \tag{2.25}$$

where the $\mathbf{x}_j$'s are the positions of the element boundaries. An example of such a $\mathbf{\Psi}_i$ is shown in Figure 2.1a. We will call such a basis function a pulse, describing a current from one element to the next. In Figure 2.1c, we show the corresponding divergence $\nabla \cdot \mathbf{\Psi}_i$, which we call a charge basis function since it is so closely related to the charge by the continuity equation (2.6). These charge basis functions will be necessary to compute the capacitive matrix C defined in equation (2.22b). A natural choice for the test functions is to make them equal to the basis functions, $\mathbf{T}_i = \mathbf{\Psi}_i$, so that the matrix A will be symmetric, as can be seen from equations (2.22).

So far, this describes a discretisation of one wire, but it does not cover multiple connected wires. In a point where three wires are joined together, we need to be able to describe a current between each pair of these wires. To achieve this, we need two extra pulses, one connecting the first and second wire, and one connecting the second and third wire. The first and third wire are now connected via the second wire, as can be seen in Figure 2.2(a).



(a) Three connected wires.

(b) Typical situation in a wire grid.

FIGURE 2.2: Extra current basis functions or pulses (arrows) needed for connected wires (dashed lines).

Now that we have chosen our basis functions $\mathbf{\Psi}_i$ and test functions $\mathbf{T}_i$, we will try to compute the matrix A according to definition (2.22). For the computation of the matrix element A_{ij}, $\mathbf{x}$ must be integrated over the surface of the wire elements of $\mathbf{T}_i$ and $\mathbf{x}'$ must be integrated over the surface of the wire elements of $\mathbf{\Psi}_j$, making the integral 4-dimensional. These integrals cannot be computed analytically. Since we have to compute so many of these integrals, accurate numerical integration turns out to be much too expensive. Therefore, we have to find an alternative.

In order to simplify the integrals to a level where we can approximate them with high accuracy, we use simplified versions of our basis and test functions. While doing this, we simplify the current and charge functions independently, after which the charge functions do not equal the divergence of the current functions any more. However, they both approximate the original functions of Figures 2.1a and 2.1c. The piecewise linear function for the current is replaced by a piecewise constant function as shown in Figure 2.1b, that preserves the total current (surface of the graph) for each element. The charge basis function is left unchanged. The test functions are simplified more drastically, as is shown in Figures 2.3b and 2.3d, where the vertical lines represent delta functions. Again we preserved the total amount of current and charge per element.

The reduction of the test functions to combinations of delta functions reduces the integrals (2.22) to 3-dimensional integrals. This is further reduced by neglecting the variation of the field around the wire of the test function, and using only one test point on the surface of this wire. This reduced the integrals in (2.22) to the evaluation of the integral over $\mathbf{x}'$ for one test point $\mathbf{x}$. This 2-dimensional integral can still not be evaluated analytically, but an accurate analytical approximation is available [10]. This approximation to the kernel has the correct short and long distance limits and shows only a small deviation at distances in the order of a few times the wire radius. These approximations allow us to compute the matrix elements A_{ij} with much lower compu-

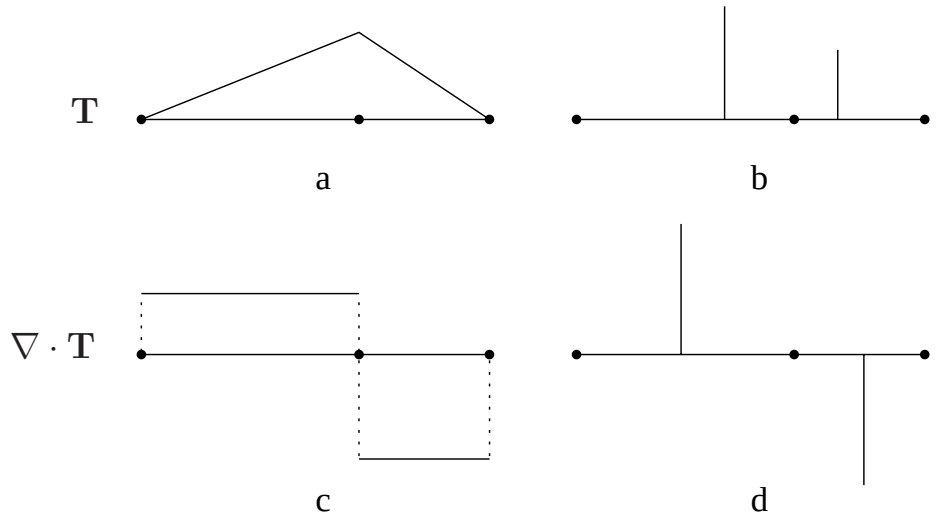


FIGURE 2.3: Current and charge test functions on a pulse. Left the original, right the simplified version.

tational cost than more conventional quadrature methods would require to approximate the correct short distance behaviour. However, because of the large number of elements in the dense matrix (n^2), computing the whole matrix A is still expensive.

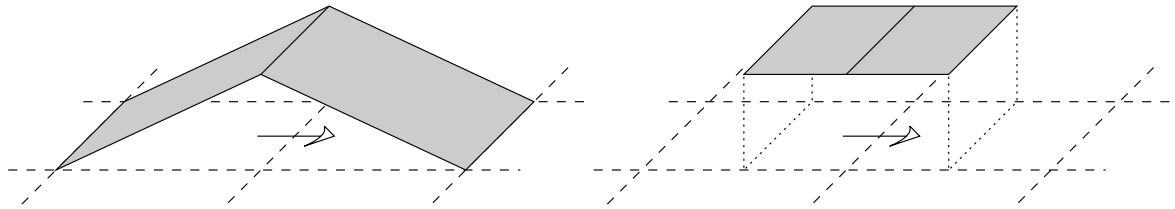
2.3.2 Drawbacks of the wire model

One problem with the wire model, is that it does not tell us which wire radius R we should use. This wire radius has the most influence on the short range interactions, especially the self interaction (the diagonal elements of A). Usually, some value in the order of $R = 0.2h$ is chosen. In principle, all values of R proportional to h should lead to correct results in the limit as $h \to 0$, but the speed of convergence will vary. This problem is not insurmountable, but there is a larger problem.

The wire model for a board will consist of a fine square grid of wires, where each square is only one segment wide. At each wire crossing, four segments must be connected, requiring three pulses, as shown in Figure 2.2(b). If we disregard the boundaries of the board, this results in three degrees of freedom per point of the grid, one horizontal current, one vertical current and one current that goes around the corner. This goes against the intuition for a real current on a board, which has only two physical degrees of freedom per point: the horizontal and vertical current. The extra pulse in the wire grid is needed to connect the horizontal and vertical wires. In the continuous case, this coupling between the horizontal currents J_x and vertical currents J_y is found in the continuity equation (2.6)

$$\frac{\partial \mathrm{J}_x}{\partial x} + \frac{\partial \mathrm{J}_y}{\partial y} = -i\omega\rho \ . \tag{2.26}$$

Both currents share the same charge density ρ, so, loosely speaking, the same charge can be part of both J_x and J_y. For the wire model described here, this is not the case. Charge is located at the horizontal or a vertical segment and to get from one to the other, the "around the corner" current is needed.



(a) Piecewise linear current basis function. (b) The piecewise constant simplification.

FIGURE 2.4: A 3-dimensional representation of the current basis function and its simplification on a surface pulse. The grid is show with dashed lines, the current density in the left-to-right direction is plotted in the vertical direction and is given by the shaded surface.

In principle, this is not a problem. The extra degree of freedom goes against intuition, but the wire model is not incorrect, and useful results have been obtained with it. However, we would like to solve the resulting linear system with an iterative solver as described in section 1.2. To get an efficient iterative solver, we will need an efficient preconditioner for the matrix A. This is where the problem arises. Due to the counter intuitive discretisation, we were initially unable to construct a good preconditioner. For this reason, we concentrated on a discretisation that does not replace the boards with a wire grid, as is described in the next section. Most of the remainder of this thesis will be devoted to finding a preconditioner for the linear systems that arise from this type of discretisation. In hindsight, we could also apply the techniques we developed, with some minor modifications, for the wire discretisation. However, we prefer the intuitively more direct surface model.

2.4 Surfaces model

In the surface model, the conducting boards are not replaced by a wire grid, but treated as a real surface. We do use the thin board approximation described in section 2.2.1, and hence discretise a single current conducting surface. For the wires in the model, we can still use the discretisation described in section 2.3.

The surface is divided in small surface elements, which are then connected by basis functions for the current, which we will still call pulses. Such a pulse will transport charge between two adjacent elements, and each edge between a pair of neighbouring elements will have one such pulse.

We will be using a regular rectangular grid, for which the natural analogue for the piecewise linear wire basis function that satisfies (2.25) is shown in Figure 2.4(a). The corresponding divergence is constant on both elements and we adopt a normalisation such that the total divergence per element is one, i.e. if Γ_i denotes the element number i,

$$\int_{\Gamma_i} \nabla \cdot \mathbf{\Psi}_i = \begin{cases} \pm 1 & \text{if } i \text{ is an edge of element } j, \\ 0 & \text{remaining elements.} \end{cases} \tag{2.27}$$

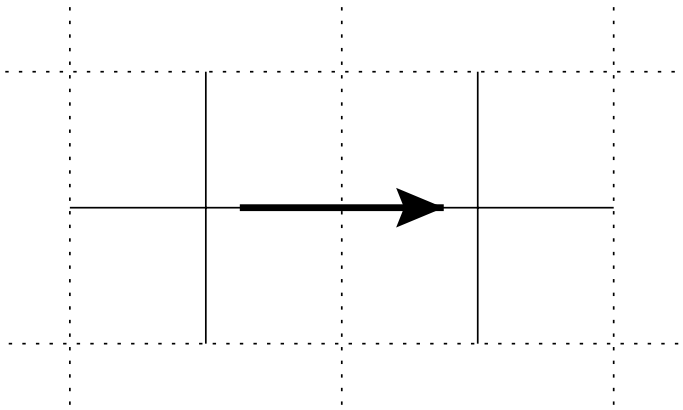


FIGURE 2.5: Wire approximation to surface basis function. The dotted lines show part of the finite element grid, the solid lines the replacement wire segments, and the arrow the wire current pulse used as current basis function.

Using this regular rectangular grid, we only need horizontal and vertical current pulses. These horizontal and vertical currents generate charges on the same elements, and are in this way connected. The "around-the-corner" pulses that made the wire grid basis counter intuitive (see section 2.3.2) are not needed here.

Choosing for both the basis functions $\mathbf{\Psi}_i$ and the test functions $\mathbf{T}_i$ the piecewise linear (but not continuous) functions of Figure 2.4(a) and equation (2.27), results in 4-dimensional integrals in the definition (2.22) of the matrix A, that cannot be evaluated analytically. Just as for the wire model of section 2.3.1, we will approximate these integrals by constructing simplified versions of the basis and test functions for which we can construct analytical expressions for the integrals (2.22).

The first simplification step is to replace the piecewise linear basis and test functions for the current by a piecewise constant function, shown in Figure 2.4(b). This reduces the integration problems to the problem of evaluating the integral

$$I(R, R') = \int_R \int_{R'} G(\mathbf{x}, \mathbf{x}') d^2x d^2x' \quad , \tag{2.28}$$

where R and R' are two rectangular surface elements. This integral still cannot be evaluated analytically, so we will discuss two ways to simplify it even further to a point where we can approximate the elements of A analytically.

2.4.1 The wire approximation to surfaces

The idea of this variant is to use the integration techniques we used for the wire model, as discussed in section 2.3, to approximate the interaction between two surface elements. To do this, we will represent a rectangular surface element by two perpendicular wire segments on the centre lines of the rectangle. The current basis functions $\mathbf{\Psi}_i$ can be approximated by the current pulse on the two wires that lie in the direction of the current, as shown in Figure 2.5. To approximate the charge basis function $\nabla \cdot \mathbf{\Psi}_i$, the charge that is accumulated by the current $\mathbf{\Psi}_i$ is spread over the two perpendicular

wire segments of the corresponding elements. This way, charge is still shared by the horizontal and vertical currents, and no "around-the-corner" currents are needed. For the approximation of the test functions $\mathbf{T}_i$, we choose the corresponding simplified wire test functions, such that we can express the interaction between surface elements as a combination of wire element interactions. These wire interactions can be evaluated using the kernel approximation that was used for the wire model as described at the end of section 2.3.1.

In general, there will be connections between wires and boards in the model. In this scenario, such a connection can be easily achieved by connecting the end of the real wire to the end of a wire representing a surface element, and adding a connecting pulse to the basis and test functions.

The last thing to do, is to decide what radius should be used for the wire segments that represent the surface. We have used a form of the self interaction of an element R to fix the wire radius. We have chosen this radius such, that the wire approximation to the integral $I(R, R)$ gives the (2-dimensional) integral over R evaluated at the centre of this element in the case of square elements R and in the low frequency limit ($k = 0$). Using numerical approximation, we found the radius $0.179844h$. The influence of the radius on the interaction approximation decreases very rapidly with increasing distance. The difference between the electrostatic nearest neighbour interaction using wires and using the full integral is already down to a few percent.

Edge wire

It is well known that the solution of the Laplace equation on a 3-dimensional domain with a plane cut in it will be singular at the end of the cut. In the same way the electrostatic problem of a thin conducting half-plane at a given potential will result in a diverging charge density toward the edge of the conductor [26]. In our electrodynamic problem, there will also be a $1/\sqrt{x}$ -type divergence for the charge and current density toward the end of a conducting plane. It is therefore important to have a closer look at the edge of the conducting plane. In the wires model of section 2.3, we have a wire at the edge of the plane so that a large charge or current can be situated at the edge of the plane. However, in our surfaces model the charge and current at the edge of the plane is spread out over the whole edge element. Therefore, it might be necessary to take very small elements at the edge, using some kind of grid refinement. Another possible way to improve this situation, is to add an extra wire at the edge of a surface so that, like in the wires model, there can be a macroscopic amount of charge and current at the edge of the board. The addition of this edge wire could thus improve the accuracy of our final results.

For this edge wire, we also need to decide what radius to use. We will use the interaction between an edge wire element W and its neighbouring element R on the surface to fix the edge wire radius. We chose the radius, such that the wire approximation to $I(R, W)$, gives the (2-dimensional) integral over R evaluated at the centre of the wire element W, in the case of a square element R and again in the low frequency limit ($k = 0$). This leads to an edge wire radius of $0.0946802h$.

We will call this discretisation variant using the extra edge wire the "wire approximation to surfaces with edge wire".

2.4.2 The point approximation to surfaces

The wire approximation to surfaces as described above is quite elaborate and it is unclear whether this results in any additional accuracy in comparison to much simpler approximations. Furthermore, it does not allow an easy combination with the Fast Multipole Method since this requires the knowledge of the multipole moments of a wire segment (see section 2.7).

A very simple alternative is based on a low order Taylor expansion of the Green function G (2.15) in the integral I (2.28),

$$\begin{aligned} G(\mathbf{x},\mathbf{x}') = G(\mathbf{x}_0,\mathbf{x}_0') &+ (\mathbf{x}-\mathbf{x}_0)\cdot\nabla_x G(\mathbf{x}_0,\mathbf{x}_0') \\ &+ (\mathbf{x}'-\mathbf{x}_0')\cdot\nabla_{x'} G(\mathbf{x}_0,\mathbf{x}_0') + \mathcal{O}(|\mathbf{x}-\mathbf{x}_0|^2 + |\mathbf{x}'-\mathbf{x}_0'|^2) \quad . \end{aligned} \tag{2.29}$$

By choosing $\mathbf{x}_0$ and $\mathbf{x}_0'$ the centre points of the rectangles R and R', and using this expression to integrate $I(R,R')$, the linear terms will give a zero result, leading to

$$I(R,R') = \int_R \int_{R'} G(\mathbf{x},\mathbf{x}') d^2x d^2x' = |R|\,|R'|\,G(\mathbf{x}_0,\mathbf{x}_0') + \mathcal{O}\left(\frac{h^6}{d(R,R')^3}\right) \quad , \tag{2.30}$$

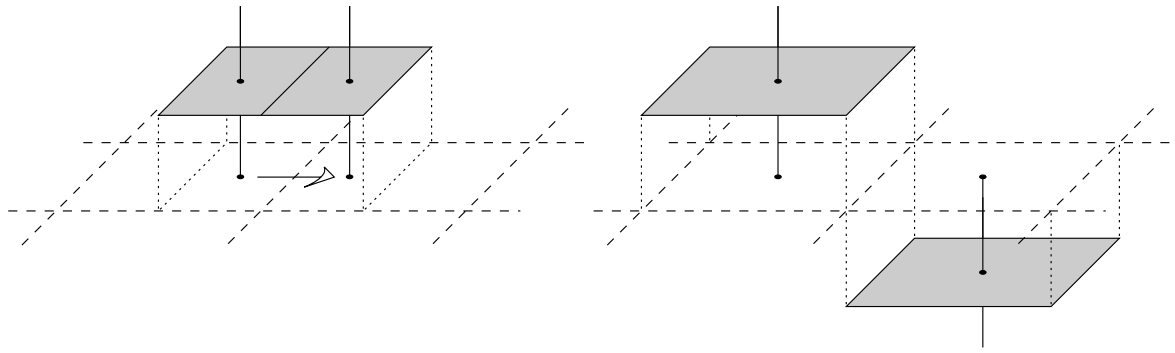
where $|R|$ and $|R'|$ are surface areas of R and R', h is the maximum size of R and R', and $d(R,R')$ is the shortest distance between R and R'. This error term does not follow directly from the error term in (2.29), but requires some more manipulations and the assumption that the length scale of our model is not much larger than the wavelength, which implies that $k|\mathbf{x}-\mathbf{x}'| = \mathcal{O}(1)$.

If we use this approximation when computing the elements of the matrix A, this can also be interpreted as using delta functions for the basis and test functions at the locations corresponding to $\mathbf{x}_0$ and $\mathbf{x}_0'$. This is very similar to the simplification of the test functions for the wire as shown in Figures 2.3b and 2.3d. Figure 2.6 shows these point-based current and charge basis functions. Note that the basis and test functions are simplified in the same way, so that they remain equal, $\mathbf{T}_i = \mathbf{\Psi}_i$.

This approximation has several advantages. The integrations reduce to a single evaluation of the kernel, which is computationally much cheaper than the wire-wire interactions used in the wires approximation. Furthermore, since the basis and test functions are the same and real valued, the matrix A is symmetric (see also section (3.1)), saving even more on the computation of A. However, using these basis and test functions has one practical problem. The approximation to the self interaction integral

$$I(R,R) = \int_R \int_R G(\mathbf{x},\mathbf{x}') d^2x d^2x' \tag{2.31}$$

becomes infinite since the Green function G is singular for zero distance. We can resolve this by using the true self interaction $I(R,R)$ when it is needed. This value can be computed numerically, but this is time consuming, so we will try to compute this value in advance. In principle, the Green function depends on the frequency (equation (2.15)), but for the small distances of the self interaction, $|\mathbf{x}-\mathbf{x}'|$ is much smaller than k and which makes the static Green function (G with $k=0$) a good approximation. Using $k=0$, $I(R,R')$ only depends on the size of R, and by scaling all lengths in the integral, the result depends only on the ratio of the lengths of the sides of the rectangle R. We can



(a) Current basis function. (b) Charge basis function.

FIGURE 2.6: Point approximation to surface basis functions. The dashed lines show part of the finite element grid, the vertical lines mark the location of the delta functions and the shaded surfaces shows the distributions for which the delta functions are an approximation.

compute this integral in advance for a wide range of ratios using numerical integration. To compute the self interaction, we can now use linear interpolation to retrieve the correct value for the integral $I(R, R)$ from a table of values computed in advance.

We will also use the delta function simplification for the wires. Both the basis and test functions on the wire elements are replaced by delta functions at the wire centre, in comparable positions as for the surface elements. We also precomputed the exact wire self interaction integral I (2.31) for $k = 0$ and a range of length to radius ratios, allowing us to compute the wire self interaction I using linear interpolation.

Exact nearest neighbour interaction

The difference between the exact integral I (2.28) and the point approximation (2.30) will be the largest for small distances, as can be seen from the error term in (2.30). In an attempt to get better final results, we might also use exact nearest neighbour interactions $I(R, R')$, for rectangular elements R and R' that share one side. To this purpose, we computed this integral numerically for $k = 0$ and a range of length to width ratios, assuming that the two elements have the same size. Using linear interpolation on these precomputed values we can use (nearly) exact nearest neighbour integrals $I(R, R')$. We will call this variant the "point approximation to surfaces with exact nearest neighbour interaction".

2.5 Convergence results

In order to compare the different integration methods for the surface model described in sections 2.4.1 and 2.4.2, we will compare the results for a test problem using the four different methods :

- the wire approximation to surfaces,

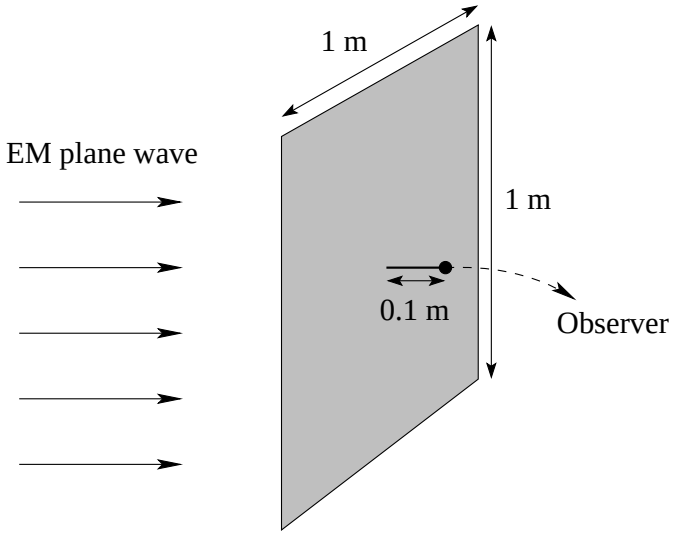


FIGURE 2.7: The test case geometry.

- the wire approximation to surfaces with edge wire,
- the point approximation to surfaces, and
- the point approximation to surfaces with exact nearest neighbour interaction.

Since, in the end, we are interested in the radiated electric fields, we will not compare the induced current computed using the different variants, but only compare the accuracy of the computed values for the resulting electric field.

In order to be able to verify the results of these methods, we have to test them on a problem for which we have an exact or very accurate solution, which is a severe restriction on the possible choices of test problems.

2.5.1 The test problem

The test problem we used is shown in Figure 2.7. We used a single 1×1 meter board lying in the xz-pane. The external field is a plane wave travelling in the positive y direction with polarisation in the x direction :

$$\mathbf{E}^E(\mathbf{x}) = \hat{\mathbf{x}} e^{-ik\mathrm{x}_y} \quad , \tag{2.32}$$

where $\hat{\mathbf{x}}$ is the unit vector in the x-direction and x_y the y-component of $\mathbf{x}$. We measure the electric field at a distance of 0.1 meter behind the centre of the board (positive y direction). Using symmetry arguments, it can be seen that the electric field in the test point will also have a component in the x-direction only.

In order to be able to measure the accuracy of the different methods, we need to know the true results with very high accuracy. As far as we know, even for a geometry as simple as this, the problem cannot be solved analytically. However, in this special case where we have one flat conducting surface, the Fast Fourier Transform (FFT) can be used to compute matrix-vector multiplications with A, without explicitly computing and storing the matrix. In combination with an iterative solver, this allows the use of very fine discretisations. Assuming that this will converge to the correct result, this

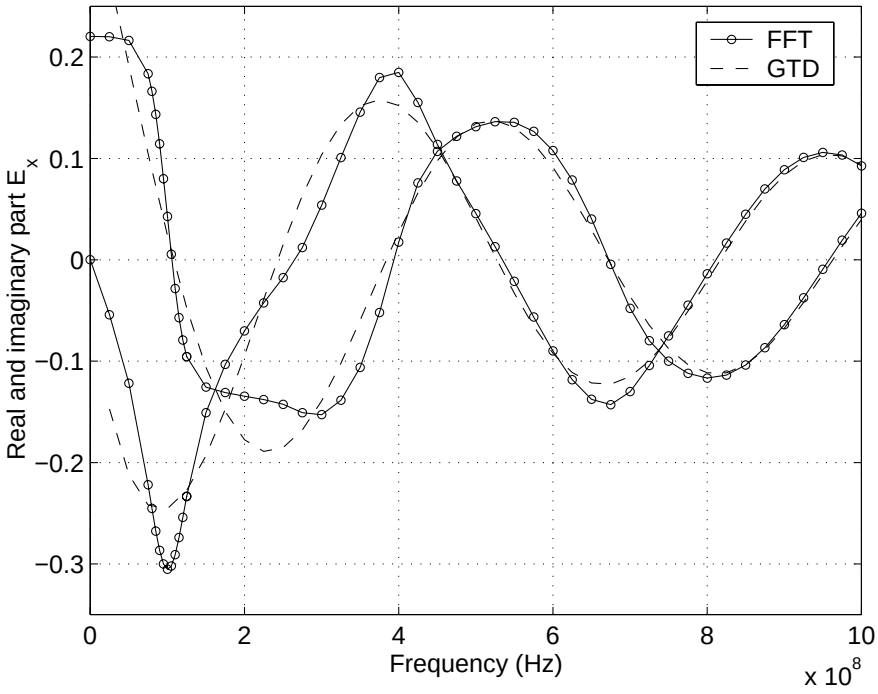


FIGURE 2.8: The real and imaginary parts of E_x in the test point, using the same units as for $\mathbf{E}^E$ in equation (2.32). The imaginary part is zero for frequency zero. Both the FFT results using Richardson extrapolation on the results for $h = 1/64$ and $h = 1/128$ and the analytic approximation (GTD) are shown.

can give high accuracy results. Application of Richardson extrapolation to the results for two different grid sizes h, extrapolating these values to $h = 0$, can give even more accurate results. This was done by Bergervoet [9], who used Richardson extrapolation on the results for $h = 1/64$ and $h = 1/128$ to get reasonably accurate values for E_x in the test point over a frequency range of 0 to 1 GHz. Comparison with more recent results using Richardson extrapolation on the results for $h = 1/256$ and $h = 1/1024$, he got error estimates for the old results of less than 0.3% for frequencies below 500 MHz and a maximum of 0.8% over the whole frequency range.

In order to attempt to check the validity of this method, these results were compared with results from a totally independent analytic approximation method that uses the geometrical theory of diffraction (GTD) [4, 3]. It is based on ray diffraction at the edge of the board. This method is only exact in the high frequency limit, and showed a reasonable correspondence to the FFT results using Richardson extrapolation on the results for $h = 1/64$ and $h = 1/128$, as shown in Figure 2.8.

Since the older Richardson extrapolation results are expected to be accurate enough for our purposes, we will still use these older results, computed by Bergervoet using Richardson extrapolation on the results for $h = 1/64$ and $h = 1/128$, as "true" values to compare the different discretisations. These "true" values are shown in Figure 2.8.

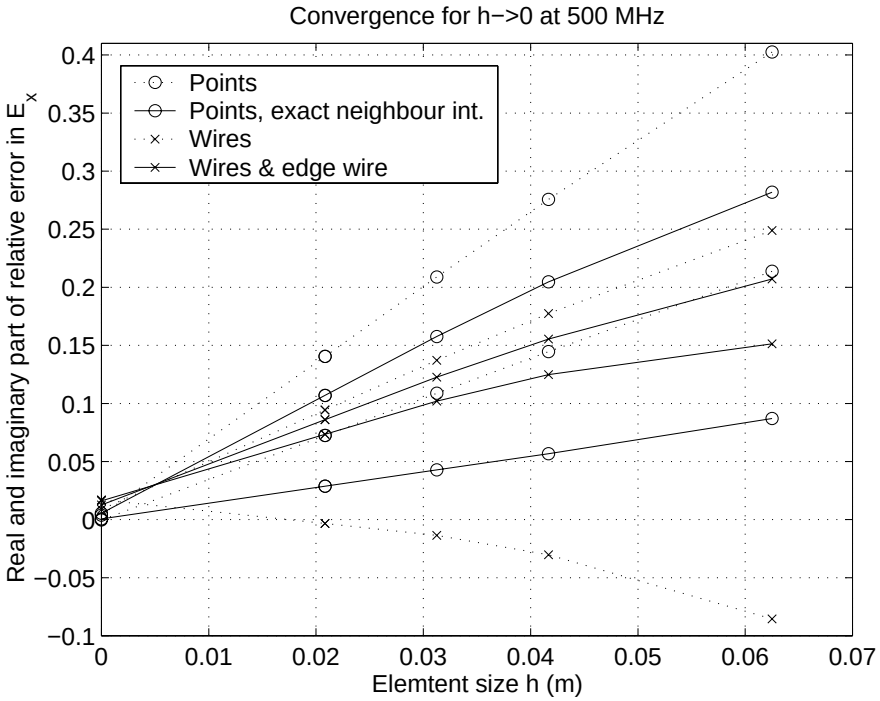


FIGURE 2.9: Errors in the real and imaginary part of the computed electric field for 500 MHz, divided by the norm of the correct value. At $h = 0$ the results of linear Richardson extrapolation on the $h = 1/32$ and $h = 1/48$ values is shown.

2.5.2 Results

We computed the electric field in the test point for the frequency range of 50 MHz to 1 GHz with 50 MHz intervals. This was done using several grid sizes ($h = 1/16$, $h = 1/24$, $h = 1/32$, and $h = 1/48$) and the four different integration methods.

In Figure 2.9, the results for a frequency of 500 MHz and for several grid sizes are shown. For all methods there is a clear tendency of the error to depend linearly on the grid size. This can be used to do linear extrapolation of the field strength values to $h = 0$. The results of this Richardson extrapolation on the $h = 1/32$ and $h = 1/48$ results is plotted at $h = 0$. We have observed this linear convergence behaviour for all frequencies. However, for high frequencies (like 1 GHz) the linear behaviour only appears for the smaller grid sizes.

Figure 2.10 shows the relative error for $h = 1/32$ as a function of the frequency. If we compare the two variants of the point approximation, we see that using the exact integrals for the nearest neighbours can lead to a nice reduction of the error, in this case for the 400 to 800 MHz region. Using the exact nearest neighbours interactions does not necessarily help, but in this example, it never hurts. The comparison between the two wire variants is less conclusive. For different frequency regions, the one or the other is better. The variant with the edge wires shows the most regular behaviour and is best for the high frequencies, which gives it a little edge over the simple wire variant. If we compare the wire variants to the point variants, there is no obvious "winner".

We have also applied Richardson extrapolation on the $h = 1/24$ and $h = 1/32$ values,

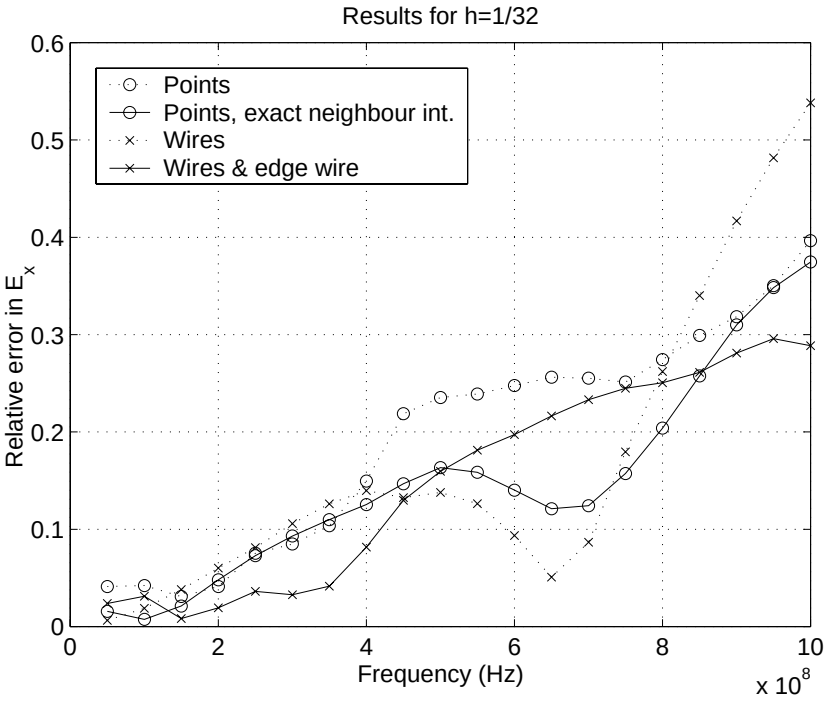


FIGURE 2.10: Relative errors in the computed electric field strength for $h = 1/32$.

for which the results are shown in Figure 2.11. From the figure, we can see that the point results extrapolate better than the wire results, and that the extrapolation does worse for the higher frequencies. For this case, the simple point approximation clearly does best. Although the method does not give the best direct results, the simplicity of the method apparently leads to a very constant linear convergence, as can also be seen in Figure 2.9, which leads to good extrapolation results.

2.6 Choosing an integration variant

The convergence results of the previous section do not show one obviously preferred method. Both the wires with edge wire and the points with exact nearest neighbour interaction show good results, but when using Richardson extrapolation, the simple points variant would seem best.

However, there are some more considerations to be made when choosing the integration variant. One is that, because of its simplicity, the points methods are more suitable to combine with Fast Multipole Methods (see section 2.7). A further nice effect of the points method is that the matrix A will be symmetric because the basis and test functions are the same. Another important argument is related to the computational cost of computing the matrix A. Again due to the simplicity of the points methods, but also due to the symmetry of A, these variants are much faster in computing A, as can be seen in Table 2.1.

In the remainder of this thesis, we have used the point approximation with exact nearest neighbour interaction to compute the matrix A. Since the approximation of the

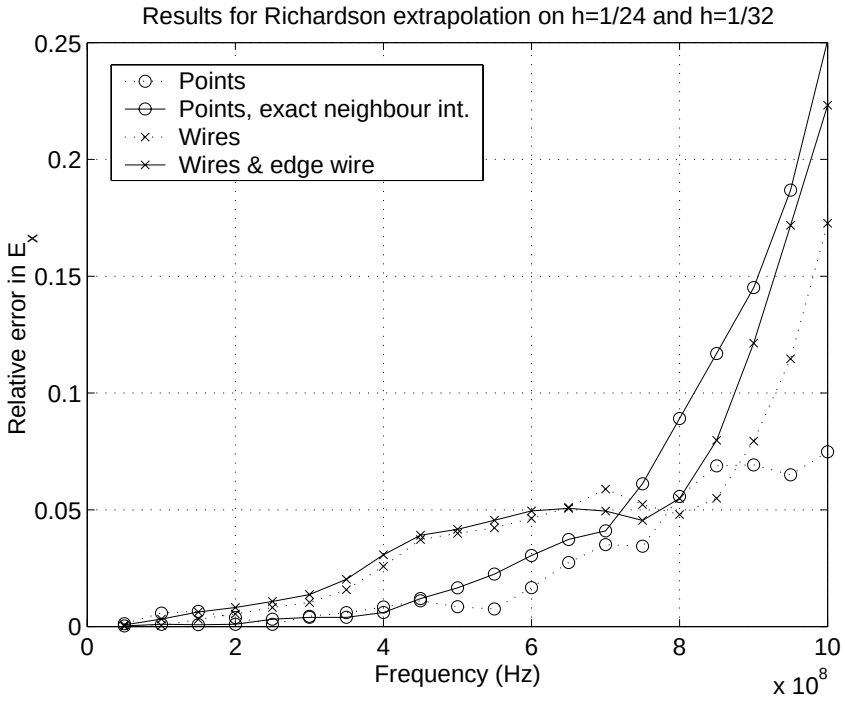


FIGURE 2.11: Relative errors when using the linear Richardson extrapolation on the $h = 1/24$ and the $h = 1/32$ results.

integration variant	seconds
wires	930
wires with edge wire	1091
points	229
points with exact nearest neighbour interaction	228

TABLE 2.1: Average CPU-time needed to compute the matrix A for a square board with 2304 surface elements and 4512 degrees of freedom (4896 for wires with edge wire) on a Sun Ultra 10 workstation.

integrals for this variant is symmetric (the same approximations for $\mathbf{T}$ and $\mathbf{\Psi}$), we are using an approximate Galerkin discretisation and we can write equations (2.22) as

$$A = C + L + R \tag{2.33a}$$

$$C_{ij} = -\frac{i}{\omega} \int\!\!\int_{\Gamma} \nabla_\Gamma \cdot \mathbf{\Psi}_i(\mathbf{x})^* \; G(\mathbf{x}, \mathbf{x}') \; \nabla'_\Gamma \cdot \mathbf{\Psi}_j(\mathbf{x}') \; d^2\mathbf{x}' d^2\mathbf{x} \tag{2.33b}$$

$$L_{ij} = \frac{i\omega}{c^2} \int\!\!\int_{\Gamma} \mathbf{\Psi}_i(\mathbf{x})^* \; G(\mathbf{x}, \mathbf{x}') \; \mathbf{\Psi}_j(\mathbf{x}') \; d^2\mathbf{x}' d^2\mathbf{x} \tag{2.33c}$$

$$R_{ij} = \int_{\Gamma} \mathbf{\Psi}_i(\mathbf{x})^* \; Z(\mathbf{x}) \; \mathbf{\Psi}_j(\mathbf{x}) \; d^2\mathbf{x} \tag{2.33d}$$

$$b_i = \int_{\Gamma} \mathbf{\Psi}_i(\mathbf{x})^* \; \mathbf{E}^E(\mathbf{x}) \; d^2\mathbf{x} \;\; . \tag{2.33e}$$

We expect that most results in the next chapters will also hold for the other variants, or at least show the same behaviour. Only the wires with edge wire variant might show some deviating behaviour due to the extra degrees of freedom.

2.7 Fast multipole method

The matrix A we defined in equation (2.33), is a dense matrix. If we have n degrees of freedom, this matrix will thus have n^2 elements. The number of degrees of freedom for the 2-dimensional boards is $\mathcal{O}(h^{-2})$, leading to $\mathcal{O}(h^{-4})$ elements for the matrix A. This number increases very rapidly with decreasing grid size. Decreasing h with a factor of 2 leads to an increase of the number of matrix elements with a factor of approximately 16.

The computational cost of calculating all these matrix elements explicitly can become very high. The amount of memory required to be able to store the matrix A in memory is $16n^2$, using double precision complex numbers. This can also become very large, and can thus form a severe limitation for the number of degrees of freedom that can be used. Furthermore, when using iterative solution methods, as described in section 1.2, we have to multiply the matrix A with a vector, which will also cost $\mathcal{O}(n^2)$ operations, making the iterations expensive.

In order to reduce the memory requirements and the computational cost of a matrix-vector multiplication, so-called "fast" methods were invented. We will describe the fast multipole method (FMM) and its predecessor, the Barnes-Hut method.

The general idea is to use the fact that the electric field induced by a group of currents (and corresponding charges) can be approximated by a multipole expansion. This approximation is already very effective if the distance to the group of currents is more than a few times the size of the region in which the currents are located.

This idea leads to the Barnes-Hut method [5]. When computing the electric field, the contribution from the nearby currents is computed directly. The contribution of currents at larger distances is computed in small clusters. The contribution of even further removed currents can be combined in larger clusters, etc. This leads to a method using a hierarchical subdivision of space. Usually the domain is divided in cubes. Groups of 8 cubes form a larger cube on the next coarser level. In order to compute the electric field in the test points, first the multipole expansions for each of the cubes in the different levels must be computed. This is done by starting with the smallest cubes. Once these expansions have been computed, they can be combined to form the expansions of the larger cubes of the coarser levels. Next, the electric field in each test point can be computed using the multipole expansions. A commonly used criterion is that a multipole expansion cannot be used for the nearest and next nearest neighbour cubes. For further removed cubes the expansion can be used. The large cubes can thus be used for far away interactions and smaller cubes can be used for closer interactions. The electric field due to the currents that are so close that they cannot be approximated using the multipole expansions on the level of the smallest cubes, is computed directly. For an illustration, see Figure 2.12. The number of contributing terms per level is bound by a constant number, leading to a total computational cost of $\mathcal{O}(n \log n)$.

This method gives a much cheaper but relatively accurate approximation of a matrix-vector multiplication with A. Furthermore, the matrix A does not have to be computed

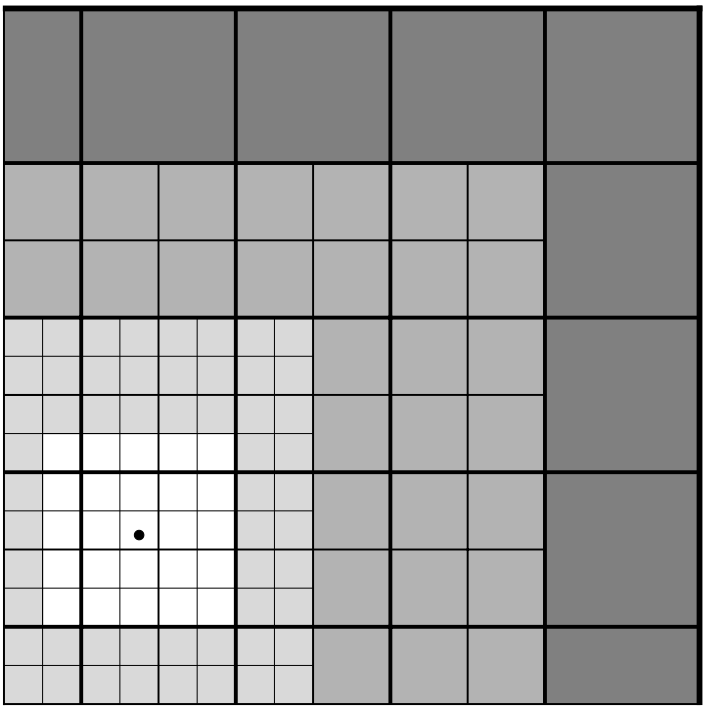


FIGURE 2.12: Illustration of a 2-dimensional variant of the Barnes-Hut method. The field at the location of the dot is computed. The field from the currents in the white region are computed directly, the field from the currents in the light shaded region using the multipole expansions of the small squares, the middle shaded region using the second level of squares, and the dark region using the next larger level of squares.

or stored. The storage requirements for the Barnes-Hut method are only $\mathcal{O}(n)$.

In order to reduce the computational work, Greengard and Rokhlin devised the fast multipole method (FMM) [19], which is an adaptation of the Barnes-Hut method described above. In the FMM, the per test point evaluation of the multipole moments is replaced by a hierarchical evaluation scheme that uses a kind of group-group interaction. The electric field evaluation works quite the same as the hierarchical computation of the multipole moments, but now the other way around. One starts at the coarsest level with the largest clusters. For each cube a local expansion of the electric field, induced by the multipole moments of the cubes that are more than 2 cubes away, is computed. In contrast with the multipole expansions, this expansion is valid *inside* the corresponding cube. On each finer level, a local expansion for each cube is computed using the coarser level local expansions and adding the induced field from the cubes of this level that are far enough away but not already included on coarser levels. For each cube at the finest level, this results in a local expansion of the electric field due to the currents in the cubes that are more than 2 cubes away. The electric field due to the close-by currents that are not included in these local expansions has to be computed explicitly. The total FMM scheme can be illustrated by the diagram in Figure 2.13. This double hierarchical scheme using the idea of group-group interactions reduces the computational cost to $\mathcal{O}(n)$.

For a given geometry, all the dependencies in Figure 2.13, as indicated by the ar-

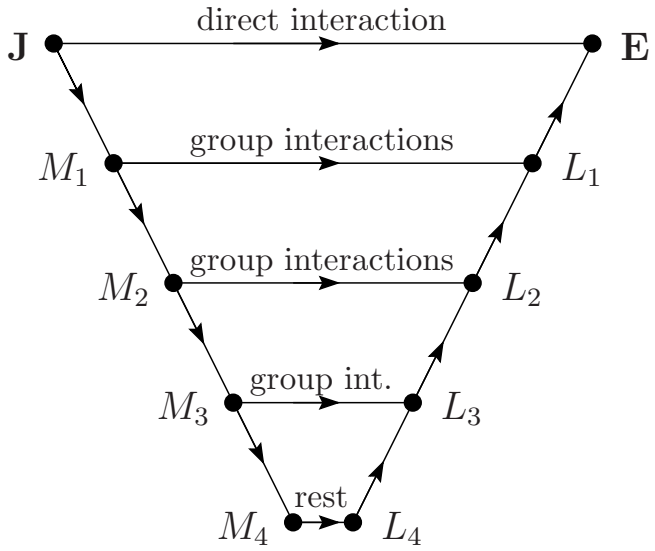


FIGURE 2.13: Schematic of the FMM algorithm. The M_ℓ's represent the multipole coefficients at level ℓ while the L_ℓ's represent the local expansion coefficients. **J** and **E** are the usual currents and electric fields. The arrows correspond to linear operators.

rows, are linear. As a result, the total FMM operator is a linear operator that is an approximation to the interaction matrix A.

The double hierarchical FMM scheme reduces the computational cost to $\mathcal{O}(n)$. However, it is possible to argue that the FMM is not a true $\mathcal{O}(n)$ scheme. The argument is that if the problem size is increased in order to get a more accurate solution, the order of the expansions should also be increased accordingly. This leads to more expansion coefficients and would lead to a higher complexity.

The Barnes-Hut method and the FMM were originally used for gravitational and electrostatic potentials, which behave like $1/r$, where r is the distance. However, in our electrodynamic problem, the potential oscillates like e^{-ikr}/r. As a result, the potential is less smooth than the $1/r$ potential, which leads to slower convergence of the expansions. Both methods can still be applied to the oscillating potential [14].

Chapter 3

Basis transformation

3.1 Matrix properties

In (partial) differential equations like the time-harmonic vacuum Maxwell equations (2.5), the interaction is *local*, i.e. the fields are coupled to derivatives of those fields at the same location. When making the step to the integral equation (2.14), we got a *global* interaction, i.e. a localised current contributes to the electric field everywhere in space, but the field strength decays with the distance to the current. This is represented by the Green function (2.15). As a result, every element in the capacitive matrix C (equation (2.33b)) and the inductive matrix L (equation (2.33c)) potentially has a non-zero value. In other words, these matrices are dense. However, we know that the value of the elements will decrease inversely proportional with the interaction distance.

The resistance matrix R (equation (2.22d)) is a direct result from the boundary condition (2.9), which is a local boundary condition. Consequently, R_{ij} is only non-zero if the basis functions $\mathbf{\Psi}_i$ and $\mathbf{\Psi}_j$ overlap, and therefore R is a sparse matrix. For most EMC applications, the surface impedance of the conductors is really small, leading to very small elements in R, compared with the corresponding elements of C and L.

Since the Green function G (equation (2.15)) is symmetric, i.e. $G(\mathbf{x}, \mathbf{x}') = G(\mathbf{x}', \mathbf{x})$, the matrices C and L are symmetric if the basis functions $\mathbf{\Psi}$ are chosen to be real valued. In this case, the resistance matrix R is also symmetric. However, the complex exponential in the Green function G and the complex surface impedance Z make the elements complex, leading to complex symmetric matrices.

The largest elements in C and L are those corresponding to short range interactions, since G is larger for smaller distances. For these small distances the complex exponential in G will have a small argument and G will be almost real. Due to the imaginary factors, these large elements in C and L will be nearly imaginary. Since these are the largest elements, we may expect that most eigenvalues lie close to the imaginary axis.

The matrix A is the sum of C, L, and R, and will thus inherit all these properties. A is dense, complex symmetric, has its largest elements for small distance interactions, which are near imaginary, and most eigenvalues are close to the imaginary axis. Using energy conservation arguments, which we will explain next, we can even show that all eigenvalues of A should have positive real parts. Unfortunately, for most eigenvalues, the real part is small compared with the imaginary part.

From (2.22e), we conclude that for any solution x of $Ax = b$

$$\begin{aligned} x^{\mathrm{H}} b &= \sum_{i=1}^{n} x_i^* \int_\Gamma \mathbf{E}^E(\mathbf{x}) \cdot \mathbf{\Psi}_i(\mathbf{x})^* d^2\mathbf{x} \\ &= \int_\Gamma \mathbf{E}^E(\mathbf{x}) \cdot \mathbf{J}(\mathbf{x})^* d^2\mathbf{x} \\ &= \int_\Gamma \left\langle \mathbf{E}^E(\mathbf{x},t) \cdot \mathbf{J}(\mathbf{x},t) \right\rangle_t d^2\mathbf{x} \quad , \end{aligned} \tag{3.1}$$

where the $\langle\ \rangle_t$ denotes the average of the real physical time dependent quantities over one period of the oscillation. The latter equality follows from the harmonic time dependence (2.4) :

$$\begin{aligned} A(t)B(t) &= \mathrm{Re}\left(Ae^{i\omega t}\right) \mathrm{Re}\left(Be^{i\omega t}\right) \\ &= \frac{1}{2}\left(\mathrm{Re}\left((Ae^{i\omega t})(Be^{i\omega t})\right) + \mathrm{Re}\left((Ae^{i\omega t})^*(Be^{i\omega t})\right)\right) \\ &= \frac{1}{2}\left(\mathrm{Re}\left(ABe^{2i\omega t}\right) + \mathrm{Re}\left(A^*B\right)\right) \quad . \end{aligned} \tag{3.2}$$

Since the time average of $e^{2i\omega t}$ vanishes, we get

$$\left\langle A(t)B(t) \right\rangle_t = \frac{1}{2}\mathrm{Re}\left(A^*B\right) \quad . \tag{3.3}$$

Physically, $\mathbf{E}^E(\mathbf{x},t)\cdot\mathbf{J}(\mathbf{x},t)$ denotes the power (energy per unit time) that the external electric field $\mathbf{E}^E$ puts into the current $\mathbf{J}$. Since we have a steady state, averaged over a period this must be equal to the energy loss due to resistance and radiation. This has to be positive for (non-trivial) solutions. In principle, this is true for all exact solutions, but not necessarily for the solutions of the discretised system. We can only speculate that, for a fine enough discretisation, this might also apply for the discrete problem. This would imply that, for all x and b satisfying $Ax = b$, the property $\mathrm{Re}\left(x^{\mathrm{H}}b\right) > 0$ should hold. As a special case, for eigenpairs $Av = \lambda v$, we get

$$\mathrm{Re}\left(v^{\mathrm{H}}\lambda v\right) = \mathrm{Re}\left(\lambda\right) > 0 \quad . \tag{3.4}$$

However, there are often many strongly oscillating eigenmodes that radiate very little energy due to cancellation, and when the surface impedance Z is small, the total energy loss can be very small, leading to many eigenvalues with very small real part.

Equation (3.4) shows that we expect the matrix A to be positive definite. In the context of Krylov subspace solvers (see section 1.2), this is a favourable property since the origin is not contained in the convex hull of the spectrum of A (see section 1.2). However, at the beginning of this section, we argued that many eigenvalues are close to the imaginary axis, and in the next subsection we will see that one part is close to the positive and another part is close to the negative imaginary axis. This puts the origin very close to the convex hull of the spectrum of A, destroying the advantage one might have expected from the fact that A is positive definite.

The first term in the electric field integral equation (EFIE) (2.16) represents the capacitive field, which is the electric field due to the charge accumulation. It is directly

seen that if the current $\mathbf{J}$ is divergence free, this term will vanish and there will be no capacitive contribution to the electric field. If this divergence free current can be numerically represented by the vector x, then

$$\nabla_\Gamma \cdot \mathbf{J}(\mathbf{x}) = \nabla_\Gamma \cdot \sum_i x_i \mathbf{\Psi}_i(\mathbf{x}) = \sum_i x_i \nabla_\Gamma \cdot \mathbf{\Psi}_i(\mathbf{x}) = 0 \ , \tag{3.5}$$

and the numerical representation of the generated capacitive part of the field, Cx, must also be zero. This can readily be checked by combining equation (3.5) with equation (2.33b). This means that the capacitive matrix C has a null space containing all vectors representing divergence-free currents.

The simplest numerically represented divergence free current is a small loop current, as shown in Figure 3.1. For every internal vertex in the grid, there is such a small loop current, and all these divergence-free currents are independent. Depending on the model geometry, the number of these small loops will be in the order of 1/2 of the total number of degrees of freedom for a quadrilateral discretisation and 1/3 of the total number of degrees of freedom for a triangular discretisation. Apart from these local loops, there can also be independent global loops, with corresponding independent divergence-free currents. All together, these independent current loops span the null space of C.

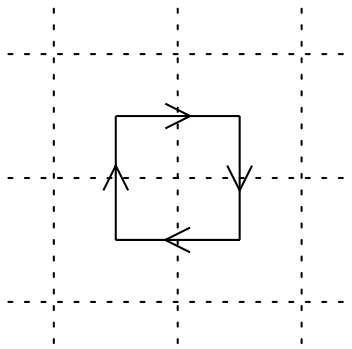


FIGURE 3.1: Simple loop current on a square grid.

From the definitions (2.33) we see that $C \sim \frac{1}{\omega}$, and $L \sim \omega$. This implies that for small enough frequencies ω (wavelength $\lambda = 2\pi c/\omega$ much larger than the largest length scale of the conductor), the inductive part L will be much smaller than the capacitive part C. Since the resistive part R is usually very small, the capacitive effects will dominate. However, on the null space of C only the inductive and resistive effects are present, which are relatively small. This results in a cluster of relatively small eigenvalues of the dimension of the null space of C. These eigenvalues can be several orders of magnitude smaller than the remainder of the eigenvalues. As we will show in section 3.1.2, a factor of 10^7 difference between the largest and the smallest eigenvalue is not extreme. Such a large cluster of very small eigenvalues is a big problem for iterative solvers.

To make this more precise, in the next section we analyse the continuous spectrum of an infinite board and infinite wire.

3.1.1 Fourier analysis

In this section we try to get an impression of the properties of the EFIE (2.16) that we have to solve. In order to get any analytic results, we have to restrict ourselves to the very simple geometries of an infinite flat plane and an infinite straight wire. These are the two "fundamental" building blocks for our models, and might give us some idea about their behaviour. For these geometries we will derive the exact eigenvalues and eigenfunctions of the integral operator in the EFIE and try to use these to draw conclusions for the discretised operator in matrix A.

The infinite board

In this section we will derive exact eigenfunctions and eigenvalues for the continuous infinite board problem. We look at the EFIE (2.16) where the conductor surface is an infinite plane with constant surface impedance Z. Any current can be decomposed in longitudinal and transversal plane waves so we will study the result of the action of the operator on the left hand side of equation (2.16) on the separate plane waves. For ease of notation, we will call this operator $\mathcal{A}$:

$$\begin{aligned}(\mathcal{A}\mathbf{J})(\mathbf{x}) &= (\mathcal{C}\mathbf{J})(\mathbf{x}) + (\mathcal{L}\mathbf{J})(\mathbf{x}) + (\mathcal{R}\mathbf{J})(\mathbf{x}) \\ &= \frac{i}{\omega}\int_\Gamma \nabla_\Gamma G(\mathbf{x},\mathbf{x}')\nabla'_\Gamma \cdot \mathbf{J}(\mathbf{x}')d^2\mathbf{x}' + \frac{i\omega}{c^2}\int_\Gamma G(\mathbf{x},\mathbf{x}')\mathbf{J}(\mathbf{x}')d^2\mathbf{x}' + Z(\mathbf{x})\mathbf{J}(\mathbf{x}) \ ,\end{aligned} \tag{3.6}$$

where the capacitive, inductive, and resistive operators $\mathcal{C}$, $\mathcal{L}$, and $\mathcal{R}$ correspond to the three terms of $\mathcal{A}$.

We consider general plane wave currents with wavenumber $\mathbf{q}$ and current direction $\mathbf{a}$ ($\mathbf{q}, \mathbf{a} \in \mathbb{R}^2$) :

$$\mathbf{J}_{\mathbf{q},\mathbf{a}}(\mathbf{x}) = \mathbf{a}e^{i\mathbf{q}\cdot\mathbf{x}} \ . \tag{3.7}$$

The inductive field is given by

$$\begin{aligned}\mathbf{E}^L_{\mathbf{q},\mathbf{a}}(\mathbf{x}) = (\mathcal{L}\mathbf{J}_{\mathbf{q},\mathbf{a}})(\mathbf{x}) &= \frac{ik}{c}\int G(\mathbf{x}-\mathbf{x}')\mathbf{J}_{\mathbf{q},\mathbf{a}}(\mathbf{x}')d^2\mathbf{x}' \\ &= \frac{ik\mathbf{a}}{c}\int G(\mathbf{x}-\mathbf{x}')e^{i\mathbf{q}\cdot\mathbf{x}'}d^2\mathbf{x}' \\ &= \frac{ik\mathbf{a}}{c}e^{i\mathbf{q}\cdot\mathbf{x}}\int G(\mathbf{y})e^{-i\mathbf{q}\cdot\mathbf{y}}d^2\mathbf{y} \\ &= \frac{ik\mathbf{a}}{c}e^{i\mathbf{q}\cdot\mathbf{x}}\widehat{G}(\mathbf{q}) \ ,\end{aligned} \tag{3.8}$$

where $G(\mathbf{y}) = \frac{e^{-ik|\mathbf{y}|}}{|\mathbf{y}|}$ is the Green function and $\widehat{G}(\mathbf{q})$ is its 2-dimensional Fourier transform. This Fourier transform cannot be computed directly but when we first compute the Fourier transform of the regularised function $\frac{e^{-ik|\mathbf{y}|}}{|\mathbf{y}|}e^{-\mu|\mathbf{y}|}$ and then let μ go to zero along the positive real axis, we find that

$$\widehat{G}(\mathbf{q}) = -i\frac{2\pi}{\sqrt{k^2 - |\mathbf{q}|^{2^*}}} \qquad \text{for} \quad k \neq |\mathbf{q}| \ , \tag{3.9}$$

where the * denotes the complex conjugation of the root. In a similar fashion, we find the capacitive field

$$\begin{aligned}\mathbf{E}^C_{\mathbf{q},\mathbf{a}}(\mathbf{x}) = (\mathcal{C}\mathbf{J}_{\mathbf{q},\mathbf{a}})(\mathbf{x}) &= \frac{i}{ck}\nabla\int G(\mathbf{x}-\mathbf{x}')\nabla'\cdot\mathbf{J}_{\mathbf{q},\mathbf{a}}(\mathbf{x}')d^2\mathbf{x}' \\ &= -\frac{i}{ck}(\mathbf{q}\cdot\mathbf{a})\mathbf{q}e^{i\mathbf{q}\cdot\mathbf{x}}\widehat{G}(\mathbf{q})\end{aligned} \tag{3.10}$$

and the resistive part

$$\mathbf{E}^R_{\mathbf{q},\mathbf{a}}(\mathbf{x}) = (\mathcal{R}\mathbf{J}_{\mathbf{q},\mathbf{a}})(\mathbf{x}) = Z\mathbf{a}e^{i\mathbf{q}\cdot\mathbf{x}} \ . \tag{3.11}$$

Adding this together we find

$$\mathbf{E}_{\mathbf{q},\mathbf{a}}(\mathbf{x}) = (\mathcal{A}\mathbf{J}_{\mathbf{q},\mathbf{a}})(\mathbf{x}) = \left(\frac{ik}{c}\widehat{G}(\mathbf{q})\mathbf{a} - \frac{i}{ck}(\mathbf{q}\cdot\mathbf{a})\widehat{G}(\mathbf{q})\mathbf{q} + Z\mathbf{a}\right)e^{i\mathbf{q}\cdot\mathbf{x}} \ . \tag{3.12}$$

We are looking for the eigenpairs of $\mathcal{A}$, defined by

$$\mathcal{A}\mathbf{J} = \lambda\mathbf{J} \ . \tag{3.13}$$

Just trying whether the $\mathbf{J}_{\mathbf{q},\mathbf{a}}$ might be eigenvectors, gives us the condition

$$\mathcal{A}\mathbf{J}_{\mathbf{q},\mathbf{a}} = \lambda_{\mathbf{q},\mathbf{a}}\mathbf{J}_{\mathbf{q},\mathbf{a}} \Leftrightarrow \tag{3.14a}$$

$$\left(\frac{ik}{c}\widehat{G}(\mathbf{q})\mathbf{a} - \frac{i}{ck}(\mathbf{q}\cdot\mathbf{a})\widehat{G}(\mathbf{q})\mathbf{q} + Z\mathbf{a}\right)e^{i\mathbf{q}\cdot\mathbf{x}} = \lambda_{\mathbf{q},\mathbf{a}}\mathbf{a}e^{i\mathbf{q}\cdot\mathbf{x}} \Leftrightarrow \tag{3.14b}$$

$$\frac{ik}{c}\widehat{G}(\mathbf{q})\mathbf{a} - \frac{i}{ck}(\mathbf{q}\cdot\mathbf{a})\widehat{G}(\mathbf{q})\mathbf{q} + Z\mathbf{a} = \lambda_{\mathbf{q},\mathbf{a}}\mathbf{a} \ . \tag{3.14c}$$

This condition is satisfied if either $\mathbf{q}$ is parallel to $\mathbf{a}$ or if the second (capacitive) term vanishes, which only happens if $\mathbf{q}$ is perpendicular to $\mathbf{a}$. In the case where $\mathbf{q}$ is perpendicular to $\mathbf{a}$, the eigenvectors are $\mathbf{J}_\perp(\mathbf{q}) = \mathbf{J}_{\mathbf{q},\mathbf{q}^\perp}$ and the corresponding eigenvalues are

$$\begin{aligned}\lambda_\perp(\mathbf{q}) &= \frac{ik}{c}\widehat{G}(\mathbf{q}) + Z \\ &= \frac{2\pi}{c}\left(1 - \frac{|\mathbf{q}|^2}{k^2}\right)^{-\frac{1}{2}^*} + Z \ ,\end{aligned} \tag{3.15}$$

and in the case where $\mathbf{q}$ is parallel to $\mathbf{a}$, the eigenvectors are $\mathbf{J}_\parallel(\mathbf{q}) = \mathbf{J}_{\mathbf{q},\mathbf{q}}$ and the corresponding eigenvalues are

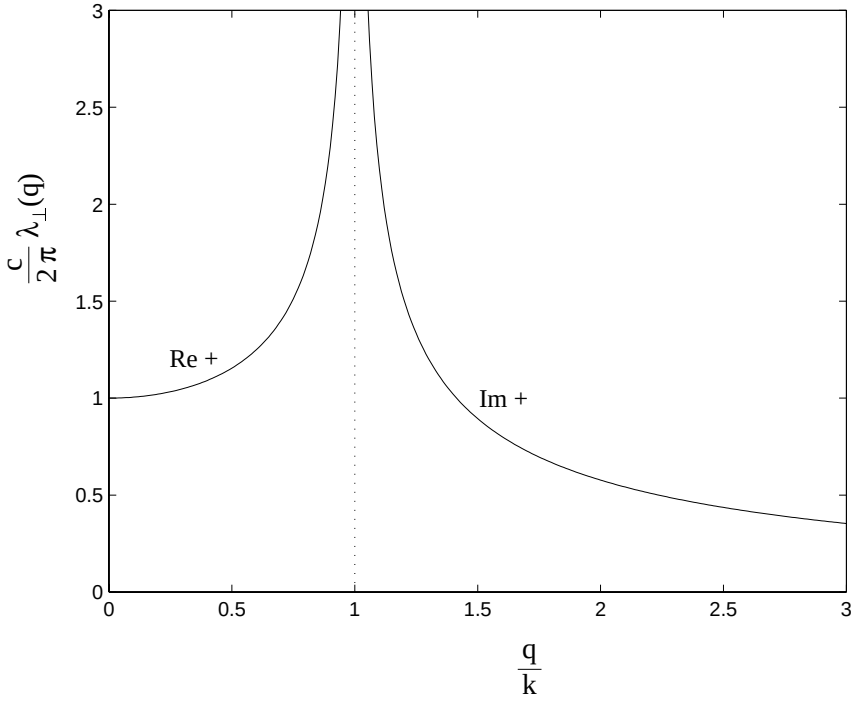
$$\begin{aligned}\lambda_\parallel(\mathbf{q}) &= \frac{ik}{c}\widehat{G}(\mathbf{q}) - \frac{i}{ck}|\mathbf{q}|^2\widehat{G}(\mathbf{q}) + Z \\ &= \frac{i}{c}\left(k - \frac{|\mathbf{q}|^2}{k}\right)\widehat{G}(\mathbf{q}) + Z \\ &= \frac{2\pi}{c}\left(1 - \frac{|\mathbf{q}|^2}{k^2}\right)^{\frac{1}{2}^*} + Z \ .\end{aligned} \tag{3.16}$$

These eigenvectors $\mathbf{J}_\perp(\mathbf{q})$ and $\mathbf{J}_\parallel(\mathbf{q})$ form a complete orthogonal Fourier basis for the space. This implies that we have found all eigenvectors.

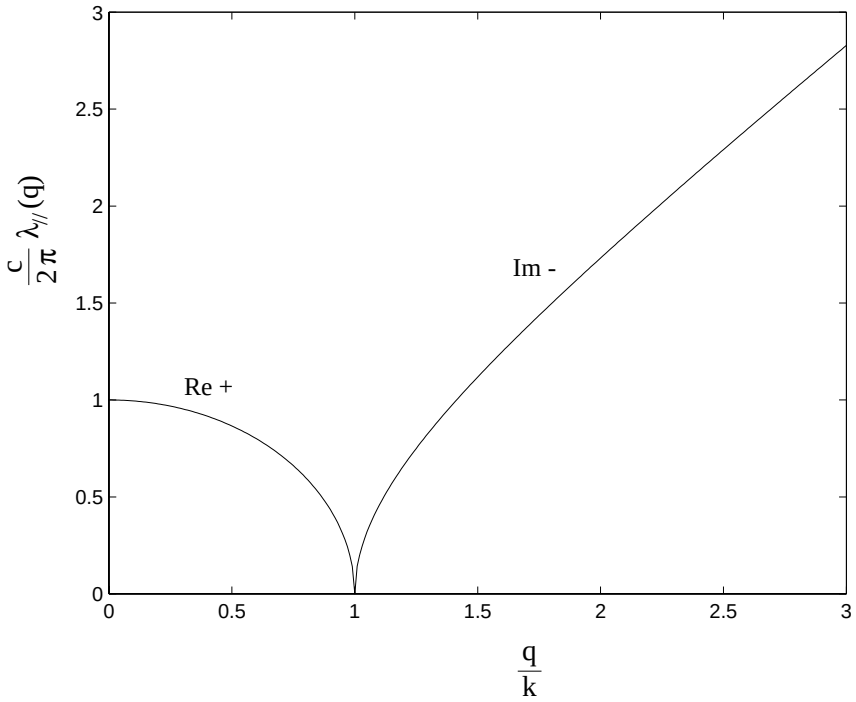
The eigenvalues are plotted in Figure 3.2. Note that the currents $\mathbf{J}_\perp$ have no charge accumulation, so $\lambda_\perp$ corresponds only to inductive and resistive effects. This is reflected by the fact that $\lambda_\perp(\mathbf{q}) \sim k$ for $|\mathbf{q}|/k \to \infty$ (short range interactions), which is in agreement with the factor ω in the inductive term in the EFIE (2.16). The longitudinal wave does have charge accumulation and therefore $\lambda_\parallel$ also corresponds to capacitive effects. As a result we see that $\lambda_\parallel(\mathbf{q}) \sim 1/k$ for $|\mathbf{q}|/k \to \infty$, which corresponds to the factor $1/\omega$ in the capacitive term in the EFIE (2.16).

Note that, in the absence of resistance,

$$\lambda_\parallel(|\mathbf{q}| = k) = 0 \ , \tag{3.17}$$



(a) Eigenvalues for transversal current waves



(b) Eigenvalues for longitudinal current waves

FIGURE 3.2: The continuous eigenvalues for the transversal ($\lambda_{\perp}$) and longitudinal ($\lambda_{\parallel}$) current waves on the infinite plane conductor in absence of resistance ($Z = 0$). To include resistance, Z should be added to the eigenvalue. The absolute value is shown while the "Re +/-" and "Im +/-" show on which complex axis the values should be.

which means that the driving force of the external field is not necessary to sustain the corresponding current $\mathbf{J}_\parallel(|\mathbf{q}| = k)$, since $\mathcal{A}\mathbf{J}_\parallel(|\mathbf{q}| = k) = 0$. This also means that if there is a driving force $\mathbf{E}^E(\mathbf{x}) = \mathbf{q}e^{i\mathbf{q}\cdot\mathbf{x}}$ with $|\mathbf{q}| = k$ for this mode, the current will keep absorbing energy and grow without restriction. A time-harmonic solution for this external field does not exist, since $(\mathcal{A}\mathbf{J})(\mathbf{x}) = \mathbf{E}^E$ will not have a finite solution. This is called resonance. In a realistic problem, we will have a (possibly small) resistance and we have a finite system size, so no resonances will occur, but we can still expect damped resonances. In this case, $\mathcal{A}$ has a very small eigenvalue, and a small driving force will result in large currents. This also means that the discretised system has at least one very small eigenvalue, which is likely to slow down the convergence of an iterative solver.

In section 3.2, we will also be using the electrostatic potential operator $\mathcal{D}$, so we will also look at the eigenvalues and eigenfunctions of this operator

$$(\widetilde{\mathcal{D}}\rho)(\mathbf{x}) = \int_\Gamma G(\mathbf{x}, \mathbf{x}')\rho(\mathbf{x}')d^2\mathbf{x}' \quad , \tag{3.18}$$

and the scaled operator $\mathcal{D} = -\frac{i}{\omega}\widetilde{\mathcal{D}}$. Assuming a plane wave for the charge density $\rho_\mathbf{q}(\mathbf{x}) = e^{i\mathbf{q}\cdot\mathbf{x}}$, we get

$$\begin{aligned}(\widetilde{\mathcal{D}}\rho_\mathbf{q})(\mathbf{x}) &= \int G(\mathbf{x} - \mathbf{x}')e^{i\mathbf{q}\cdot\mathbf{x}'}d^2\mathbf{x}' \\ &= e^{i\mathbf{q}\cdot\mathbf{x}}\widehat{G}(\mathbf{q}) \quad ,\end{aligned} \tag{3.19}$$

which shows that the $\rho_\mathbf{q}(\mathbf{x})$'s form an orthogonal eigenbasis with corresponding eigenvalues

$$\lambda_{\widetilde{D}}(\mathbf{q}) = \widehat{G}(\mathbf{q}) = -\frac{2\pi i}{k}\left(1 - \frac{|\mathbf{q}|^2}{k^2}\right)^{-\frac{1}{2}^*} \quad . \tag{3.20}$$

Since the $\mathcal{D}$ operator has an extra factor $-\frac{i}{\omega}$, its eigenvalues are

$$\lambda_D(\mathbf{q}) = -\frac{i}{\omega}\lambda_{\widetilde{D}}(\mathbf{q}) = -\frac{2\pi}{ck^2}\left(1 - \frac{|\mathbf{q}|^2}{k^2}\right)^{-\frac{1}{2}^*} \quad . \tag{3.21}$$

These have the same behaviour as $\lambda_\perp(\mathbf{q})$ in Figure 3.2(a) but with a $-k^{-2}$ scaling. We also see that $\lambda_D(\mathbf{q}) \sim 1/k$ for $|\mathbf{q}|/k \to \infty$.

The infinite wire

The other fundamental building block of our models are the wires. In this section we will compute the exact eigenfunctions and eigenvalues for the continuous infinite wire problem, using the same techniques we use above for the infinite board.

We consider an infinitely long straight wire with radius R and surface impedance Z. Under the model restriction that the surface current $\mathbf{J}$ is in the tangential direction (parallel to the wire axis) and uniform in a plane perpendicular to the wire, we have a Fourier basis for the surface current consisting of

$$\mathbf{J}_q(\mathbf{x}) = \widehat{\mathbf{z}}e^{iq\mathrm{x}_z} \quad , \tag{3.22}$$

where $\widehat{\mathbf{z}}$ is the unit vector in the tangential direction and x_z is the tangential component of $\mathbf{x}$: $\mathrm{x}_z = \widehat{\mathbf{z}} \cdot \mathbf{x}$. The resulting inductive field is given by

$$\begin{aligned}\mathbf{E}_q^L(\mathbf{x}) = (\mathcal{L}\mathbf{J}_{\mathbf{q},\mathbf{a}})(\mathbf{x}) &= \frac{ik}{c}\int_\Gamma G(\mathbf{x}-\mathbf{x}')\mathbf{J}_q(\mathbf{x}')d^2\mathbf{x}' \\ &= \frac{ik\widehat{\mathbf{z}}}{c}\int_\Gamma G(\mathbf{x}-\mathbf{x}')e^{iq\mathbf{x}'_z}d^2\mathbf{x}' \\ &= \frac{ik\widehat{\mathbf{z}}}{c}e^{iq\mathrm{x}_z}\int_\Gamma G(\mathbf{y})e^{-iq\mathrm{y}_z}d^2\mathbf{y} \\ &= \frac{ik\widehat{\mathbf{z}}}{c}e^{iq\mathrm{x}_z}\widetilde{G}(q) \ ,\end{aligned} \tag{3.23}$$

where we have introduced a special 1-dimensional Fourier transform of G on the wire surface Γ, given by

$$\begin{aligned}\widetilde{G}(q) &= \int_\Gamma G(\mathbf{y})e^{-iq\mathrm{y}_z}d^2\mathbf{y} \\ &= \int_\Gamma \frac{e^{-ik|\mathbf{y}|}}{|\mathbf{y}|}e^{-iq\mathrm{y}_z}d^2\mathbf{y} \\ &= 2R\int_{z=0}^{\infty}\int_{\theta=0}^{2\pi}\frac{e^{-ikR\sqrt{2-2\cos\theta+z^2}}}{\sqrt{2-2\cos\theta+z^2}}\cos(qRz)d\theta dz \\ &= 8R\int_0^\pi K_0\left(2R\sqrt{q^2-k^2}\sin(\theta/2)\right)d\theta \ ,\end{aligned} \tag{3.24}$$

with K_n the modified Bessel functions of the second kind. Unfortunately, this cannot be evaluated analytically, but we can see that $\widetilde{G}(q)$ has a singularity for $|q| = k$. We can evaluate the integral numerically, so we continue by computing the capacitive field

$$\begin{aligned}\mathbf{E}_q^C(\mathbf{x}) = (\mathcal{C}\mathbf{J}_{\mathbf{q},\mathbf{a}})(\mathbf{x}) &= \frac{i}{ck}\nabla\int_\Gamma G(\mathbf{x}-\mathbf{x}')\nabla'\cdot\mathbf{J}_q(\mathbf{x}')d^2\mathbf{x}' \\ &= -\frac{iq^2}{ck}\widehat{\mathbf{z}}e^{iq\mathrm{x}_z}\widetilde{G}(q)\end{aligned} \tag{3.25}$$

and the resistive part

$$\mathbf{E}_q^R(\mathbf{x}) = (\mathcal{R}\mathbf{J}_{\mathbf{q},\mathbf{a}})(\mathbf{x}) = Z\widehat{\mathbf{z}}e^{i\mathbf{q}\cdot\mathbf{x}} \ . \tag{3.26}$$

Adding (3.23), (3.25), and (3.26) we find

$$\mathbf{E}_q(\mathbf{x}) = (\mathcal{A}\mathbf{J}_{\mathbf{q},\mathbf{a}})(\mathbf{x}) = \left(\frac{ik}{c}\widetilde{G}(q) - \frac{iq^2}{ck}\widetilde{G}(q) + Z\right)\mathbf{J}_q(\mathbf{x}) \ , \tag{3.27}$$

which shows that $\mathbf{J}_q$ is an eigenfunction with eigenvalue

$$\begin{aligned}\lambda(q) &= \frac{ik}{c}\left(1-\frac{q^2}{k^2}\right)\widetilde{G}(q) + Z \\ &= 8\frac{ikR}{c}\left(1-\frac{q^2}{k^2}\right)\int_0^\pi K_0\left(2kR\sin(\theta/2)\sqrt{(q/k)^2-1}\right)d\theta + Z \ .\end{aligned} \tag{3.28}$$

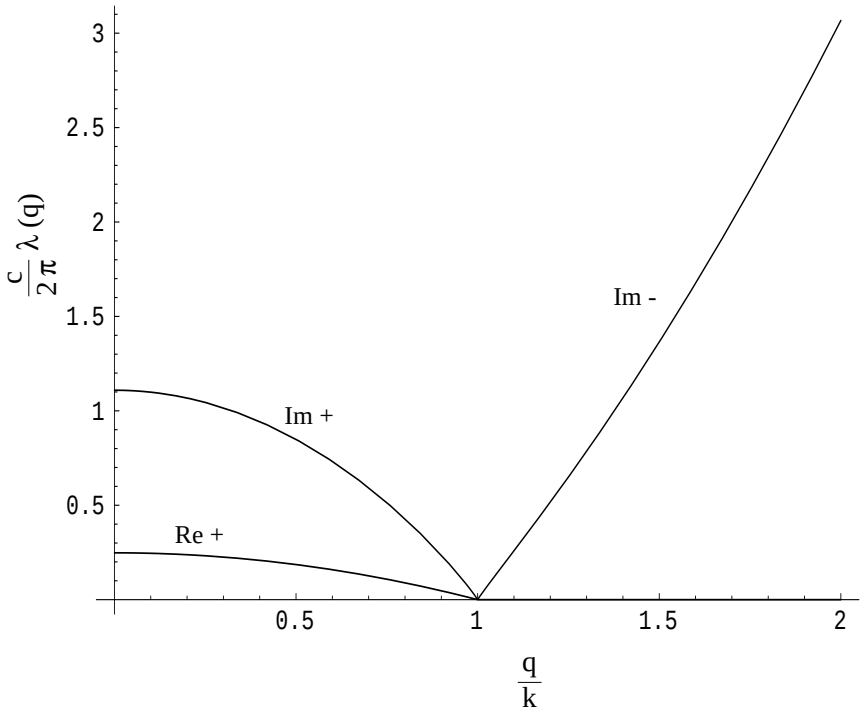


FIGURE 3.3: The continuous eigenvalues for the current waves on the infinite thin wire in absence of resistance ($Z = 0$). To include resistance, Z should be added to the eigenvalue. The real and imaginary parts are shown while the "Re +/-" and "Im +/-" show on which complex axis the values should be.

Note that $\lambda(q)$ not only depends on q/k and Z, but also on kR. Using numerical integration, we plotted $\lambda(q)$ in Figure 3.3. We used $kR = 0.001$, which is not an unreasonable value. We can see that the zero in the $1 - (q/k)^2$ term is stronger than the singularity in $\widetilde{G}(q)$, which leads to a resonance for $|q| = k$. In chapter 8 of Jackson [26], we find that the infinite cylindrical conductor has this resonance for $q = k$, which is known as the transverse electromagnetic (TEM) mode. The other resonant modes of the infinite cylindrical conductor are not allowed by the model restrictions mentioned above equation (3.22). Without these restrictions, other resonances would only occur at very high frequencies ($\lambda \lessapprox R$). For more information on resonant modes of conducting wires see reference [26], chapter 8.

Just as we did for the infinite board, we can also look for the eigenfunctions and eigenvalues of the operator $\mathcal{D}$ for the wire. In a similar way, we derive that the eigenfunctions are

$$\rho_q = e^{iq\mathrm{x}_z} \quad , \tag{3.29}$$

with eigenvalues

$$\lambda_D(q) = \frac{i}{\omega}\widetilde{G}(q) \quad . \tag{3.30}$$

3.1.2 Discussion of the Fourier analysis

Using a uniform grid, the continuous spectra of $\mathcal{A}$ for the board and wire may give an indication of the behaviour of the discrete spectrum of A. In the case of a discretised board/wire of finite size, we might expect the eigenvectors to correspond approximately to the continuous eigenvectors that approximately satisfy the edge conditions (no current flow though the edge) and that can be represented on the grid.

The edge condition excludes all modes for which half the spatial oscillation length ($\pi/|\mathbf{q}|$) is more than the largest length scale L of the board/wire, giving a lower cutoff for the wavenumber $\mathbf{q}$: $\pi/L \leq |\mathbf{q}|$. This is similar to the eigenmodes of a piano string, for which the lower cutoff for the wavenumber corresponds to the base tone, for which half the wavelength precisely fits the length of the string. For the higher wavenumbers, half the spatial oscillation length must fit an integer number of times in the length L, and so $|\mathbf{q}|$ must be an integer multiple of this lower cutoff (higher harmonics for the string). This gives an approximately uniform distribution of $\mathbf{q}$ values. So far, these restrictions are due to the finite size of the conductor, and are real physical restrictions. The exact shape eigenfunctions and position of the eigenvalues is strongly dependent on the exact shape of the conductor, but this dependence will decrease for larger wavenumbers $\mathbf{q}$.

The grid introduces artificial restrictions on the numerical eigenfunctions of A. In order to be represented on a grid with grid size h, the spatial oscillation length ($2\pi/|\mathbf{q}|$) must be at least $2h$, giving an upper cutoff for the wavenumber: $|\mathbf{q}| \leq \pi/h$. The exact behaviour of the high $\mathbf{q}$ side of the spectrum is governed by the precise discretisation. This dependence decreases for smaller values of $\mathbf{q}$ since these modes can be represented better on all types of grids with the same h.

These arguments give upper and lower cutoff frequencies of the order of $\pi/L \leq |q| \leq \pi/h$, with uniformly distributed values in between. Due to the $|\mathbf{q}| = k$ resonance we saw in (3.17) and (3.28), we expect the solution to oscillate with wavelength λ over interior regions. Consequentially, we have to choose a grid sufficiently fine to be able to accurately represent this resonant mode. In practice $h \leq \lambda/20$ is used, which means that $\pi/h \geq 20\pi/\lambda = 10k$. The result is that most of the eigenpairs of the matrix will be related to values of $\mathbf{q}$ in the region beyond the resonance ($|\mathbf{q}| > k$).

For the inductive part of the spectrum, the eigenvalues behave like $1/|\mathbf{q}|$ so that the high $|\mathbf{q}|$ eigenvalues tend to cluster towards the eigenvalue belonging to the $\mathbf{q}$ cutoff. This cutoff eigenvalue will tend to zero for increasingly fine grids, which leads to an increasing condition number of the matrix A. The capacitive part of the spectrum is proportional to $|\mathbf{q}|$, which does not lead to clustering, but here finer grids also lead to an increasing condition number.

In total this leads to the following estimate for the condition number of A for a board at low frequencies and no resistance

$$\kappa(A) \gtrsim \frac{\lambda_{\parallel}(\pi/h)}{\lambda_{\perp}(\pi/h)} = \frac{\pi^2}{h^2 k^2} = \frac{1}{4}\frac{\lambda^2}{h^2} \quad , \tag{3.31}$$

where possible resonances may lead to even worse values. For low frequencies, the edge effects and the geometry will dictate the element size h, leading to a much smaller h than required by the $h < \lambda/20$ mentioned above. Note that this estimate is independent of the size L of the board. As a conservative example, for a rather coarse grid size of 1 cm

(for e.g. a device of $L \approx 20$ cm) and a moderately low frequency of 10 MHz (λ=30 m) we still get $\kappa(A) \gtrapprox 10^7$.

For the frequencies, not the high $|\mathbf{q}|$ modes but the (near) resonances lead to extreme eigenvalues, as they lead to very small eigenvalues for A. This occurs if there are modes for which $|\mathbf{q}|$ is very close to k. Since these resonances are related to the slowly varying modes ($|\mathbf{q}| \approx k$), they depend strongly on the geometry and the frequency, which makes it is much harder to get an estimate for the resulting condition number. An estimate would require the minimal distance $|\mathbf{q}| - k$, which is hard to get.

As an illustration, Figure 3.4(a) shows the spectrum of A for a simple example. We did not use a very low frequency here, so that the cluster of inductive eigenvalues is still visible along the positive imaginary axis. For much lower frequencies, it will appear as a single point in the origin. Figure 3.4(b) shows the spectrum of the discretisation of the electrostatic operator $\mathcal{D}$ (matrix D defined in equation (3.39)).

3.2 Constructing a new basis

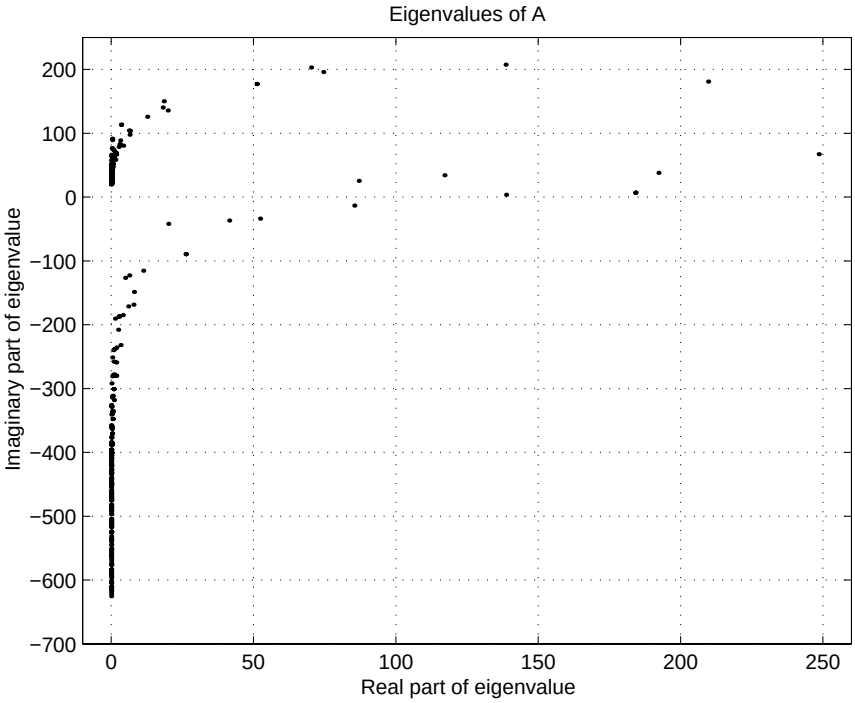
For low frequencies, standard preconditioning techniques like Gauss-Seidel [6], ILU [13], and many others will experience severe difficulties because of the large cluster of small eigenvalues dictated by $L + R$. Since the contribution of $L + R$ to the elements of A will be very small compared with the contribution of C, a standard preconditioner will not be able to capture the behaviour of A on the null space of C. It is possible to precondition the part dominated by C, but the large cluster of small eigenvalues will remain unaffected. We have not succeeded in finding a preconditioner that captures both the effects of C and of $L + R$.

To overcome the problem of the cluster of small eigenvalues, we will try to separate the contribution of $L + R$, visible in the small eigenvalues, from the contribution of C, seen in the large eigenvalues. To achieve this, we will use a basis for the small eigenvalue subspace and a basis for the remaining large eigenvalue subspace. The simplest way to define such a basis would be to use the basis of eigenvectors of A. In practice this is much to expensive to compute, however, we will show that it is possible to construct a basis that will achieve this separation at low computational cost.

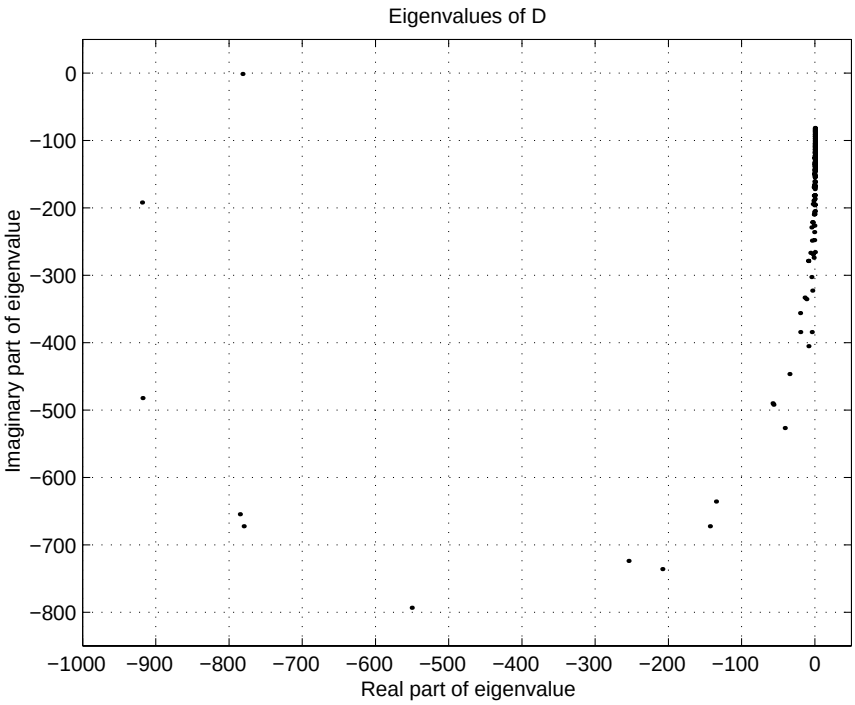
The new basis consist of two parts, K_l and K_c, and together they form the complete basis $Q = (K_l, K_c)$. In order to choose suitable K_l and K_c, we look at what happens to A if we change to the new basis :

$$A_Q \equiv Q^{\mathrm{T}} A Q = \begin{pmatrix} K_l^{\mathrm{T}} A K_l & K_l^{\mathrm{T}} A K_c \\ K_c^{\mathrm{T}} A K_l & K_c^{\mathrm{T}} A K_c \end{pmatrix} \quad . \tag{3.32}$$

Note that this new matrix A_Q is still symmetric. Our goal is to restrict the effects of C to the $K_c^{\mathrm{T}} A K_c$ block of A_Q. This can be achieved by choosing K_l such that $C K_l = 0$, since this would imply that also $K_l^{\mathrm{T}} C = 0$, using the symmetry of C. As we argued in section 3.1, the null space of C is the space of all the divergence free currents. Numerically, this space is spanned by the loop currents. These are currents that follow a closed loop in the discretisation. A complete collection of independent loop currents will thus give a basis for the null space of C, and can be used in K_l. It is fairly inexpensive to construct such a complete set of independent loops, some details about this process



(a) Eigenvalues of A



(b) Eigenvalues of D

FIGURE 3.4: The eigenvalues of A and D for a square board of 1×1 meter at 200 MHz ($\lambda = 1.5$ meter), discretised using 16×16 squares.

can be found in section 3.5. Most of these loop currents will be very local, as shown in Figure 3.1, and consist only of four current elements of the square discretisation, which makes this part of K_l sparse. Depending on the model geometry, there can be a few independent global loops, which will add a few dense vectors to K_l.

Defining K_l this way changes equation (3.32) to

$$A_Q \equiv Q^{\mathrm{T}} A Q = \begin{pmatrix} K_l^{\mathrm{T}}(L+R)K_l & K_l^{\mathrm{T}}(L+R)K_c \\ K_c^{\mathrm{T}}(L+R)K_l & K_c^{\mathrm{T}}(C+L+R)K_c \end{pmatrix} \quad , \tag{3.33}$$

and takes care of our objective to separate the contribution of C. This contribution is restricted to one block and as a result, the contribution of $L+R$ that determines the behaviour of A on the null space of C, has been made visible in the leading first block of A_Q, where it is not dominated by C. Now that this important part of $L+R$ is not hidden behind C any more, a preconditioner may be able to capture it.

The basis vectors K_c for the remainder of the space can still be chosen freely, as long as it completes the basis Q. K_c need not be orthogonal to K_l, but should not make small angles with K_l either, because this would lead to numerical problems in the solution process. Completing Q in some arbitrary way will lead to unpredictable results and can lead to a very badly conditioned $K_c^{\mathrm{T}} A K_c$ or very small angles between K_l and K_c. We need a structured K_c that is, in some sense, complementary to K_l. Since the first part of Q deals with chargeless currents, K_c should represent all possible charges. Furthermore, we know that the operator $\mathcal{D}$ associated with the electrostatic potential due to charges (equation (3.18)) is relatively well behaved (see section 3.1.1). We will try to construct a K_c such that the second diagonal block of A_Q approximates this electrostatic operator, in order to get this block to behave in the same nice way as $\mathcal{D}$. To achieve this, we define charge basis functions $\Phi_j(\mathbf{x})$, $j = 1 \ldots m$. Φ_j has unit charge in element j and is zero everywhere else. In section 2.4, we defined the current basis functions $\mathbf{\Psi}_i$ such that

$$\nabla_\Gamma \cdot \mathbf{\Psi}_i = \Phi_{l_i} - \Phi_{k_i} \quad , \tag{3.34}$$

where l_i and k_i are the elements that share edge i. We can express this by defining a matrix P by

$$P_{ij} = \begin{cases} +1 & \text{if } \mathbf{\Psi}_i \text{ takes charge from element } j \\ -1 & \text{if } \mathbf{\Psi}_i \text{ puts charge in element } j \\ 0 & \text{remaining part} \end{cases} \quad , \tag{3.35}$$

such that

$$\nabla_\Gamma \cdot \mathbf{\Psi}_i = \sum_j P_{ij} \Phi_j \quad . \tag{3.36}$$

P^{T} can be seen as the discretised divergence operator: if $\mathbf{J} = \sum x_i \mathbf{\Psi}_i$, then

$$\nabla \cdot \mathbf{J} = \sum_i x_i \nabla \cdot \mathbf{\Psi}_i = \sum_{ij} x_i P_{ij} \Phi_j = \sum_j (P^{\mathrm{T}} x)_j \Phi_j \quad . \tag{3.37}$$

When equation (3.36) is substituted in equation (2.33), we find

$$C = P D P^{\mathrm{T}} \quad , \tag{3.38}$$

in which

$$D_{ij} = -\frac{i}{\omega} \iint_\Gamma \Phi_i(\mathbf{x})^* \; G(\mathbf{x}, \mathbf{x}') \; \Phi_j(\mathbf{x}') \; d^2\mathbf{x}' d^2\mathbf{x} \tag{3.39}$$

is a discretisation of $\mathcal{D}$, a scaled version of the scaled electrostatic integral operator $\widetilde{\mathcal{D}}$.

For notational clarity, we will restrict ourselves to the case of a single connected conductor surface Γ. It is straightforward to generalise this to multiple disconnected conductors. Since P^{T} is the discretised divergence operator and K_l only contains divergence free currents, we conclude that $P^{\mathrm{T}} K_l = 0$, confirming that $C K_l = 0$ (because of (3.38)). We also see that $P\mathbf{1} = 0$ where $\mathbf{1} = (1, 1, \ldots, 1)^{\mathrm{T}}$. This is related to the fact that the total charge accumulation of a current is always zero: if $\mathbf{J} = \sum x_i \mathbf{\Psi}_i$, then

$$\int_\Gamma \nabla \cdot \mathbf{J}(\mathbf{x}) \; d^2\mathbf{x} = \int_\Gamma \sum_{ij} x_i P_{ij} \Phi_j(\mathbf{x}) \; d^2\mathbf{x} = \sum_{ij} x_i P_{ij} = x^{\mathrm{T}} P \mathbf{1} = 0 \;\; , \tag{3.40}$$

for any x.

Our goal was to choose K_c such that $K_c^{\mathrm{T}} A K_c$ approximates the matrix D of equation (3.39). Since this part of A_Q is dominated by C, we will try to construct K_c in such a way that $K_c^{\mathrm{T}} C K_c$ approximates D. If we do this, we may expect that $K_c^{\mathrm{T}} A K_c$ also approximates D.

To make $K_c^{\mathrm{T}} C K_c = K_c^{\mathrm{T}} P D P^{\mathrm{T}} K_c$ equal to D, we have to choose K_c such that $K_c^{\mathrm{T}} P = P^{\mathrm{T}} K_c = I$. Unfortunately, this is impossible because P does not have full rank, since $P\mathbf{1} = 0$. Since $\mathbf{1}$ is the only non-trivial vector for which the projection P gives the zero vector, we can choose K_c such that

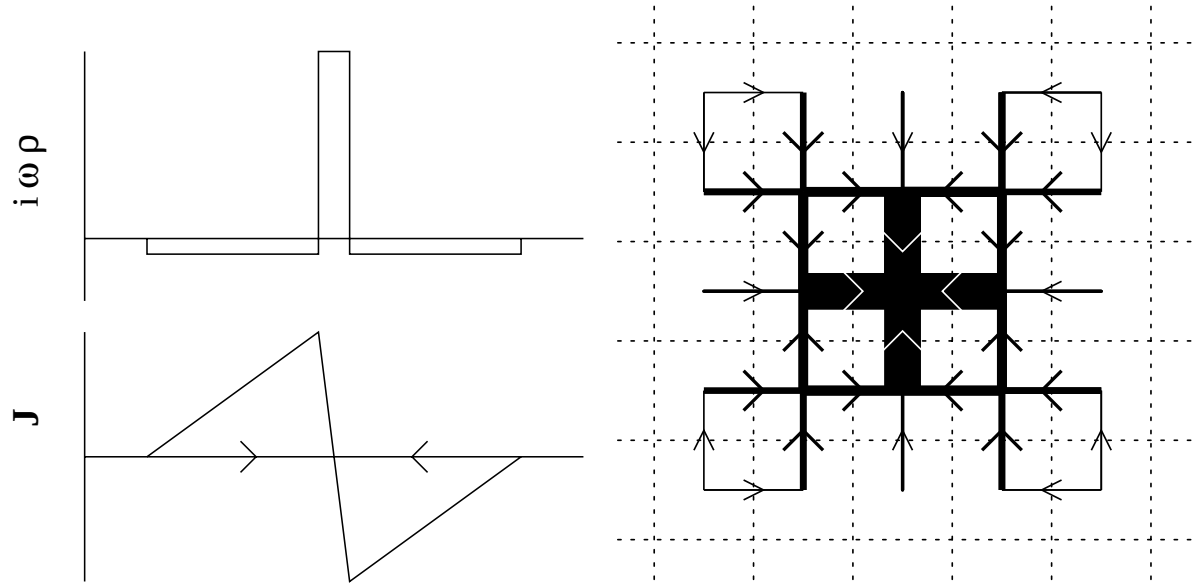
$$K_c^{\mathrm{T}} P = P^{\mathrm{T}} K_c = I - \frac{1}{m} \mathbf{1}\,\mathbf{1}^{\mathrm{T}} \;\; , \tag{3.41}$$

which is a projection on the space of total zero charge. $m = \|\mathbf{1}\|^2$ is the number of basis vectors in K_c. We now see that

$$K_c^{\mathrm{T}} C K_c = K_c^{\mathrm{T}} P D P^{\mathrm{T}} K_c = (I - \frac{1}{m} \mathbf{1}\,\mathbf{1}^{\mathrm{T}}) D (I - \frac{1}{m} \mathbf{1}\,\mathbf{1}^{\mathrm{T}}) \;\; , \tag{3.42}$$

which is D restricted to the space of total zero charge and potential. This is physically the same, because the total charge is zero and only potential differences are important.

If we keep in mind that P^{T} is the discretised divergence operator (equation (3.37)), the relation (3.41) shows that the j-th column of K_c represents a current with divergence $1 - 1/m$ on element j and divergence $-1/m$ on all other elements. Using this, we can construct a K_c, but this will be expensive and K_c will be dense, which also makes basis transformations expensive. However, we can make a cheap sparse approximation by quasi-charge currents. In our implementation, we approximate the K_c described above with an extra minus sign. In this case the j-th column of K_c represents a current with divergence -1 on element j and divergence $1/k$ on k surrounding elements. For the essentially 1-dimensional wire, the divergence requirements fully determine the current, which is shown in Figure 3.5(a). For the 2-dimensional board, the current is chosen to transport the charge by an approximately radial flow to the central element, keeping as much symmetry as possible. In Figure 3.5(b), we show our flow pattern for the regular square mesh with $k = 24$. The flow pattern, together with the divergence requirements,



(a) Current and charge of the quasi-charge basis functions on the wires.

(b) Current for the quasi-charge basis functions on the boards. The dotted lines denote the mesh, the arrows the current (thickness is proportional to the current).

FIGURE 3.5: The quasi-charge basis functions.

fully determines the current. For more details about the construction of the quasi-charge currents see section 3.5.

By choosing these quasi-charge currents as the second part of the new basis, we have constructed an "over-complete" basis Q with $n+1$ vectors. The null space of Q will correspond to the null space of K_c. For the ideal K_c this was the $\mathbf{1}$ vector, but the approximation with the quasi-charge currents changes this vector somewhat. It is convenient to know this vector in order to avoid problems in the iterative solver, and will need this vector for the multigrid preconditioning in section 4.4.2. In order to ensure that we know the null space of K_c, we slightly adapt our K_c to have zero row sums while preserving the sparsity. This is done by adding a correction term to all non-zero elements in a row. This correction term is the same for all elements in that row. Hence the $\mathbf{1}$ vector is again the null space of the adapted K_c.

The basis transformation changes the linear system (2.21) to

$$A_Q x_Q = b_Q \quad , \tag{3.43}$$

with $b_Q = Q^{\mathrm{T}} b$ and $x = Q x_Q$. Note that although the new matrix A_Q is singular, system (3.43) has a solution because the right-hand side b_Q is in the range of A_Q :

$$b_Q = Q^{\mathrm{T}} b \in \mathrm{range}(Q^{\mathrm{T}}) = \mathrm{range}(Q^{\mathrm{T}} A Q) = \mathrm{range}(A_Q) \quad . \tag{3.44}$$

The system even has an infinite number of solutions, since any element of the null space

of Q can be added to the solution to obtain another solution. However, all solutions x_Q give one unique x which is equal to the solution of (2.21).

3.3 The continuous analogue

The new basis $Q = (K_l, K_c)$ can also be seen as an approximate discrete Helmholtz decomposition, as it decomposes our finite element space for the currents $\mathbf{J}$ into a divergence free part K_l and a (non-orthogonal) complement K_c. The Helmholtz decomposition has been used before (in combination with multigrid) for solving partial differential equations related to our integral equation (see [2] (2-dimensional) and [24] (3-dimensional)). However, in [2] and [24], only the K_l part of the new basis is used.

The discrete basis transformation (3.33) described in the previous section can be seen as an approximation to an operator projection of the original integral equation (2.16). To get some more insight in what is happening in this transformation, we will construct and analyse this continuous analogy to the basis transformation (3.33).

The first part of our new basis, K_l, contains all current loops, most of which are small local loops. From this, we can see that applying K_l^{T} to a discrete representation of an electric field E will measure the local rotation, so the continuous analogy of K_l^{T} is the operator $\mathcal{K}_l^{\mathrm{T}} = \mathrm{curl}$, where the curl denotes the component of the curl that is normal to the conductor surface. In section 3.2 we constructed K_c such that $K_c^{\mathrm{T}} P \approx I$ with P the discrete divergence operator. For the continuous analogy we require that $\mathcal{K}_c^{\mathrm{T}} \,\mathrm{div} = 1$, which does not specify a unique $\mathcal{K}_c$. We will use the notation $\mathcal{K}_c^{\mathrm{T}} = \mathrm{div}^+$ for this requirement. Using that the adjoint of the div is the $-\,\mathbf{grad}$ operator, as a result of integration by parts and the fact that Γ is a closed surface, we get

$$\mathcal{Q}^{\mathrm{T}} = \begin{pmatrix} \mathcal{K}_l^{\mathrm{T}} \\ \mathcal{K}_c^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} \mathrm{curl} \\ \mathrm{div}^{\mathrm{T}+} \end{pmatrix} = \begin{pmatrix} \mathrm{curl} \\ -\mathbf{grad}^{+} \end{pmatrix} \tag{3.45}$$

and

$$\mathcal{Q} = \begin{pmatrix} \mathcal{K}_l & \mathcal{K}_c \end{pmatrix} = \begin{pmatrix} \mathrm{curl}^{\mathrm{T}} & \mathrm{div}^{+} \end{pmatrix} \quad . \tag{3.46}$$

In order to rewrite $\mathrm{curl}^{\mathrm{T}}$, we first note that on the 2-dimensional surface, using some local Cartesian coordinates x and y,

$$\mathrm{curl}\,\mathbf{E} = \partial_x \mathrm{E}_y - \partial_y \mathrm{E}_x = -\,\mathrm{div}(\mathbf{n} \times \mathbf{E}) \quad , \tag{3.47}$$

where $\mathbf{n}$ is the outward normal vector on the surface Γ. This shows that the curl is actually the divergence of the rotated vector field. Using this we find that

$$\mathrm{curl}^{\mathrm{T}} = -\mathbf{n} \times \mathbf{grad} \quad . \tag{3.48}$$

3.3.1 Specification of div^+ and $\mathbf{grad}^+$

It is slightly more complicated to derive explicit expressions for div^+ and $\mathbf{grad}^+$. We first consider div^+. The complicating factor, is that the average divergence of a vector field on Γ is always zero :

$$\int_\Gamma \mathrm{div}\,\mathbf{v}(\mathbf{x}) d^2\mathbf{x} = -\int_\Gamma (\mathbf{grad}\ 1)\ \mathbf{v}(\mathbf{x}) d^2\mathbf{x} = 0 \quad , \tag{3.49}$$

where we used integration by parts to get the gradient of the constant function 1. This property (3.49) makes it impossible to satisfy the requirement $\operatorname{div}\operatorname{div}^+ u = u$ for all u, we therefore define div^+ by

$$\operatorname{div}\operatorname{div}^+ u = u - \overline{u} \ , \qquad \text{for all scalar fields } u \text{ on } \Gamma \ , \tag{3.50}$$

with $\overline{u}$ being the average of u over Γ. This corresponds precisely with equation (3.41) for the discrete case. Note that this definition does not specify a unique div^+. We can find div^+ using a Green function for div,

$$\operatorname{div}\boldsymbol{\Xi}(\mathbf{x}) = \delta(\mathbf{x}) - \epsilon \ , \tag{3.51}$$

with $1/\epsilon$ the area of Γ, such that

$$\operatorname{div}\int_\Gamma \boldsymbol{\Xi}(\mathbf{x}-\mathbf{x}')u(\mathbf{x}')d^2\mathbf{x}' = \int_\Gamma (\delta(\mathbf{x}-\mathbf{x}') - \epsilon)u(\mathbf{x}')d^2\mathbf{x}' = u(\mathbf{x}) - \epsilon\int_\Gamma u(\mathbf{x}')d^2\mathbf{x}' \ . \tag{3.52}$$

Given such a $\boldsymbol{\Xi}$, we can define div^+ as

$$(\operatorname{div}^+ u)(\mathbf{x}) = \int_\Gamma \boldsymbol{\Xi}(\mathbf{x}-\mathbf{x}')u(\mathbf{x}')d^2\mathbf{x}' \ . \tag{3.53}$$

We can write requirement (3.50) and result (3.52) in the form

$$\operatorname{div}\operatorname{div}^+ = \delta - \epsilon \ , \tag{3.54}$$

where we use the shorthand notation

$$((\delta-\epsilon)w)(\mathbf{x}) = \int_\Gamma (\delta(\mathbf{x}-\mathbf{x}') - \epsilon)w(\mathbf{x}')d^2\mathbf{x}' = w - \overline{w} \ . \tag{3.55}$$

We actually write the name of the kernel to indicate the corresponding integral operator.

Having defined div^+ we can find $\operatorname{div}^{+\mathrm{T}}$. Using

$$\begin{aligned}\langle \mathbf{v}, \operatorname{div}^+ u\rangle &= \int\!\!\int_\Gamma \mathbf{v}(\mathbf{x})\cdot\boldsymbol{\Xi}(\mathbf{x}-\mathbf{x}')u(\mathbf{x}')d^2x'd^2x = \\ &= \int\!\!\int_\Gamma \boldsymbol{\Xi}(\mathbf{x}-\mathbf{x}')\cdot\mathbf{v}(\mathbf{x})u(\mathbf{x}')d^2x'd^2x = \\ &= \langle \operatorname{div}^{+\mathrm{T}}\mathbf{v}, u\rangle \ ,\end{aligned} \tag{3.56}$$

we obtain

$$(\operatorname{div}^{+\mathrm{T}}\mathbf{v})(\mathbf{x}) = \int_\Gamma \boldsymbol{\Xi}(-(\mathbf{x}-\mathbf{x}'))\cdot\mathbf{v}(\mathbf{x}')d^2\mathbf{x}' \ . \tag{3.57}$$

With $\mathbf{grad} = -\operatorname{div}^{\mathrm{T}}$, this leads to

$$(\mathbf{grad}^+\,\mathbf{v})(\mathbf{x}) = \int_\Gamma -\boldsymbol{\Xi}(-(\mathbf{x}-\mathbf{x}'))\cdot\mathbf{v}(\mathbf{x}')d^2\mathbf{x}' \tag{3.58}$$

and

$$\mathbf{grad}^+\,\mathbf{grad} = \delta - \epsilon \ . \tag{3.59}$$

For explicit expressions, we still need to choose a $\boldsymbol{\Xi}(\mathbf{x})$ that fulfils requirement (3.51). This does not define a unique $\boldsymbol{\Xi}$, which reflects the fact that div^+ and $\mathbf{grad}^+$ are not unique. The solution space for $\boldsymbol{\Xi}$ depends on the geometry of Γ and cannot be expressed explicitly for general Γ. In section 3.3.3, we will investigate the case of the infinite flat conducting surface, and solve $\boldsymbol{\Xi}$ for that case. Note that we will use the same $\boldsymbol{\Xi}$ for both div^+ and $\mathbf{grad}^+$ to make sure that $\operatorname{div}^{+\mathrm{T}} = -\mathbf{grad}^+$.

3.3.2 The new operator

We can now apply the transformation to our integral operator $\mathcal{A}$ defined in (3.6). For ease of notation we will again write the name of the kernel to indicate the corresponding integral operator, such that

$$(\mathcal{G}w)(\mathbf{x}) = \int_\Gamma G(\mathbf{x},\mathbf{x}')w(\mathbf{x}')d^2\mathbf{x}' \ , \tag{3.60}$$

for both scalar and vector fields w. Using this notation, we can write

$$\mathcal{A} = \mathcal{C} + \mathcal{L} + \mathcal{R} = \frac{i}{\omega}\,\mathbf{grad}\ \mathcal{G}\ \operatorname{div} + \frac{i\omega}{c^2}\mathcal{G} + \mathcal{R} \ . \tag{3.61}$$

The transformed capacitance operator $\mathcal{C}$ decomposes into :

$$\begin{aligned}\mathcal{Q}^{\mathrm{T}}\mathcal{C}\mathcal{Q} &= \frac{i}{\omega}\begin{pmatrix} \operatorname{curl}\mathbf{grad}\,\mathcal{G}\,\operatorname{div}\operatorname{curl}^{\mathrm{T}} & \operatorname{curl}\mathbf{grad}\,\mathcal{G}\,\operatorname{div}\operatorname{div}^{+} \\ -\mathbf{grad}^{+}\mathbf{grad}\,\mathcal{G}\,\operatorname{div}\operatorname{curl}^{\mathrm{T}} & -\mathbf{grad}^{+}\mathbf{grad}\,\mathcal{G}\,\operatorname{div}\operatorname{div}^{+}\end{pmatrix} \\ &= \frac{i}{\omega}\begin{pmatrix} 0 & 0 \\ 0 & -(\delta-\epsilon)\mathcal{G}(\delta-\epsilon)\end{pmatrix} \ .\end{aligned} \tag{3.62}$$

All the zeroes follow from the observation that

$$\operatorname{curl}\mathbf{grad} = (\operatorname{div}\operatorname{curl}^{\mathrm{T}})^{\mathrm{T}} = 0 \ . \tag{3.63}$$

This shows that the transformation restricts the capacitive effects to the second diagonal block, just as in the discrete setting of equation (3.33). Furthermore, the non-zero block corresponds nicely to the discrete version in equation (3.42). For the inductive and resistive effects we find

$$\mathcal{Q}^{\mathrm{T}}(\mathcal{L}+\mathcal{R})\mathcal{Q} = \begin{pmatrix} \operatorname{curl}\,(\frac{i\omega}{c^2}\mathcal{G}+\mathcal{R})\,\operatorname{curl}^{\mathrm{T}} & \operatorname{curl}\,(\frac{i\omega}{c^2}\mathcal{G}+\mathcal{R})\,\operatorname{div}^{+} \\ -\mathbf{grad}^{+}\,(\frac{i\omega}{c^2}\mathcal{G}+\mathcal{R})\,\operatorname{curl}^{\mathrm{T}} & -\mathbf{grad}^{+}\,(\frac{i\omega}{c^2}\mathcal{G}+\mathcal{R})\,\operatorname{div}^{+}\end{pmatrix} \ , \tag{3.64}$$

which can, in general, not be further simplified. However, for the special case discussed in section 3.3.3, we will see that the off-diagonal blocks vanish.

3.3.3 Special case: the infinite plane conductor

In this special case, the conductor surface Γ is an infinite plane with constant surface impedance Z. The first simplification for this case is that the area of Γ is infinite, leading to $\epsilon = 0$. Also, we can explicitly write down all $\boldsymbol{\Xi}$:

$$\boldsymbol{\Xi}(\mathbf{x}) = \boldsymbol{\Xi}_0(\mathbf{x}) + \boldsymbol{\Xi}_1(\mathbf{x}) = \frac{\mathbf{x}}{2\pi|\mathbf{x}|^2} + \operatorname{curl}^{\mathrm{T}}\xi(\mathbf{x}) \ , \tag{3.65}$$

where ξ can be chosen freely, apart from some smoothness conditions. To see that this is correct, we compute the divergence of $\boldsymbol{\Xi}$. From (3.63) we see immediately that $\operatorname{div}\boldsymbol{\Xi}_1 = 0$. Some more computation shows that

$$\operatorname{div}\boldsymbol{\Xi}_0(\mathbf{x}) = \operatorname{div}\frac{\mathbf{x}}{2\pi|\mathbf{x}|^2} = 0 \qquad \forall\,\mathbf{x}\neq 0 \ , \tag{3.66}$$

and

$$\int_{|\mathbf{x}|<1} \operatorname{div} \boldsymbol{\Xi}_0(\mathbf{x}) d^2\mathbf{x} = \int_{|\mathbf{x}|=1} \boldsymbol{\Xi}_0(\mathbf{x}) \cdot \mathbf{x} dx = \int_{|\mathbf{x}|=1} \frac{1}{2\pi} dx = 1 \ . \tag{3.67}$$

All together, this shows that $\boldsymbol{\Xi}$ satisfies requirement (3.51) with $\epsilon = 0$,

$$\operatorname{div} \boldsymbol{\Xi}(\mathbf{x}) = \delta(\mathbf{x}) \ . \tag{3.68}$$

For this specific case, we can obtain the off-diagonal blocks of $\mathcal{Q}^{\mathrm{T}}\mathcal{A}\mathcal{Q}$ explicitly. From symmetry arguments we see that $\operatorname{curl} \boldsymbol{\Xi}_0 = 0$, so if we set $\xi = 0$, we get

$$(\mathcal{K}_l^{\mathrm{T}}\mathcal{R}\mathcal{K}_c u)(\mathbf{x}) = (\operatorname{curl}\mathcal{R}\operatorname{div}^+ u)(\mathbf{x}) = Z \iint_\Gamma \operatorname{curl} \boldsymbol{\Xi}(\mathbf{x}-\mathbf{x}') u(\mathbf{x}') d^2\mathbf{x}' = 0 \ , \tag{3.69}$$

giving $\mathcal{K}_l^{\mathrm{T}}\mathcal{R}\mathcal{K}_c = 0$. Still with $\xi = 0$, we also obtain

$$\begin{aligned}
(\mathcal{K}_l^{\mathrm{T}}\mathcal{L}\mathcal{K}_c u)(\mathbf{x}) &= (\operatorname{curl}\mathcal{G}\operatorname{div}^+ u)(\mathbf{x}) \\
&= \iint_\Gamma \operatorname{curl} G(\mathbf{x},\mathbf{x}') \ \boldsymbol{\Xi}(\mathbf{x}'-\mathbf{x}'') \ u(\mathbf{x}'') d^2x'' d^2x' \\
&= \iint_\Gamma \mathbf{n} \cdot \mathbf{grad}\, G(\mathbf{x},\mathbf{x}') \times \boldsymbol{\Xi}(\mathbf{x}'-\mathbf{x}'') \ u(\mathbf{x}'') d^2x'' d^2x' \\
&= \mathbf{n} \cdot \iint_\Gamma \frac{(ik|\mathbf{r}|-1)e^{ik|\mathbf{r}|}}{|\mathbf{r}|^3} \ \mathbf{r}\times\mathbf{r}' \ \frac{1}{2\pi|\mathbf{r}'|^2} \ u(\mathbf{x}'') d^2x'' d^2x' \ ,
\end{aligned} \tag{3.70}$$

where we used the abbreviations $\mathbf{r} = \mathbf{x} - \mathbf{x}'$ and $\mathbf{r}' = \mathbf{x}' - \mathbf{x}''$. We can now see that in the integrand, all terms are symmetric with respect to the line through $\mathbf{x}$ and $\mathbf{x}''$, except for the $\mathbf{r}\times\mathbf{r}'$ term, which is anti-symmetric. Because of this anti-symmetry, the integration over $\mathbf{x}'$ will give a zero result, and thus, $\mathcal{K}_l^{\mathrm{T}}\mathcal{L}\mathcal{K}_c = 0$.

Stated differently, we can see that the current $\boldsymbol{\Xi}_0(\mathbf{x})$ is symmetric with respect to all lines through the origin. This means that the same holds for the electric field due to this current, and in turn the rotation of this field is anti-symmetric with respect to these lines. This anti-symmetry requirement for the rotation, which is a scalar quantity in 2D, only allows zero rotation everywhere, showing that $\mathcal{K}_l^{\mathrm{T}}\mathcal{L}\mathcal{K}_c = 0$.

The above arguments show that $\mathcal{K}_l^{\mathrm{T}}\mathcal{A}\mathcal{K}_c = \mathcal{K}_c^{\mathrm{T}}\mathcal{A}\mathcal{K}_l = 0$ if $\xi = 0$. This means that the two diagonal blocks in A fully decouple, which is very nice since then we can analyse them separately and, what is even more interesting, we can precondition them separately. Note that this decoupling result does not require $\boldsymbol{\Xi}$ to be a Green function for the divergence, as defined in (3.51). It holds for all $\boldsymbol{\Xi}$ with the symmetry property. This is the reason for trying to approximate this symmetry property when choosing our discrete quasi-charge currents.

3.3.4 Fourier analysis on the infinite plane conductor

It is relatively easy to determine the Fourier transform of the operators for the infinite plane discussed in section 3.3.3. We get

$$\widehat{\mathcal{Q}} = \begin{pmatrix} -i\mathbf{n}\times\mathbf{q} & \widehat{\boldsymbol{\Xi}}(\mathbf{q}) \end{pmatrix} \tag{3.71}$$

and

$$\widehat{\mathcal{Q}}^{\mathrm{T}} = \begin{pmatrix} i(\mathbf{n}\times\mathbf{q})\cdot \\ \widehat{\boldsymbol{\Xi}}(-\mathbf{q})\cdot \end{pmatrix} \quad , \tag{3.72}$$

where the Fourier transform of $\boldsymbol{\Xi}$ is

$$\widehat{\boldsymbol{\Xi}}(\mathbf{q}) \equiv \int \boldsymbol{\Xi}(\mathbf{x}) e^{-i\mathbf{q}\cdot\mathbf{x}} d^2x = -i\frac{\mathbf{q}}{|\mathbf{q}|^2} - i\mathbf{n}\times\mathbf{q}\widehat{\xi}(\mathbf{q}) \quad , \tag{3.73}$$

assuming $\widehat{\xi}$ exists. The Fourier transform of $\mathcal{A}$ is given by

$$\widehat{\mathcal{A}} \equiv -\frac{i}{\omega}\widehat{G}(\mathbf{q})\mathbf{q}\mathbf{q}\cdot + \frac{i\omega}{c^2}\widehat{G}(\mathbf{q}) + Z \quad . \tag{3.74}$$

Writing the projected operator explicitly gives

$$\widehat{\mathcal{Q}}^{\mathrm{T}}\widehat{\mathcal{A}}\widehat{\mathcal{Q}} = \begin{pmatrix} \left(\frac{i\omega}{c^2}\widehat{G}(\mathbf{q}) + Z\right)|\mathbf{q}|^2 & \left(\frac{i\omega}{c^2}\widehat{G}(\mathbf{q}) + Z\right)|\mathbf{q}|^2\widehat{\xi}(\mathbf{q}) \\ \left(\frac{i\omega}{c^2}\widehat{G}(\mathbf{q}) + Z\right)|\mathbf{q}|^2\widehat{\xi}(-\mathbf{q}) & -\frac{i}{\omega}\widehat{G}(\mathbf{q}) + \left(\frac{i\omega}{c^2}\widehat{G}(\mathbf{q}) + Z\right)\left(\frac{1}{|\mathbf{q}|^2} + |\mathbf{q}|^2\widehat{\xi}(-\mathbf{q})\widehat{\xi}(\mathbf{q})\right) \end{pmatrix}, \tag{3.75}$$

and if we choose $\xi = 0$ (and hence $\widehat{\xi} = 0$), this reduces to

$$\widehat{\mathcal{A}}_Q = \begin{pmatrix} \left(\frac{i\omega}{c^2}\widehat{G}(\mathbf{q}) + Z\right)|\mathbf{q}|^2 & 0 \\ 0 & -\frac{i}{\omega}\widehat{G}(\mathbf{q}) + \left(\frac{i\omega}{c^2}\widehat{G}(\mathbf{q}) + Z\right)\frac{1}{|\mathbf{q}|^2} \end{pmatrix} \quad . \tag{3.76}$$

This confirms that the two diagonal blocks of $\mathcal{Q}^{\mathrm{T}}\mathcal{A}\mathcal{Q}$ are decoupled for this situation. Since this decoupling is very nice for both the analysis and practical preconditioning, we will use our freedom to choose ξ, and fix $\xi = 0$.

In equation (3.76) we see that $\widehat{\mathcal{A}}_Q$ is an ordinary diagonal 2×2 matrix, of which the elements depend on $\mathbf{q}$. This means that $\mathcal{A}_Q$ has the Fourier modes as eigenfunctions and the corresponding eigenvalues can be found on the diagonal of $\widehat{A}_Q$. The eigenvalues corresponding to the Fourier modes of the $\mathcal{K}_l$ block are thus given by

$$\lambda_l(\mathbf{q}) = \left(\frac{ik}{c}\widehat{G}(\mathbf{q}) + Z\right)|\mathbf{q}|^2 = \frac{2\pi}{c}|\mathbf{q}|^2\left(1 - \frac{|\mathbf{q}|^2}{k^2}\right)^{-\frac{1}{2}^*} + |\mathbf{q}|^2 Z \quad , \tag{3.77}$$

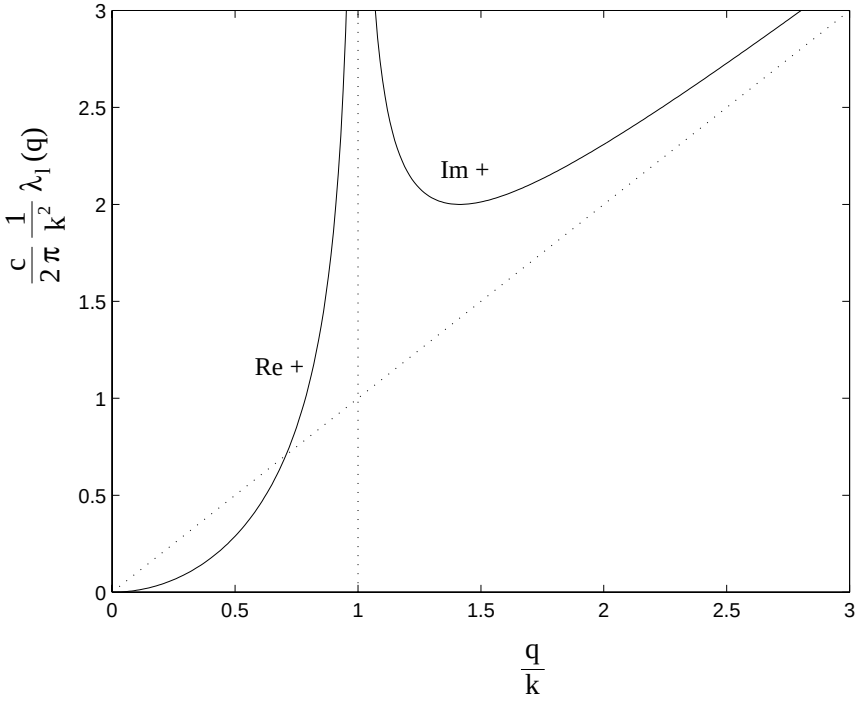
where we substituted $\omega = ck$ and used the expression (3.9) for $\widehat{G}(\mathbf{q})$. For the eigenvalues corresponding to the Fourier modes of the $\mathcal{K}_c$ block, we find

$$\lambda_c(\mathbf{q}) = -\frac{i}{kc}\widehat{G}(\mathbf{q}) + \left(\frac{ik}{c}\widehat{G}(\mathbf{q}) + Z\right)\frac{1}{|\mathbf{q}|^2} = \frac{2\pi}{c}\frac{1}{|\mathbf{q}|^2}\left(1 - \frac{|\mathbf{q}|^2}{k^2}\right)^{\frac{1}{2}^*} + \frac{1}{|\mathbf{q}|^2}Z \quad . \tag{3.78}$$

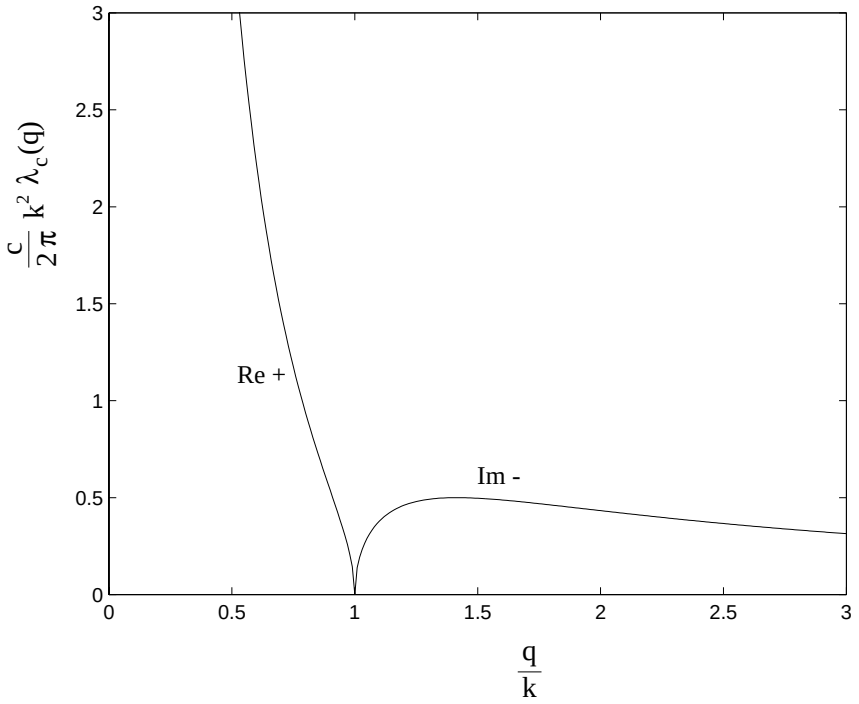
Note that these eigenvalues are closely related to the eigenvalues $\lambda_\perp$ in (3.15) and $\lambda_\parallel$ in (3.16) of the original operator $\mathcal{A}$:

$$\lambda_l(\mathbf{q}) = |\mathbf{q}|^2\lambda_\perp(\mathbf{q}) \qquad \text{and} \qquad \lambda_c(\mathbf{q}) = \frac{1}{|\mathbf{q}|^2}\lambda_\parallel(\mathbf{q}) \quad . \tag{3.79}$$

These relations are not surprising, if one remembers the two extra differential operators involved in λ_l and the two extra inverses of differential operators involved in λ_c, giving the factors $|\mathbf{q}|^2$ and $1/|\mathbf{q}|^2$ respectively. These extra factors reverse the short wavelength ($\mathbf{q}$ large) behaviour of the capacitive and inductive eigenvectors. The eigenvalues of the transformed system can be seen in Figure 3.6.



(a) Inductive modes



(b) Capacitive modes

FIGURE 3.6: The scaled continuous eigenvalues of $\mathcal{A}_Q$ for the inductive (λ_l) and capacitive (λ_c) modes on the infinite plane conductor in absence of resistance ($Z = 0$). The absolute value is shown while the text shows on which complex axis the values are.

3.3.5 Discussion of the Fourier analysis

Having done the Fourier analysis, the question is how to interpret the results. Are the properties of the transformed system better than those of the original one? We will restrict ourselves to the effect of the basis transformation on the application of iterative solvers. For some more background knowledge on iterative Krylov subspace solvers, see section 1.2.

Let us first compare the eigenvalues for the infinite plane, shown in Figures 3.2 and 3.6. For the highly oscillating part of the spectrum ($|\mathbf{q}| \gg k$), we see that the $|\mathbf{q}|$ and the $1/|\mathbf{q}|$ behaviour switched between the inductive parts ($\lambda_\perp$ and λ_l) and the capacitive parts ($\lambda_\parallel$ and λ_c). Both types of behaviour are present in both cases, so in this respect there was no change.

For the smooth part of the spectrum ($|\mathbf{q}| \ll l$), we had nice behaviour in the original system, where both $\lambda_\perp$ and $\lambda_\parallel$ tended to a constant. However, for the transformed system, λ_l behaves like $1/|\mathbf{q}|^2$ while λ_c behaves like $|\mathbf{q}|^2$, which shows that the transformed matrix A_Q may have very large or very small eigenvalues. This can slow down the convergence of an iterative solver, which makes this a change for the worse. In practice however, this part of the spectrum is less important. For a finite and discretised surface, the spectrum becomes discrete and finite, as described in section 3.1.2. On a large and finely discretised surface, we can expect most of the eigenfunctions and values to be close to the continuous spectrum, but now for a discrete set of $\mathbf{q}$-values approximately uniformly distributed over the $\mathbf{q}$ plane. However, eigenfunctions with too high $|\mathbf{q}|$ values cannot be represented by the discretisation, while eigenfunctions with too low $|\mathbf{q}|$ values will not fit on the finite size conductor, leading to a lower and upper cutoff of the spectrum, as we already mentioned in section 3.1.2. The approximately uniform distribution of $\mathbf{q}$ values between these cutoffs will lead to a density of $|\mathbf{q}|$ values proportional to $|\mathbf{q}|^2$, due to the fact that the area in the $\mathbf{q}$ plane with $q < |\mathbf{q}| < q + \Delta q$ is proportional to q^2. This shows that there are only a few smooth eigenfunctions. In this context of counting eigenfunctions, we may say that the smooth part of the spectrum is less relevant. If, for instance, the wavelength $\lambda = (2\pi k)^{-1}$ is larger than twice the size of the surface, we may expect to find no eigenvalues from the $|\mathbf{q}| < k$ region. On the other hand, if the problem frequency is increasing, we may expect an increasing number of eigenvalues in the $|\mathbf{q}| < k$ region. This may lead to convergence problems for an iterative solver due to very large and very small eigenvalues. We thus expect that this transformation will not be a good idea for very high frequency problems in which λ is much smaller than the size of the conductor.

However, the initial motivation for the transformation, to separate the inductive effects from the capacitive ones, was successful, and will lead to effective preconditioners (see chapter 4). In the continuous operator case on the infinite plane, we even decoupled the two blocks.

3.4 Properties of the transformed matrix

In the previous section we did Fourier analysis of the continuous analogue of the transformed system for the infinite plan conductor. We still have to address the question

whether this Fourier analysis gives us useful information for the discretised operator on a more complex and finite geometry.

First we have to go from the infinite plane to some finite geometry, leading to a discrete spectrum, allowing only modes that fit the geometry. How the spectrum is influenced depends on the oscillation length of the eigenmode $\lambda_{\mathbf{q}} = (2\pi|\mathbf{q}|)^{-1}$, compared to the typical size of the geometry L. This introduces a notion of large and small $\mathbf{q}$ that is independent of k.

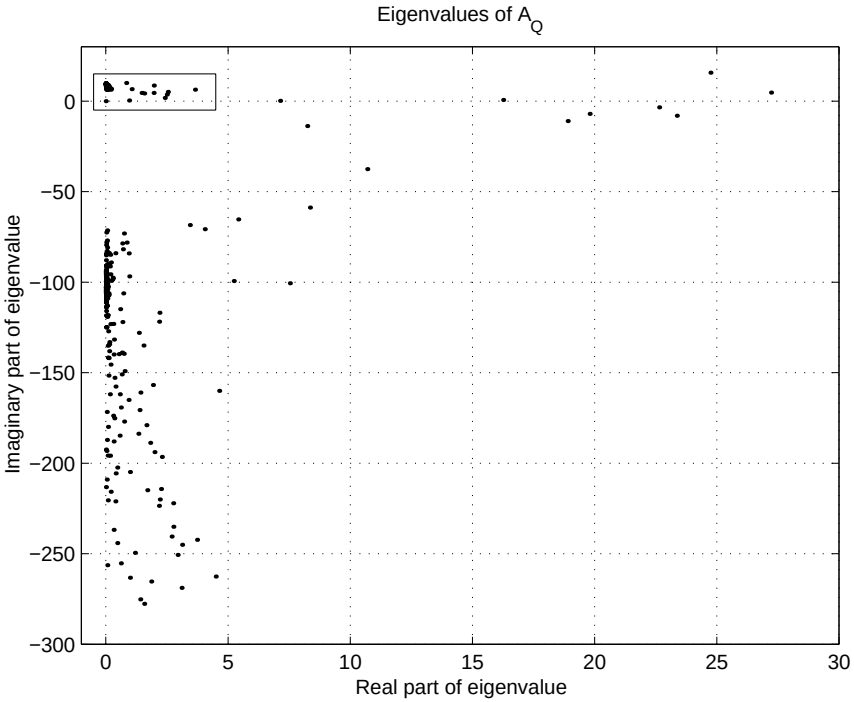
Since the highly oscillating modes ($\lambda_{\mathbf{q}} \ll L$ or $L|\mathbf{q}| \gg 1$) are mostly determined by the local interactions, they will not be influenced very much by a change in geometry. We may also expect the decoupling of the K_l and K_c blocks to be approximately true for this high $\mathbf{q}$ region. The allowed values of $\mathbf{q}$ will be spaced with steps of approximately $(\pi L)^{-1}$, every step adding one half oscillation at the boundary.

The low wavenumber modes ($\lambda_{\mathbf{q}} \not\ll L$ or $L|\mathbf{q}| \not\gg 1$) will be more strongly dependent on the geometry. This will make the low $\mathbf{q}$ part of the Fourier spectrum obtained for the infinite board, unreliable for finite geometries. Based on the $\mathbf{q}$ value spacing of approximately $(\pi L)^{-1}$, there will be only a few very low wavenumber modes. Where exactly and how fast the transition between the good approximation of the high $\mathbf{q}$ modes and the bad approximation of the low $\mathbf{q}$ modes occurs, is not clear and this will again depend on the geometry.

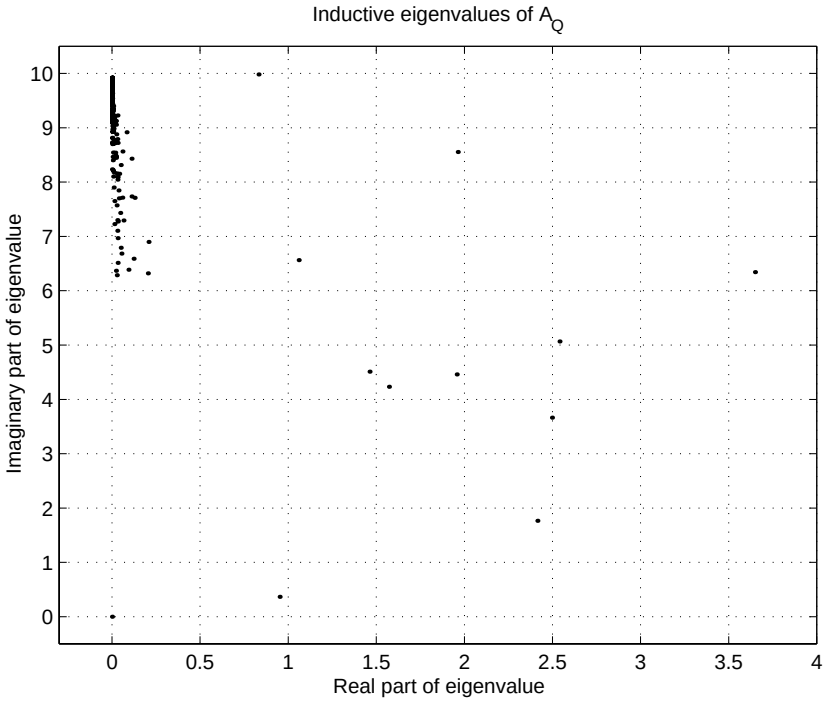
Next we introduce the discretisation. As a result, the very high $\mathbf{q}$ modes cannot be represented any more. This results in a cutoff for the spectrum at the high $\mathbf{q}$ end. The cutoff value will be determined by the grid size h and the type of discretisation. In general, the cutoff will be around $\lambda_{\mathbf{q}} \approx 2h$ or equivalently $|\mathbf{q}| \approx (4\pi h)^{-1}$, giving yet another measure of relative size for $\mathbf{q}$. For smooth modes ($h|\mathbf{q}| \ll 1$), the discretisation will be very good and will thus change little with respect to the continuous case. For the non-smooth modes ($h|\mathbf{q}| \not\ll 1$), the discretisation will not be very accurate and will have effect on the eigenvalues and eigenvectors. However, for reasonably regular discretisations the general behaviour will be preserved.

One further difference between the discrete A_Q and the continuous $\mathcal{A}_Q$ is, that we approximated the div^+ operator with local quasi-charge currents for $\boldsymbol{\Xi}$. This will affect the low $\mathbf{q}$ modes (with $\lambda_{\mathbf{q}}$ large compared to the cutoff radius of the approximate $\boldsymbol{\Xi}$, which we chose $2h$), but will be less important for the high $\mathbf{q}$ modes. We tried to preserve the symmetry properties of the truncated $\boldsymbol{\Xi}$, which are responsible for the decoupling of the K_l and K_c blocks in the continuous infinite plane conductor case. It is impossible to maintain the continuous symmetry for the discrete case, but we hope that the slight deviations from symmetry in our choice for the quasi-charge current shown in Figure 3.5(b), will only result in weak couplings.

In order to find out what really happens, we have transformed the matrix A that we used for Figure 3.4(a), and computed the eigenvalues of A_Q. The spectrum of A_Q is shown in Figure 3.7(a). The eigenvalues corresponding to inductive effects are contained in the box and are shown in the enlargement (Figure 3.7(b)). Note that we have one zero eigenvalue due to the over-complete K_c basis. The eigenvectors corresponding to the inductive eigenvalues in Figure 3.7(b) "live" mostly in the K_l part of the space, while the other eigenvectors "live" mainly in the other part. This separation is very good for the high $\mathbf{q}$ modes and a little less for a few low $\mathbf{q}$ modes. To illustrate this, in Figure 3.8(a) we have plotted a histogram of the angles between the eigenvectors and the subspace

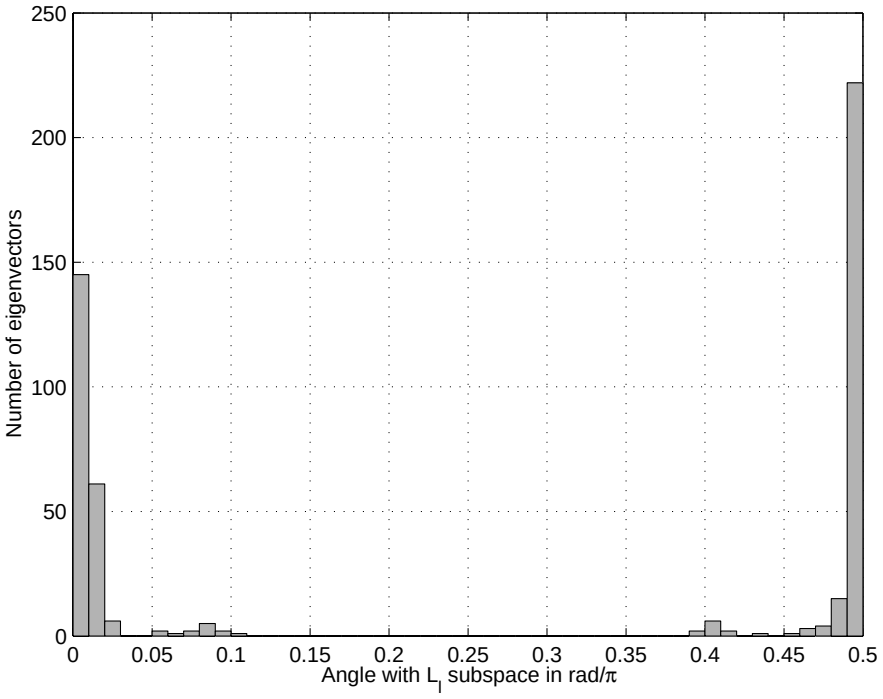


(a) All the eigenvalues.

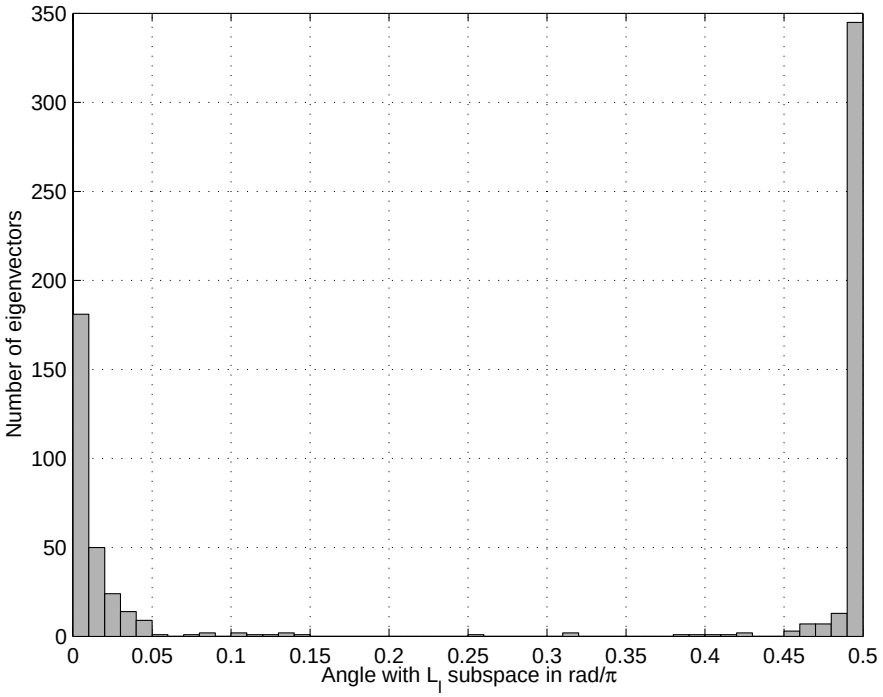


(b) The inductive part of the spectrum (λ_l), from the box in subfigure (a). Only the zero eigenvalue is due to the capacitive part.

FIGURE 3.7: The eigenvalues of A_Q for a square board of 1×1 meter at 200 MHz ($\lambda = 1.5$ meter), discretised using 16×16 squares.



(a) For a square board of 1×1 meter at 200 MHz ($\lambda = 1.5$ meter), discretised using 16×16 squares.



(b) For a more complex geometry of 2 boards (approximately 1×1 meter) and 3 wires at 200 MHz ($\lambda = 1.5$ meter).

FIGURE 3.8: Histograms of the angles the eigenvectors of A_Q make with the subspace related to the K_l block.

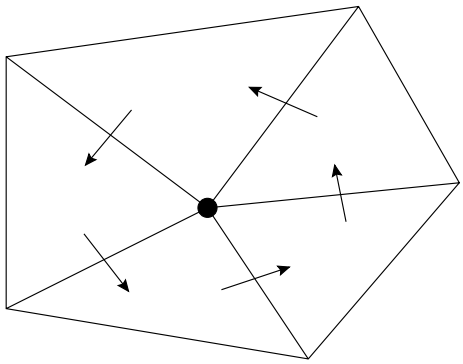


FIGURE 3.9: A local loop current in a triangular discretisation. The internal vertex is marked with a •, the currents crossing the connected edges are shown with the arrows.

related to the K_l block. We clearly see a separation between the inductive (small angle) and capacitive (large angle) modes. This splitting corresponds to the splitting of the eigenvalues shown in Figure 3.7. This confirms that we still have an effective decoupling for the high $\mathbf{q}$ modes, just as we expected. As we have already mentioned, this will be important for preconditioning later on. The decoupling means that we can approximate A_Q (and its inverse) by considering only the two diagonal blocks. To see whether this still holds for more complicated geometries, we also plotted a histogram for a more complicated geometry in Figure 3.8(b). This shows that the separation is a little less strict here, but still almost all eigenvectors "live" either on the K_l or the K_c subspace.

The scaling of the eigenvalues in the Fourier analysis does not correspond to the scaling of the eigenvalues in the discretised system. This is due to a different scaling of the columns of Q. To obtain a true approximation for the rotation in $\mathcal{Q}$, we should have scaled this part of the basis with a factor of h^2. Since this scaling would make the transformation Q ill-conditioned, we have chosen a scaling with a norm of order 1.

3.5 Implementation details

For the first part of our new basis K_l, we have to find all independent current loops in the discretisation. In general, around each internal vertex in the grid there is a local loop current that can be represented by the currents that flow through the edges that are connected to the vertex, as is shown in Figure 3.9. In this way we find almost all divergence free currents. This part of K_l will have on average 6 or 4 elements per column for triangular or quadrilateral discretisation respectively. Note that the wires are modelled as 1-dimensional structures and therefore they cannot have local loop currents.

We still need to find the independent global loops that arise due to holes in the geometry. A very simple example would be a closed wire where the one global current loop is obvious. In real applications, the geometry will consist of a number of conducting plates and many wires connecting the plates in different ways, creating various global loops. A very simple example of such a structure can be seen in Figure 4.5. This structure has one global loop, passing through the two boards and the two connecting wires. For this simple geometry, the global loop is easily found, but this becomes more

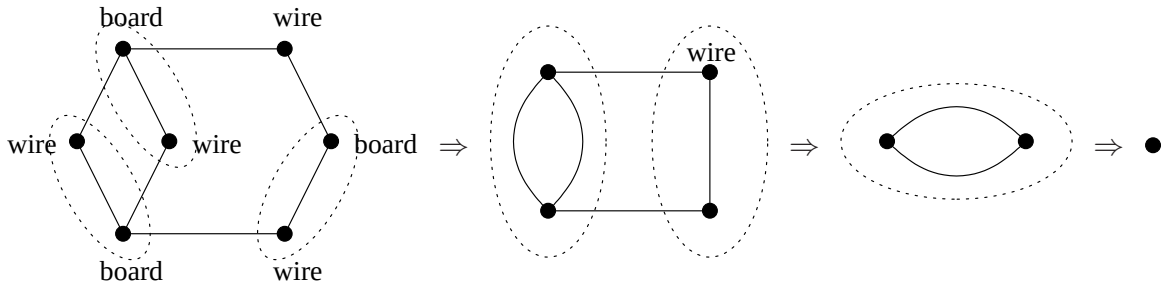


FIGURE 3.10: Example of global loop detection using graph reduction. The initial graph contains three boards and four wires. In each step the nodes in the dotted ellipses are merged, until we end up with one node only. In the second reduction step we find the smaller left loop of the initial graph, and in the third reduction step we find the larger loop.

complicated for more complicated geometries. In order to let the computer find all the global loops, we use the following method.

The structure of the model is represented by a graph, each node representing a part of the conductor for which all independent loops are already listed. Initially, a node represents for instance a single board or wire. The loops in the graph will represent the global loops that are not listed yet. We now take 2 connected nodes of the graph, and merge them into one node. In order to make sure that all the loops in the new node are listed, we have to add the loops resulting from multiple connections between the 2 nodes to the list. Turning these graph loops into current loops requires some bookkeeping of how these conductor parts are connected and how we can transport charge trough these parts, from the one to the other connection. By doing this, we reduced the graph by one node. By repeating this, we can reduce the graph to one point, and we are done since all loops are listed. This method of finding the global loops is very efficient and is not expensive in terms of computing time. The disadvantage of the method is that the loops that we find are not necessarily the shortest ones, which can lead to more elements in K_l than are needed. To reduce the occurrence of unnecessary long loops, it is best to start by merging nodes that are physically small and close together. This leads to a strategy where, in each step, all nodes are grouped in pairs after which each pair is merged. An example graph reduction is shown in Figure 3.10. There are probably many more strategies that might work even better, but this has worked for us in our examples.

In principle, we can apply this method for the local loops as well. However, since the graph reduction method will not necessarily give the shortest loops, this will lead to much more elements in K_l. This will also distort the idea that the local loops in K_l represent a rotation operator. On top of that, it is easier to find the local loops using the nodes in the discretisation, than using the graph method at this level.

For the K_c part of the basis, we have to define a quasi-charge current for each element in the discretisation. The quasi-charge currents that we have used on the interior of our domains are shown in Figure 3.5. They are generated using the idea of taking a little charge from each element in some region around the central element, and using a current to move this charge to the central element via the shortest path. Putting all these currents together and using the remaining freedom to keep symmetry, we found

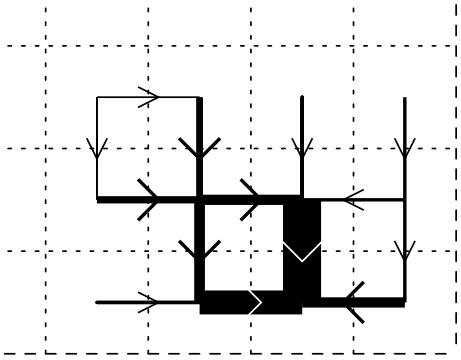


FIGURE 3.11: A quasi-charge current near the edge of a board. The dashed lines denote the board edge, the dotted lines denote the mesh, and the arrows denote the current (thickness is proportional to the current).

the currents shown in Figure 3.5. At the boundary, we use the same scheme, but now the region from which charge is moved to the central element is cut off by the boundary. An example of such a boundary quasi-charge current is shown in Figure 3.11. Note that the resulting current will not be a cut off version of the interior currents, but this will be true for the the divergence of the current. This scheme can also be generalised for different discretisations, although the preservation of symmetry might be more difficult for irregular discretisations.

At places where two components are connected, like a connection between a wire and a plate, we have to extend the charge region to the other component in order to allow a current between the two components. This is implemented by first creating the quasi-charge currents for the separate components, and then mixing the two quasi-charge currents of the connected boundary elements. We will illustrate this with an example of joining two wires, shown in Figure 3.12. On the left, the two initial quasi-charge currents are shown. Let these be $\mathbf{J}_1$ and $\mathbf{J}_2$ respectively. If we use $\mathbf{J}_{12}$ to denote a unit current from the left wire to the right wire, we can combine these to get the two new quasi-charge currents :

$$\widetilde{\mathbf{J}}_1 = \frac{k}{k+\ell}\mathbf{J}_1 + \frac{\ell}{k+\ell}\mathbf{J}_2 - \frac{\ell}{k+\ell}\mathbf{J}_{12} \tag{3.80a}$$

$$\widetilde{\mathbf{J}}_2 = \frac{k}{k+\ell}\mathbf{J}_1 + \frac{\ell}{k+\ell}\mathbf{J}_2 + \frac{k}{k+\ell}\mathbf{J}_{12} \ . \tag{3.80b}$$

$\widetilde{\mathbf{J}}_2$ is shown on the right in Figure 3.12. This way of combining the two currents makes sure that new quasi-charge currents have equal charge for all elements (except the central element). However, this does not mean that the corresponding charge density is constant over the region, since the element sizes can be different on the two components. It is still an open question how to connect two components in an optimal fashion, especially when the two components are of a different nature, like a 2-dimensional plate and an essentially 1-dimensional wire. One might expect that the capacitance of the elements

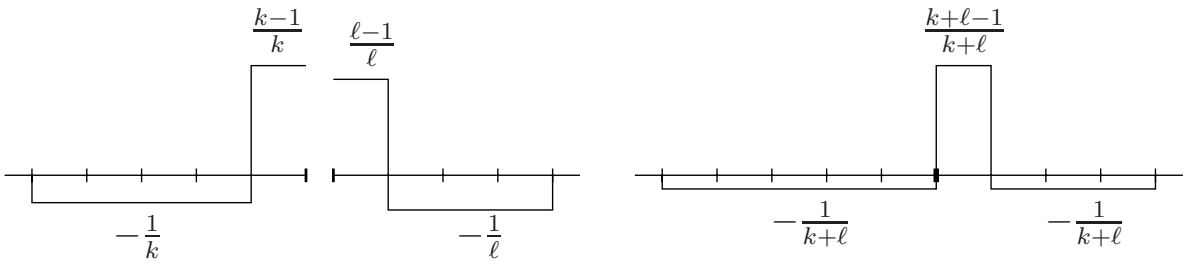


FIGURE 3.12: Example of joining two quasi-charge currents for two connected wires. On the left the charge for the separate edge quasi-charge currents. On the right the charge for one of the joint quasi-charge currents. For this example, $k = 5$ and $\ell = 4$.

should be used to get a smooth electrostatic potential from the charge distribution. We have not done any experiments in this direction.

The size of the region used for the quasi-charge current is still a parameter that has to be chosen. Large values will give accurate approximations of $P^{\mathrm{T}}K_c = I$, but will lead to relatively dense K_c, while small values have the opposite effect. The compromise that worked for us is a 5×5 square region on the plates and 11 elements on the wires.

Chapter 4

Geometric multigrid

With the basis transformation proposed in the previous chapter, we have separated the inductive effects from the capacitive effects. Furthermore, we have shown that in the resulting 2×2 block matrix A_Q, the diagonal blocks are approximately decoupled for the highly oscillating modes. Using this information, we can make a preconditioner for this matrix using the approximate inverse of the two diagonal blocks given by some standard preconditioner. In this way we obtain a preconditioner that works nicely for the fast oscillating modes (see section 4.2). However, we cannot expect to get good results for the slowly oscillating modes as well, which will make the effectiveness of the preconditioner strongly dependent on the mesh size. We will attempt to remove this mesh dependence with a multigrid preconditioner for A_Q.

4.1 Multigrid in a nutshell

Multigrid is described extensively by Hackbusch [21] and many others (for introductions see, for instance, [45] and [12]). We will give a very short introduction to the subject here.

Multigrid is a method to solve a finely discretised problem, by using several levels of coarser discretisations for that same problem. The basic idea of multigrid is to use a projection of the fine grid problem on a coarser grid to remove the slowly varying components from the error, the so-called coarse grid correction. The quickly oscillating components in the error are then removed with an iterative technique, which is called the smoother. For standard Poisson-like elliptic problems, a few steps of the damped Jacobi or Gauss-Seidel iteration will do the job. By applying both corrections in an iterative fashion, the error is reduced. In order for this to work the coarse grid correction and the smoother must be each others complement, in the sense that modes that are not damped by the smoother are damped by the coarse grid correction and vice versa.

The coarse grid correction uses a restriction operator R, to restricts a fine grid residual to the next coarser grid. Also, an interpolation operator P is used, to interpolate a coarser grid error to the fine grid. Suppose we have to solve $A\bar{x} = b$ and have some iterate x, we project the residual $r = b - Ax$ to the coarse grid using the restriction operator to get the coarse grid residual $r_c = Rr$. Then, we (approximately) solve the coarse grid system $A_c e_c = r_c$, where A_c is the representation of A on the coarse grid. Since the dimensions

of this coarse grid problem are small, this is much cheaper than solving the equations on the fine grid. Next, we prolongate the coarse grid error e_c back to the fine grid an use this to update the iterate, $x = x + Pe_c$.

The coarse grid matrix can be formed by discretising the equations on the coarse grid. In this case the restriction and prolongation matrices should be chosen to match this coarse discretisation, in the sense that A_c should be an approximate projection of A, $A_c \approx RAP$. In this way the sparsity and other properties of A_c are comparable to those of A, and A_c can usually be computed cheaply. We will, however, form the coarse matrix by explicitly projecting the fine grid matrix, $A_c = RAP$. If A is sparse, this will in general not preserve the level of sparsity for A_c. In our context, A is already dense, so preserving sparsity is not relevant. Furthermore, calculating the matrix A_c as a new discretisation of the equations, would have been expensive. Calculating A_c by projecting the matrix A is much cheaper. An extra advantage is that we do not have to worry about how good our R and P are with respect to approximating $A_c \approx RAP$, since $A_c = RAP$ by construction.

For the coarse grid correction, we have to solve the projected coarse grid problem $A_c e_c = r_c$. This is a smaller system, but it can be still to big to solve directly, in which case we apply the same strategy to approximately solve this coarse grid problem, using a smoother for the coarse grid and an even coarser grid for the coarse grid correction. This strategy can be recursively applied, leading to the use of multiple levels of coarser grids, hence the name multigrid. The recursion has to stop at some level, at which the problem is solved directly.

At each level we combine the smoother and coarse grid correction in 3 steps. The recursive application of these three steps on each level is called a V-cycle:

- **Pre-smoothing**: The high-frequency modes are damped by the smoother M. This can be repeated α times.

$$\begin{aligned} &\alpha \text{ times} : x \leftarrow x + Mr \\ &r \leftarrow (1 - AM)^{\alpha} r \\ &e \leftarrow (1 - MA)^{\alpha} e \end{aligned} \tag{4.1}$$

- **Coarse grid correction**: Now the residual is restricted to the coarser grid ($r_c = Rr$). The smaller coarse grid system is inverted approximately ($e_c = A_c^{\sim 1} r_c \approx A_c^{-1} r_c$) with the same three steps on the coarser level. The solution is interpolated to the finer grid for the correction of the previous solution :

$$\begin{aligned} &x \leftarrow x + P\ A_c^{\sim 1}\ Rr \\ &r \leftarrow (1 - A\ P\ A_c^{\sim 1}\ R) r \\ &e \leftarrow (1 - P\ A_c^{\sim 1}\ R\ A) e \end{aligned} \tag{4.2}$$

- **Post-smoothing**: The high frequency modes are again removed using a number of smoothing steps :

$$\begin{aligned} &\beta \text{ times} : x \leftarrow x + Mr \\ &r \leftarrow (1 - AM)^{\beta} r \\ &e \leftarrow (1 - MA)^{\beta} e \end{aligned} \tag{4.3}$$

Note that the application of the V-cycle requires for every level $\alpha + \beta + 1$ matrix-vector multiplications, $\alpha + \beta$ applications of the preconditioner M, and one restriction and prolongation.

A classical multigrid solver will now repeatedly apply the V-cycle, reducing the error in every step and thus converging to the solution. This will only work if *all* modes are damped by the V-cycle, and the convergence speed then depends on the worst damped mode. Since this V-cycle is designed to reduce the error, we can also use it as an approximation for A^{-1}, and use it as a preconditioner in an iterative solver (see section 1.2). By using a Krylov subspace solver, we can reduce the effect of single modes that are damped poorly by the V-cycle, and obtain a converging method even if not all modes are damped by the V-cycle. This leads to a more robust method. If the V-cycle is very effective, then classical multigrid already converges very quickly, and in that case the Krylov subspace solver will contribute almost nothing to the convergence speed and only introduce some extra overhead, but this will be marginal because of the fast convergence.

4.2 Smoother

This section will be devoted to the smoother part of multigrid. As we described above, we have to find a smoother for our system that will damp the highly oscillating modes.

As we have argued at the beginning of section 3.2, it is very difficult to construct a good preconditioner for the original matrix A. This was the motivation for devising the basis transformation Q, such that we have to solve a linear system with the matrix $A_Q = Q^{\mathrm{T}}AQ$. In section 3.4, we found that for this new A_Q, the capacitive terms are separated from the important inductive terms. We also found the nice side effect that the two diagonal block of A_Q are approximately decoupled for the highly oscillating modes. For the smoother, we are only after these highly oscillating modes, so we can consider the two diagonal blocks as being decoupled and use a two block diagonal smoother. This implies that we can make a smoother for the two diagonal blocks independently, and then put them together.

Most standard preconditioning and smoothing techniques are designed for sparse matrices that stem from elliptic problems. Unfortunately, our diagonal blocks are dense and directly applying these techniques would be very expensive. This is one reason to use only a sparse subset of the elements of the matrix. Another reason is that we do not know A_Q explicitly. If we wanted to use all elements of the A_Q diagonal blocks for the construction the smoother, we would have to calculate $A_Q = Q^{\mathrm{T}}AQ$ explicitly. This would have been expensive due to the dense nature of A and A_Q. For more details on implementation, see section 4.4.1.

Of course, we want to calculate the important elements in A_Q only. A first heuristic to decide on what is important is to monitor the size of the elements: keep the large ones and forget about the small ones. This is hard, since we do not have the elements of A_Q explicitly. However, we know that interactions between nearby elements are physically stronger than long range interactions. This leads to a different heuristic: keep the close range interaction elements and drop the long distance ones. This heuristic is easy to implement and we will see that it has some nice theoretical properties as well. The sparsified version of A_Q, using this distance criterion, will be called A_Q^{sp}.

In the sparsification, we ignore the terms that are important for the long range behaviour. Hence we cannot expect to get a good preconditioner for long range effects, if it is based on the sparsified matrix. For the highly oscillating eigenfunctions, the resulting long distance field will be negligible because of the cancellation. For these highly oscillating eigenfunctions the near interaction terms are the most important ones and they are represented well by the sparsification. This means that we can still construct a good preconditioner for the removal of the highly oscillating modes, based on the sparsified matrix. These modes correspond to the extreme eigenvalues that caused the bad condition (3.31) for low source frequencies. To make this more precise, we will again look at the continuous operators on the infinite domain in section 4.2.2. The analysis will confirm the hand-waving arguments that we have used above.

In the ideal case, we would like to use the inverse of the sparsified A_Q as a smoother, but this is too expensive. As an alternative we have to use approximations in the form of some standard preconditioning/relaxation scheme. We have used the sparsified matrix for the standard point relaxation methods Jacobi and Gauss-Seidel (GS). Jacobi smoothing performed rather poorly, but might be improved by a suitable damping scheme. The triangular solve for Gauss-Seidel becomes instable for thin, low resistance wires when the frequency is too high. The critical point appeared to be approximately $\lambda/h \approx 12$. To overcome these problems, we have mainly used a sparse approximate inverse method using Frobenius norm minimisation, described in the next section.

4.2.1 Frobenius norm minimisation

Frobenius norm minimisation can be used to construct an approximation of the inverse of a matrix. To get an approximation M of the inverse of a sparse matrix A, we minimise

$$\min_M \|AM - I\|_F^2 \quad . \tag{4.4}$$

If there are no restrictions on M, then $M = A^{-1}$ will be the minimiser with zero residual, and if the residual is small, the matrix M must, in some sense, approximate A^{-1}. By restricting the sparsity pattern that is allowed for M to some pattern given by a matrix S,

$$S_{jk} = 0 \quad \Rightarrow \quad M_{jk} = 0 \quad , \tag{4.5}$$

this minimisation can be performed at relative low computational cost. In order to see this, we write the Frobenius norm in (4.4) explicitly :

$$\|AM - I\|_F^2 = \sum_k \left(\sum_i \left| \sum_j A_{ij} M_{jk} - I_{ik} \right|^2 \right) \quad . \tag{4.6}$$

Since both A and M are sparse, a lot of the products $A_{ij}M_{jk}$ will be zero and need not be considered while minimising. To use this, we introduce the index sets

$$J_k = \{j | S_{jk} \neq 0\} \tag{4.7}$$

$$I_k = \{i | \exists_{j \in J_k} A_{ij} \neq 0\} \quad , \tag{4.8}$$

both of which will have only a few elements due to the sparsity of A and M. The index sets can be used in (4.6)

$$\begin{aligned} \|AM - I\|_F^2 &= \sum_k \left(\sum_{i \in I_k} \left| \sum_{j \in J_k} A_{ij} M_{jk} - I_{ik} \right|^2 \right) \\ &= \sum_k \|A(I_k, J_k) M(J_k, k) - I(I_k, k)\|^2 \ , \end{aligned} \tag{4.9}$$

in which $A(I_k, J_k)$ is the sub-matrix of A corresponding to the two index sets and $M(J_k, k)$ is a subset of the kth column of A. To minimise $\|AM - I\|_F$ is thus equivalent to minimising

$$min_{M(J_k,k)} \|A(I_k, J_k) M(J_k, k) - I(I_k, k)\| \tag{4.10}$$

for each k. These are all small linear least squares problems, and can be solved using standard methods. Since all these small linear least squares problems are independent, this can easily be solved in parallel. As was shown by Grote and Huckle [20], this scheme can be extended to be able to automatically find a suitable sparsity pattern for M, which can be very useful if there is no suitable a priori pattern available, but it also make the method more time consuming.

We use this Frobenius norm minimisation to construct a sparse approximate inverse M_Q for A_Q^{sp}. For this problem, we expect that also for the inverse, the short range terms are the most important, so we fix the sparsity pattern of M_Q to be equal to the sparsity pattern of A_Q^{sp}. We will refer to this as the sparse approximate inverse (SAI) in the remainder of this thesis.

4.2.2 Preconditioning with truncated interaction

We will now discuss the consequences of the sparsification for the smoother if further detail. We sparsify A_Q by dropping all elements except those that correspond to interactions between physically nearby elements. We still expect that the sparsified matrix A_Q^{sp} is a good approximation to the original A_Q for the highly oscillating eigenfunctions, and also $(A_Q^{\text{sp}})^{-1} \approx A_Q^{-1}$ for these functions. We will verify this for the case of the infinitely large flat plane conductor used in section 3.1.1 and section 3.3.3.

We first consider the results of truncating the Green function (2.15) in the original EFIE operators in equation (2.16). This would be similar to truncating the discretised interaction, which would correspond to sparsifying the original matrix A. We consider the operator $\mathcal{A}_T$, which is the same as $\mathcal{A}$ (3.6) but with the Green function truncated at distance R_T. We can repeat the analysis from section 3.1.1 for this operator, but now replacing the true Green function by the truncated Green function

$$G_T(\mathbf{y}) = \begin{cases} G(\mathbf{y}) & \text{for} \quad |\mathbf{y}| \leq R_T \\ 0 & \text{for} \quad |\mathbf{y}| > R_T \end{cases} \ . \tag{4.11}$$

This means that the eigenfunctions of the truncated operator $\mathcal{A}_T$ are the same as those found for $\mathcal{A}$ in section 3.1.1, but the eigenvalues are different. To measure how well the truncated operator corresponds to the real operator, we will consider the original operator preconditioned with the inverse of the truncated operator :

$$\mathcal{A}\mathcal{A}_T^{-1} \ . \tag{4.12}$$

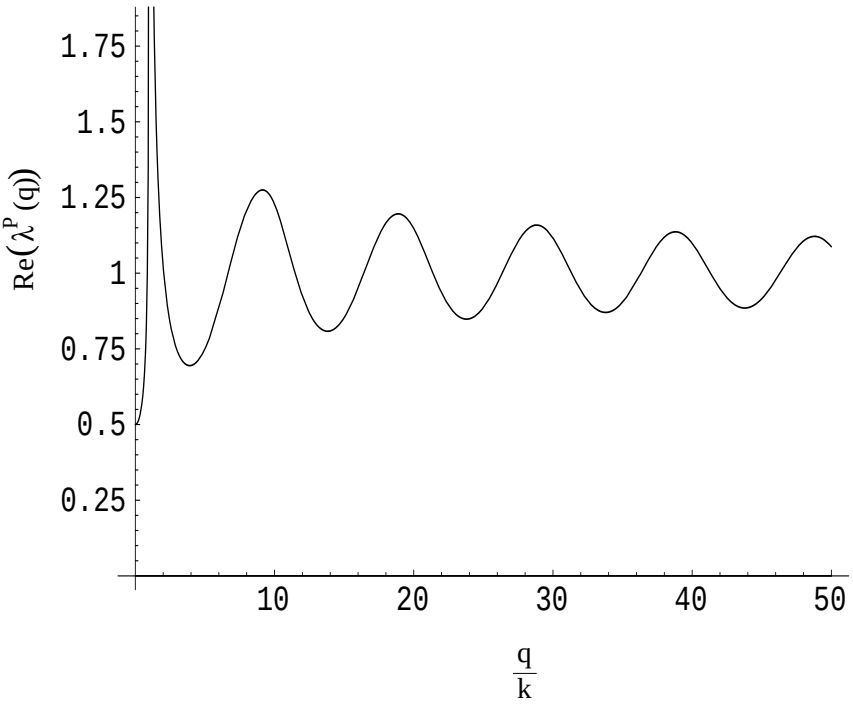


FIGURE 4.1: The continuous eigenvalues of the preconditioned operators ($\lambda^P(\mathbf{q})$) in absence of resistance, using a truncation distance of $R_T = 0.1\lambda$. Only the real part is shown, the imaginary part is small. At $|\mathbf{q}| = k$ there is a pole.

This product has the same eigenfunctions as $\mathcal{A}$ and $\mathcal{A}_T$, and the eigenvalues are those of $\mathcal{A}$ divided by those of $\mathcal{A}_T$, leading to (for Z=0)

$$\lambda^P(\mathbf{q}) = \frac{\lambda(\mathbf{q})}{\lambda^T(\mathbf{q})} = \frac{\widehat{G}(\mathbf{q})}{\widehat{G}_T(\mathbf{q})} \quad , \tag{4.13}$$

for both the parallel and the orthogonal eigenvalues $\lambda_\perp(\mathbf{q})$ and $\lambda_\parallel(\mathbf{q})$ of equations (3.15) and (3.16). The truncated Fourier transform $\widehat{G}_T$ cannot be found analytically, but we can make a numerical approximation. In Figure 4.1 we show $\lambda^P(\mathbf{q})$ for an example where $R_T = 0.1\lambda$. We see that the high $|\mathbf{q}|$ eigenvalues of the original eigenvalues in Figure 3.2 have been moved close to 1, which shows that the preconditioned operator (4.12) was close to the identity for these modes. However, the low $|\mathbf{q}|$ part of the spectrum of the preconditioned operator still has a pole. This shows that by truncating the Green function the low $|\mathbf{q}|$ behaviour of the operator is changed significantly, but the highly $|\mathbf{q}|$ modes are preserved rather well, or in other words, for high $|\mathbf{q}|$

$$\mathcal{A}_T\ \mathbf{a}e^{i\mathbf{q}\cdot\mathbf{x}} \approx \mathcal{A}\ \mathbf{a}e^{i\mathbf{q}\cdot\mathbf{x}} \quad , \tag{4.14}$$

but for low $|\mathbf{q}|$ this does not hold.

As we described above, we actually do not sparsify A, but A_Q. This means that we do not truncate the Green function, but the operator $\mathcal{Q}^{\mathrm{T}}\mathcal{A}\mathcal{Q}$, which we found in section 3.3.2 and for which we showed in section 3.3.3 that for the infinite plane conductor its off-diagonal blocks are zero. Both diagonal block operators can be seen as a simple integral

operator

$$(\mathcal{K}u)(\mathbf{x}) = \int_\Gamma K(\mathbf{x}-\mathbf{x}')u(\mathbf{x}')d^2x' \quad . \tag{4.15}$$

Truncating the interaction at some distance R_T, as in equation (4.11), can be seen as multiplication with a "step function"

$$H_T(\mathbf{y}) = \begin{cases} 1 & \text{for} \quad |\mathbf{y}| \leq R_T \\ 0 & \text{for} \quad |\mathbf{y}| > R_T \end{cases} \quad , \tag{4.16}$$

giving $K_T(\mathbf{y}) = H_T(\mathbf{y})K(\mathbf{y})$. This implies that the Fourier transform of the truncated interaction is a convolution of the Fourier transform of the original interaction and the Fourier transform of H_T,

$$\widehat{K}_T(\mathbf{p}) = \frac{1}{4\pi^2}\int \widehat{H}_T(\mathbf{p}-\mathbf{q})\widehat{K}(\mathbf{q})d^2q \quad . \tag{4.17}$$

The Fourier transform of the original interaction $\widehat{K}(\mathbf{q})$ has been calculated in section 3.3.4 ($\lambda_l(\mathbf{q})$ and $\lambda_c(\mathbf{q})$ in equations (3.77) and (3.78)), and the Fourier transform of H_T is

$$\widehat{H}_T(\mathbf{q}) = 2\pi R\frac{J_1(|\mathbf{q}|R)}{|\mathbf{q}|} \quad , \tag{4.18}$$

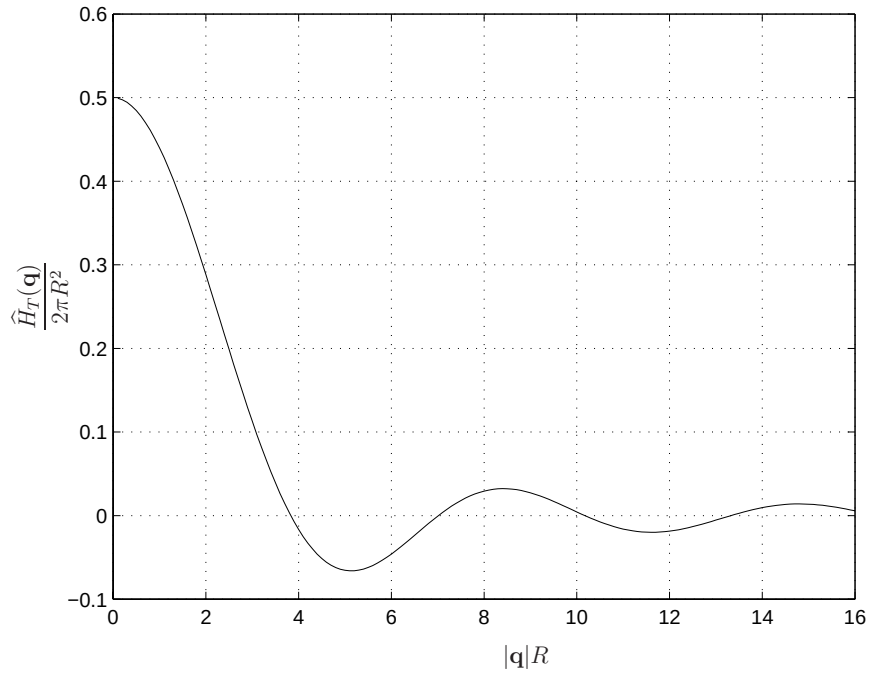
with J_1 being the first Bessel function of the first kind. As can be seen in Figure 4.2(a), $\widehat{H}_T$ is peaked around the origin with oscillatory decay for $|\mathbf{q}| \to \infty$. By scaling we see that the peak has a width inversely proportional to R. It can also be shown that $\int \widehat{H}_T(\mathbf{q})d^2q = 4\pi^2$. Hence, we can view the convolution (4.17) as a weighted average of $\widehat{K}(\mathbf{q})$, where the main contribution comes from a region of size $\mathcal{O}(\infty/\mathcal{R})$ centred around $\mathbf{p}$. If $\widehat{K}(\mathbf{q})$ is approximately linear on this region, $\widehat{K}_T(\mathbf{p})$ will approximate $\widehat{K}(\mathbf{p})$. In this case, the eigenvalue for the preconditioned system $(\widehat{K}(\mathbf{p})/\widehat{K}_T(\mathbf{p}))$ will be near 1 for this value of $\mathbf{p}$. We also see that increasing R will improve K_T and make the effective averaging region smaller, thereby bringing the eigenvalue of the preconditioned operator closer to 1. Since both $\lambda_l(\mathbf{q})$ and $\lambda_c(\mathbf{q})$ are more and more smooth for increasingly large $\mathbf{q}$, we expect better and better correspondence between $\widehat{K}(\mathbf{q})$ and $\widehat{K}_T(\mathbf{q})$. This shows again that truncation does not affect the highly oscillating modes very much.

In practice, we have used a square cutoff region instead of the circular one described above. This will slightly change our $\widehat{H}_T$, losing the radial symmetry

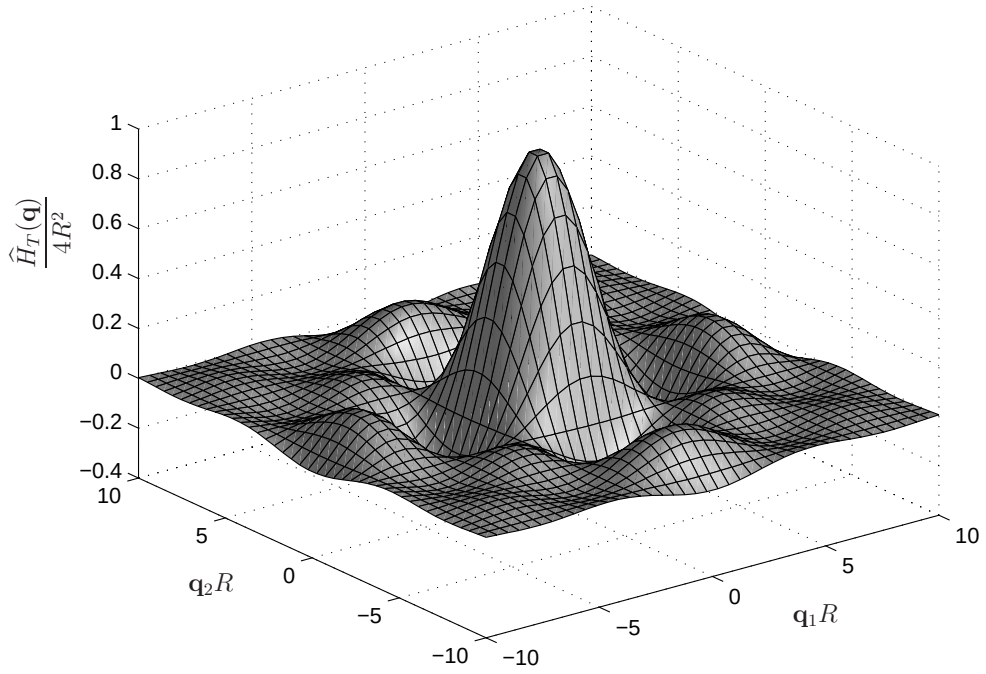
$$\widehat{H}_T = 4\frac{\sin(q_1R)\sin(q_2R)}{q_1q_2} \quad , \tag{4.19}$$

which is shown in Figure 4.2(b). Only the shape changed slightly, but the main principles do not change.

We still have to be sceptical about these results, since they are only valid for the continuous operators. In practice, we will choose R in the order of a few elements in order to get a nicely sparse matrix. For decreasing mesh size, we will thus also decrease the cutoff length R, which means that even for $h \to 0$ the continuous analysis will not apply. However, we can still use these ideas in order to get a feeling for what is happening when we sparsify our matrix.



(a) Circular cutoff.



(b) Square cutoff.

FIGURE 4.2: A scaled plot of $\widehat{H}_T(\mathbf{q})$ for both circular and square cutoff.

4.3 Coarse grid correction

For the coarse grid correction, we need to find an appropriate subspace to project on. As the "geometric" in geometric multigrid already indicates, we choose this subspace based on geometric information. The smoother is supposed to damp all highly oscillating modes and the coarse grid correction has to deal with smooth modes only. Since these smooth modes can be represented well on the finite element space of a coarser discretisation, we will try to let the subspace approximate this coarser finite element space.

After the basis transformation the finite element space consists effectively of two scalar finite element spaces. We choose separate projection subspaces for these scalar finite element spaces and then combine them into the total projection subspace. The corresponding projection operators will also be generated separately and combined afterwards.

Let us consider one of the fine grid scalar finite element spaces and assume that it has basis functions ϕ_i^h and corresponding Galerkin matrix A_h. In standard geometric multigrid we would now choose an analogous coarser finite element space with basis functions ϕ_j^H, and calculate the corresponding matrix A_H. In order to project the fine grid problem onto this coarser subspace, one has to construct an interpolation matrix P that (approximately) connects the two spaces,

$$\phi_j^H \approx \sum_i \phi_i^h P_{ij} \quad , \tag{4.20}$$

which implies that $A_H \approx P^{\mathrm{H}} A_h P$. Note that if the coarse grid space is not a subspace of the fine grid space, relation (4.20) cannot be exact.

The motivation for choosing the coarse grid finite element basis ψ_j^H, and calculating A_H as the discretisation using this basis, is that it is often cheap to calculate the matrix A_H. By choosing the coarse basis analogous to the fine one, A_H will have a lot of properties in common with A_h, for instance the sparsity. The disadvantage is that the relation (4.20) might not be exact, which means that the projection used for the coarse grid correction will not be exact.

Since we do not need to worry about preserving sparsity for our dense problems, and since computing A_H from scratch is not cheap, we can deviate from this scheme. Instead of choosing ψ_j^H, we will choose a P, and then define the ϕ_j^H with that P, such that we have an exact match for relation (4.20). This implies that we define $A_H = P^{\mathrm{H}} A_h P$, which is also the way we construct A_H. Because the coarse grid problem should represent the slowly oscillating modes relatively accurately, we choose our P such that A_H still approximates a coarser discretisation.

For the wires with regular discretisation, we use Figure 4.3 to illustrate our choice of P. The figure shows a very short piece of the wire. The large ●'s represent the quasi-charge basis functions on the fine grid (the ϕ_i^h), located at the centres of the segments ($\mathbf{x}_i$). The tent functions represent the different coarse grid degrees of freedom (φ_j), and the small •'s denote the resulting values for P_{ij} :

$$P_{ij} = \varphi_j(\mathbf{x}_i) \quad . \tag{4.21}$$

Note that these φ_j are not the coarse grid basis functions ϕ_j, but only templates for P_{ij}.

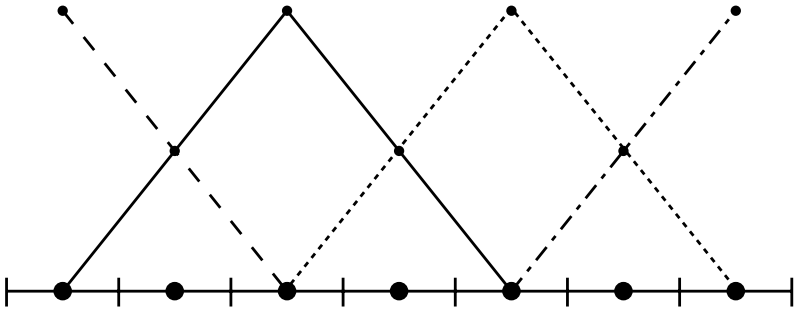


FIGURE 4.3: Example construction of P. The four φ_j's are denoted with different lines and the $\mathbf{x}_i$'s with the big ●'s. The small •'s correspond to values of P_{ij}.

At the boundaries, the charge density will not be zero, so we use a half tent function for φ_j at the boundary.

In the middle of the wire, there is one coarse degree of freedom for two fine degrees of freedom, which corresponds to a coarsening factor of 2. On this inner region of the wire, the non-zero P_{ij} values are $[1/2, 1, 1/2]$ for consecutive values of i and fixed j. This $1/2[1, 2, 1]$ is called the stencil.

For the regular square grids on the boards, this is generalised by using Cartesian products of the φ_j tent functions for the wire, resulting in a coarsening factor of 2 in both directions and results in the well-known 2D stencil

$$\frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{4.22}$$

for P.

In order to be able to use these stencils with coarsening factor 2, the fine grid needs to have an odd number of elements in each direction, as in Figure 4.3. On the boards, there are two scalar quantities, the quasi-charge and the variable associated with the current loops. Due to the different nature, there is always one more quasi-charge element than current loops in each direction, which means that they can never be both odd. To resolve this, we relax the stencil to allow coarsening with factors other than 2. In this case, the φ_j will not line up with the fine gird, as shown in the example in Figure 4.4, but this is not a big problem.

For the construction of P, we use the fact that we have a regular discretisation. If we want to generalise to irregular discretisations, then we could do something similar by distributing the points $\mathbf{x}_i$ over a domain according to the geometric position of the corresponding elements and choosing appropriate functions φ_j. A straightforward option would be to use the centre points of the fine elements for the $\mathbf{x}_i$. The coarse functions can then be simple finite element functions from a new (irregular) coarse discretisation.

So far, we constructed separate projectors P for the different components of the conductor (boards and wires). However, we did not yet consider the global current loops. Each global current loops has one degree of freedom, which will be kept in the coarse grid space. This means that for each of these degrees of freedom, P will contain

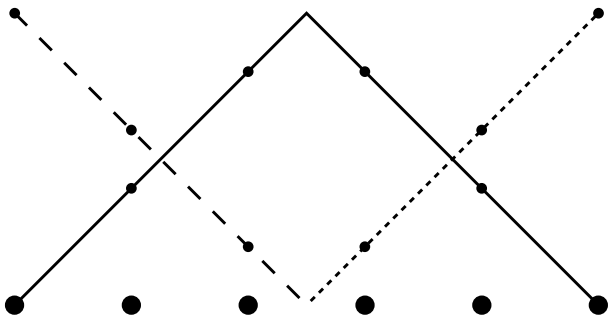


FIGURE 4.4: Example construction of P with an even number of fine grid variables. The three φ_j's are denoted with different lines and the $\mathbf{x}_i$'s with the big ●'s. The small •'s correspond to values of P_{ij}.

one unit entry such that P will act as the identity for these degrees of freedom. This means that the global current loops are present in all the coarse systems, where they are resolved in the direct solve at the coarsest level. We therefore the smoother does not need to damp the error associated with these global loop variables.

We have now constructed projectors P for both scalar variables and different parts (wires and boards) of the conductor. We will now combine them into one P. Two matrices P can be combined by

$$P_{1\&2} = \begin{pmatrix} P_1 & 0 \\ 0 & P_2 \end{pmatrix} \quad , \tag{4.23}$$

which can be repeated for more matrices P. Note that by combining both interpolation operators into one, we avoid to do separate multigrid or separate coarse grid corrections for the two types of variables, but the coarse grid problem contains both variables and the interaction between them. Only the interpolation and smoothing is done separately.

Inherent to this method of generating P, is that the constant function on the fine grid is alway represented exactly by the constant functions on the coarse grid. This follows from the fact that the sum of all the tent functions φ_j is the constant function. This is important to realise, since the constant quasi-charge vector is an eigenvector, with eigenvalue zero, of the matrix A_Q associated with the fine grid. As a result, for all matrices A_Q associated with the different levels of coarse grids, the constant vector is an eigenvector with eigenvalue zero. This will present problems for the standard direct solver that we want to use on the coarsest level, since the coarsest grid system is singular. To resolve this we use a regularised LU-decomposition to solve this coarsest grid system, as is described in section 4.4.2.

4.4 Implementation details

4.4.1 Computation of matrices

Note that the explicit calculation of A_Q from A is very expensive because A is large and dense. However, we do not need to know A_Q explicitly, but we only need a sparsified

version for the smoother, which is not so expensive to construct. The matrix-vector multiplications with A_Q can be performed by consecutive multiplication with Q, A, and Q^{T}. The coarse gird problem matrix $A_H = P^{\mathrm{T}} A_Q P$ is calculated explicitly using

$$A_H = (QP)^{\mathrm{T}} A (QP) \ , \tag{4.24}$$

which is cheaper because of the low column dimension of QP.

4.4.2 Regularisation of an LU-decomposition

In our multigrid implementation, we solve the linear system on the coarsest level with a direct method. As we saw in section 3.2, our linear system matrix will have one zero eigenvalue per connected part of the conductor due to the over complete quasi-charge basis. These zero eigenvectors are preserved by our coarse gird projections, which means that the small linear system for the coarsest grid will have a null space of the same dimension as the large linear system for the fine grid. The corresponding eigenspace will be the coarse analogue of the null eigenspace of the large system, and therefore it is known.

For simplicity, we will now consider the case of a single connected conductor surface, resulting in only one zero eigenvector for the large system, and also one corresponding zero eigenvector for the small system. We would like to use an LU-decomposition (with row pivoting) to solve the small system equation. Since the small system matrix is singular, the U part of the LU-decomposition will have a zero pivot at the last equation that involves the quasi-charge basis. Without loss of generality, we assume that this will be the last diagonal element of U. We will show that this coarsest grid problem can be solved by simply replacing this zero pivot by the value 1 and removing the singular direction from the residual and the solution, before and after the triangular solves.

Let A be the small system matrix. The LU-decomposition will result in $PA = LU$ with P a permutation matrix, L lower triangular with unit diagonal, and U upper triangular with a non-zero diagonal except for one zero at the last diagonal position. By replacing the zero pivot by one, we obtain a regularised matrix

$$\widetilde{A} = P^{-1} L (U + R) = A + P^{-1} R \ , \tag{4.25}$$

where $R = e_n e_n^{\mathrm{T}}$ is the zero matrix with a 1 at the last diagonal position. We will now show that if the zero eigenvector is deflated from $\widetilde{A}$, the result is the same as A.

Theorem 4.1 *Let $A \in \mathbb{C}^{n \times n}$ be represented by*

$$A = \begin{pmatrix} B & c \\ d^{\mathrm{H}} & \alpha \end{pmatrix} \ , \tag{4.26}$$

with $B \in \mathbb{C}^{(n-1)\times(n-1)}$, $c, d \in \mathbb{C}^{n-1}$, $\alpha \in \mathbb{C}$, and B non-singular. Let $AV = V\Lambda$ with

$$\Lambda = \begin{pmatrix} 0 & 0 \\ 0 & \Gamma \end{pmatrix} \ , \tag{4.27}$$

$\Gamma \in \mathbb{C}^{(n-1)\times(n-1)}$, *and* V *non-singular. Let* $\widetilde{A} = A + R$ *be non-singular,with*

$$R = \begin{pmatrix} 0 & c' \\ 0 & \alpha' \end{pmatrix} \ , \tag{4.28}$$

then

$$V^{-1} A \widetilde{A}^{-1} V = \begin{pmatrix} 0 & 0 \\ \star & I \end{pmatrix} \ , \tag{4.29}$$

where the $\star$ *denotes an unspecified* $(n-1) \times 1$ *matrix block.*

Proof: The block LDU decomposition of A is given by

$$A = \begin{pmatrix} I & 0 \\ d^{\mathrm{H}} B^{-1} & 1 \end{pmatrix} \begin{pmatrix} B & 0 \\ 0 & s \end{pmatrix} \begin{pmatrix} I & B^{-1}c \\ 0 & 1 \end{pmatrix} \ , \tag{4.30}$$

where $s = \alpha - d^{\mathrm{H}} B^{-1} c$ is the Schur complement. Because the matrix A is singular and B is not, s must be zero. In the same way,

$$\widetilde{A} = \begin{pmatrix} I & 0 \\ d^{\mathrm{H}} B^{-1} & 1 \end{pmatrix} \begin{pmatrix} B & 0 \\ 0 & s' \end{pmatrix} \begin{pmatrix} I & B^{-1}(c+c') \\ 0 & 1 \end{pmatrix} \ , \tag{4.31}$$

in which $s' = \alpha' - d^{\mathrm{H}} B^{-1} c'$ is the Schur complement, which is non-zero since $\widetilde{A}$ is non-singular. By combining (4.30) and (4.31), we find that

$$\begin{aligned} A\widetilde{A}^{-1} &= \begin{pmatrix} I & 0 \\ d^{\mathrm{H}} B^{-1} & 1 \end{pmatrix} \begin{pmatrix} B & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} I & -B^{-1}c' \\ 0 & 1 \end{pmatrix} \begin{pmatrix} B^{-1} & 0 \\ 0 & s' \end{pmatrix} \begin{pmatrix} I & 0 \\ -d^{\mathrm{H}} B^{-1} & 1 \end{pmatrix} \\ &= G^{-1} \begin{pmatrix} I & \star \\ 0 & 0 \end{pmatrix} G \ , \end{aligned} \tag{4.32}$$

where

$$G = \begin{pmatrix} I & 0 \\ -d^{\mathrm{H}} B^{-1} & 1 \end{pmatrix} \ . \tag{4.33}$$

We use the $\star$ notation to denote unspecified matrix blocks.

From (4.30), we see that the null space of A is spanned by

$$\begin{pmatrix} -B^{-1}c \\ 1 \end{pmatrix} \ , \tag{4.34}$$

so that we can write V as

$$V = \begin{pmatrix} -B^{-1}c & W \\ 1 & z^{\mathrm{H}} \end{pmatrix} \ . \tag{4.35}$$

Evaluation of GV leads to

$$GV = \begin{pmatrix} -B^{-1}c & W \\ d^{\mathrm{H}} B^{-2} c + 1 & d^{\mathrm{H}} B^{-1} W + z^{\mathrm{H}} \end{pmatrix} \ . \tag{4.36}$$

Using $AV = V\Gamma$,

$$\begin{aligned} AV &= \begin{pmatrix} B & c \\ d^{\mathrm{H}} & \alpha \end{pmatrix} \begin{pmatrix} -B^{-1}c & W \\ 1 & z^{\mathrm{H}} \end{pmatrix} = \begin{pmatrix} 0 & BW + cz^{\mathrm{H}} \\ 0 & d^{\mathrm{H}} W + \alpha z^{\mathrm{H}} \end{pmatrix} \ , \\ V\Lambda &= \begin{pmatrix} -B^{-1}c & W \\ 1 & z^{\mathrm{H}} \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & \Gamma \end{pmatrix} = \begin{pmatrix} 0 & W\Gamma \\ 0 & z^{\mathrm{H}}\Gamma \end{pmatrix} \ , \end{aligned} \tag{4.37a}$$

and $s = \alpha - d^{\mathrm{H}}B^{-1}c = 0$ (from (4.30)), one can show that $d^{\mathrm{H}}B^{-1}W + z^{\mathrm{H}} = 0$. As a result, the inverse of GV can be written as

$$(GV)^{-1} = \begin{pmatrix} -B^{-1}c & W \\ d^{\mathrm{H}}B^{-2}c + 1 & 0 \end{pmatrix}^{-1} = \begin{pmatrix} 0 & \star \\ W^{-1} & \star \end{pmatrix} \quad . \tag{4.38}$$

Combining (4.32) and (4.38), we see that

$$V^{-1}A\widetilde{A}^{-1}V = (GV)^{-1}\begin{pmatrix} I & \star \\ 0 & 0 \end{pmatrix} GV = \begin{pmatrix} 0 & \star \\ W^{-1} & \star \end{pmatrix}\begin{pmatrix} I & \star \\ 0 & 0 \end{pmatrix}\begin{pmatrix} \star & W \\ \star & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ \star & I \end{pmatrix}, \tag{4.39}$$

which concludes the proof. □

Since $P^{-1}R$ can only have elements in the last column and $\widetilde{A} = P^{-1}L(U + R) = A + P^{-1}R$ is a non-singular matrix, the above Theorem applies. For $Ax = b$ to have a solution, we must assume that b does not contain a component in the zero eigenvector direction. To make sure of this, we can use a projection S, given on the V basis by

$$V^{-1}SV = \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} \quad , \tag{4.40}$$

and solve $Ax = Sb$. If we now use our regularised LU-decomposition for the solution of x, we find

$$V^{-1}Ax = V^{-1}A\widetilde{A}^{-1}Sb = \begin{pmatrix} 0 & 0 \\ \star & I \end{pmatrix}\begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} V^{-1}b = V^{-1}Sb \quad . \tag{4.41}$$

This shows that using our regularised LU-decomposition gives us $Ax = Sb$, but we still need to make sure that x does not contain a component in the zero eigenvector direction. This can be taken care of with the projection S. We thus use

$$x = S(U + R)^{-1}L^{-1}PSb \tag{4.42}$$

to find our coarsest grid solution.

To make this practical, we still need to determine S explicitly. In general this would be costly, but for complex symmetric matrices, a real valued zero eigenvector is always orthogonal with respect to all the other eigenvectors, as is shown in the next lemma.

Lemma 4.2 *Let $A = A^{\mathrm{T}} \in \mathbb{C}^{n\times n}$, $y \in \mathbb{C}^n$, $x \in \mathbb{R}^n$, and $\lambda \in \mathbb{C}\backslash\{0\}$. If $Ax = 0$ and $Ay = \lambda y$, then $x \perp y$.*

Proof:

$$y^{\mathrm{H}}x = (\frac{1}{\lambda}Ay)^{\mathrm{H}}x = \frac{1}{\lambda^*}y^{\mathrm{H}}A^{\mathrm{H}}x = \frac{1}{\lambda^*}y^{\mathrm{H}}A^*x = \frac{1}{\lambda^*}y^{\mathrm{H}}(Ax)^* = 0 \quad , \tag{4.43}$$

which shows that $x \perp y$. □

Since our small coarse grid matrix A is complex symmetric and our zero eigenvector v_1 is real, v_1 is orthogonal to all the other eigenvectors and S is just an orthogonal projection: $S = I - v_1 v_1^{\mathrm{T}}/\|v_1\|^2$.

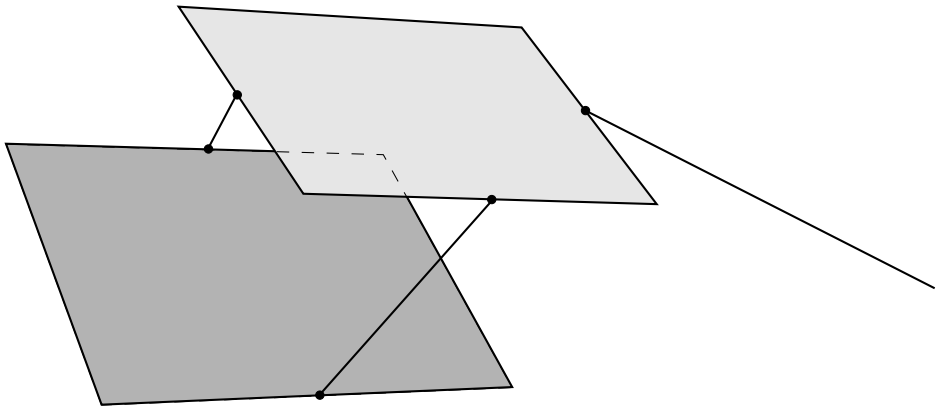


FIGURE 4.5: A 3-dimensional representation of the test problem "complex", consisting of two boards and 3 wires.

4.5 Experimental results

In our experiments we have used 5 different methods for the solution of the linear system (3.43)

- A direct solver from LAPACK (zgesv) [1],
- Full GMRES with Gauss-Seidel (GS) preconditioning,
- Full GMRES with the sparse approximate inverse (SAI) preconditioning,
- Full GMRES with multigrid with Gauss-Seidel smoother (MG with GS) as preconditioner.
- Full GMRES with multigrid with SAI smoother (MG with SAI) as preconditioner.

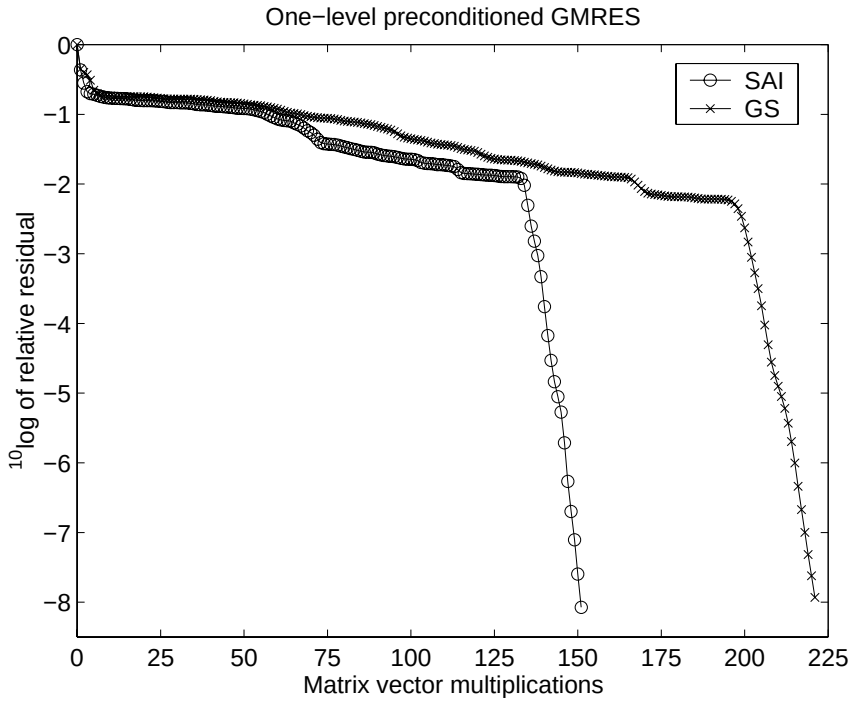
More information about GMRES can be found in section 1.2 and SAI is discussed in section 4.2.1.

All computations in this section were done in double precision on a Sun SPARC-server-1000. The iterative solvers where terminated when the relative residual norm ($\|b_Q - A_Q x\|/\|b_Q\|$) was less than $2 \cdot 10^{-8}$. The factor of 2 is there for convenience.

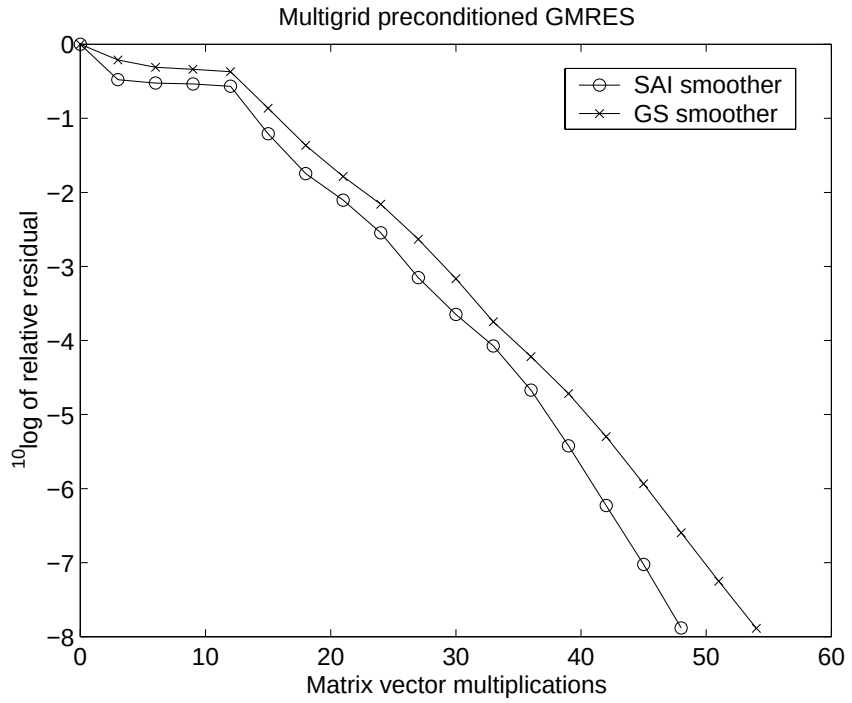
We have tested the different methods for a test problem with two boards, two connecting wires, and a long antenna wire, see Figure 4.5. The boards are square and their physical size is 2×2 meter, the antenna is about 4 meter long. We will call this test problem "complex", in contrast with the even simpler test problems. This is still a relatively simple geometry, but it already gives a good impression of the behaviour of the different methods.

Figure 4.6 shows an example of the convergence history using the different iterative solvers mentioned above. Since one iteration of GMRES preconditioned with a multigrid V-cycle costs approximately 3 (fine grid) matrix-vector multiplications, we show the number of (fine grid) matrix-vector multiplications along the horizontal axis. This is a reasonable measure for the real costs, since this is the most expensive part of each iteration. What is not seen here are the preparation costs. These can be seen in Table 4.1.

In Table 4.1 we show the CPU-times for the "complex" test problem at a frequency of 100 MHz ($\lambda = 3$ m), using 7 different grid sizes ranging from 9 to 37 elements per board-edge or equivalently 13.5 to 55.5 elements per wavelength. Since the physics is scalable, the numerical results are the same when all lengths (including the wavelength) are scaled.



(a) Using simple one-level preconditioning.



(b) Using multigrid preconditioning.

FIGURE 4.6: The result of preconditioned GMRES for the test problem "complex" with 2900 degrees of freedom at 100MHz.

n	calc. A	Direct	SAI			MG with GS			MG with SAI		
320	1	0	1 +	0	(42)	0 +	0	(11)	1 +	0	(9)
672	5	4	2 +	3	(61)	1 +	2	(12)	2 +	1	(10)
1430	22	40	4 +	19	(94)	3 +	11	(18)	6 +	8	(14)
2102	47	134	7 +	54	(125)	5 +	24	(19)	10 +	19	(15)
2900	89	488	12 +	121	(150)	10 +	43	(18)	18 +	38	(16)
3828	156	1439	16 +	262	(185)	18 +	78	(19)	27 +	66	(16)
5450	316	4182	27 +	632	(229)	37 +	192	(23)	54 +	152	(18)
measured order	2.0	3.5	1.4	2.6	(0.7)	2.0 +	2.2	(0.2)	1.7	2.2	(0.2)
expected order	2.0	3.0	1.0	2+?	(?)	2.0	2.0	(0)	2.0	2.0	(0)

TABLE 4.1: CPU-time in seconds for the calculation of A and four solution methods for the "complex" problem at 100 MHz. Times are split in preparation and solve times. The number of iterations is shown in parenthesis. The bottom lines show a crude estimation of the maximum exponent in the dependency on n, and the expected exponent. The "?" denotes that we do not know what exponent to expect.

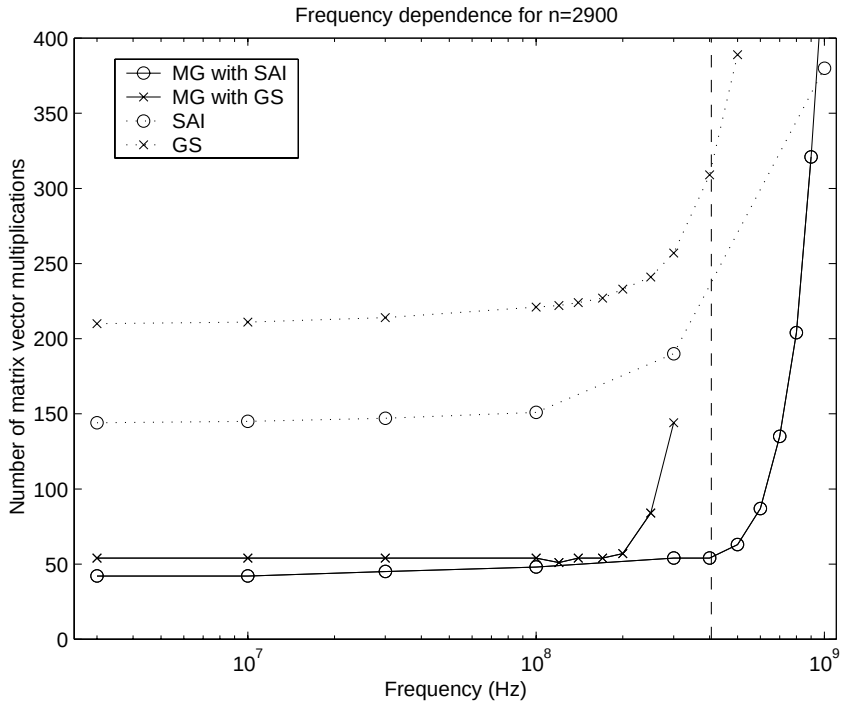


FIGURE 4.7: Frequency dependence for problem size $n = 2900$. The maximum element size was $h = 7.4$ cm. The vertical line shows the frequency where $\lambda = 10h$. Lines stop where convergence was not reached within 450 matrix-vector multiplications.

For this frequency MG with GS and MG with SAI are comparable in efficiency.

We may expect that the effectiveness of our approach is frequency dependent. The idea of the basis transformation (section 3.2) was inspired by low frequency problems and based on relatively strong capacitive effects. From Figure 4.7 we see for which frequency range this approach works well. We see that the convergence deteriorates as the frequency increases. For the largest frequencies the solvers did not converge within 450 matrix-

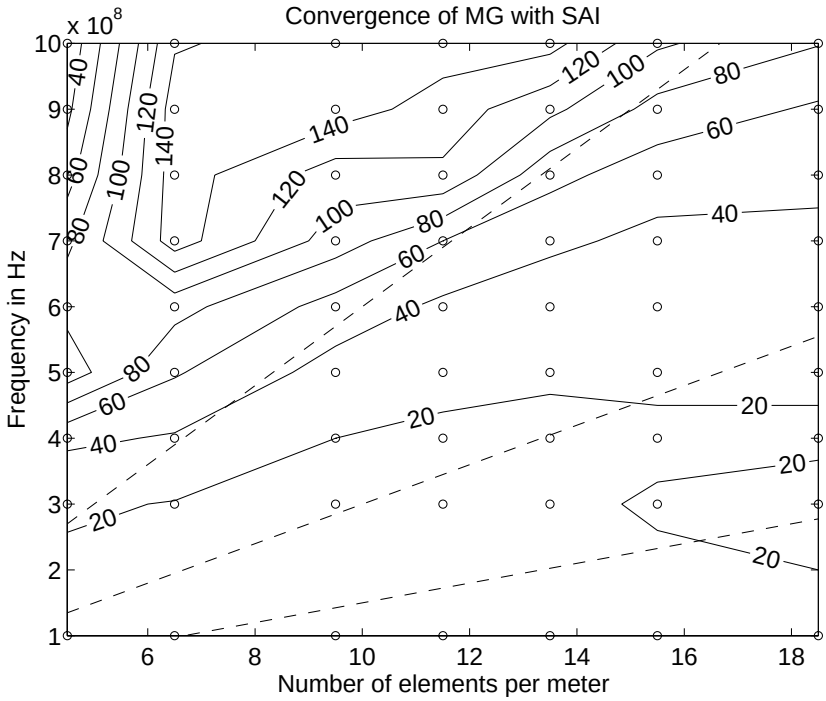


FIGURE 4.8: Dependence of the convergence on the mesh size and frequency. The level curves denote the number of GMRES iterations when using MG with SAI smoother. The dashed lines are lines with a constant number of elements per wavelength, from top to bottom 5, 10, and 20 elements per wavelength. The ∘ show the data points and we used a maximum of 150 iterations.

vector multiplications. This happens earlier for MG with GS, because of instabilities in the GS solve. This makes MG with SAI more reliable for the high frequency range. For the frequencies where MG with SAI does not converge quickly, the wavelength λ is so small compared with the element size ($h = 7.4$ cm) that this discretisation does not make much sense. Drastic refinement of the discretisation was not possible because of computer memory limitations. Reducing h with a factor of 2 would result in a 2 Gbyte matrix. This will only be possible on a parallel machine or with a matrix free method for the matrix-vector multiplication.

In order to get an impression of the dependence of this frequency barrier, we also show a contour plot the convergence depending on the mesh size and the frequency in Figure 4.8.

4.6 Conclusions

We have seen that the simple Gauss-Seidel (GS) and sparse approximate inverse (SAI) preconditioners in combination with GMRES lead to long stagnation of the convergence (see Figure 4.6(a)). The length of this stagnation phase depends on the system size n, leading to an increasing number of iterations for increasing n, as can be seen in Table 4.1. Using a multigrid V-cycle as preconditioner leads to much faster convergence, which is

only very mildly dependent on the system size n, as can be seen in the Table 4.1.

Unfortunately, the convergence of the multigrid preconditioned GMRES is frequency dependent. For high frequencies, the GS smoother becomes unstable, and the solver does not converge. The critical point is approximately $\lambda/h \approx 12$. The SAI smoother does not suffer from these problems, but for high frequencies, the number of iterations increases strongly, as can be seen in Figure 4.7. The frequency beyond which the number of iterations goes up, increases with decreasing element size, as can be seen in Figure 4.7. However, for constant λ/h, the number of iterations seems to increase for increasing system size. This is relevant, because for high frequencies there is a restriction on the element size $h \ll \lambda$ in order to be able to obtain accurate results. Fortunately, for the frequency and grid size range we have been able to test, the number of iterations was always acceptable for $h \leq \lambda/10$. For much smaller ratios λ/h, the discretisation is too coarse to accurately represent the oscillations in the Green function (2.15), and the resulting linear system does not represent the EFIE (2.16) any more. As a result, our preconditioning methods fail.

These results show that we can efficiently solve the EFIE (2.16) iteratively, but as can be seen in Table 4.1, the computation of the matrix A is now the bottleneck for the CPU-time. One way to resolve this is, would be the use of the Fast Multipole Method (FMM) described in section 2.7. This method makes it possible to do matrix-vector multiplications with (an approximation of) A without explicitly calculating A. This removes the need to explicitly calculate A and also reduces the cost of a matrix-vector multiplication. However, if the matrix A is not explicitly available, the multigrid method cannot be implemented as described in this chapter. In section 4.7 we will discuss a possible combination of FMM and multigrid.

As can be seen in Figure 4.6(b), multigrid preconditioned GMRES still shows a few steps of initial stagnation. In the next chapter, we will discuss a method to remove this stagnation for possible second and further right-hand sides, reducing the incremental cost for one extra right-hand side.

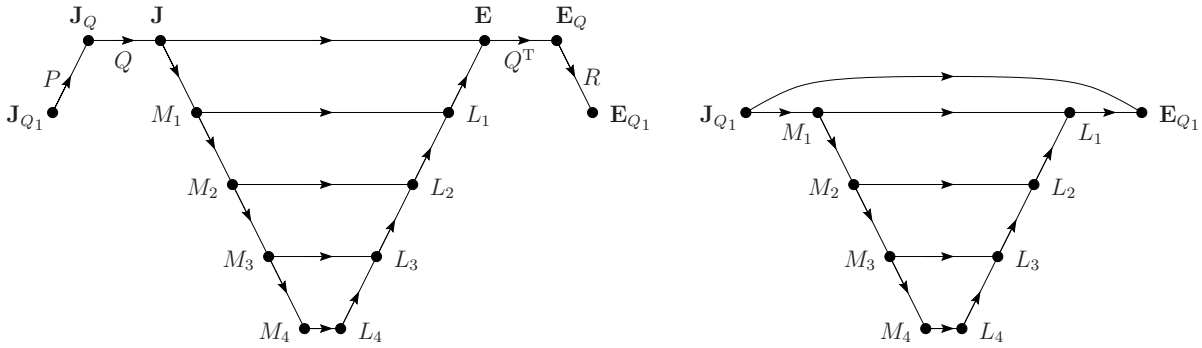
4.7 Combination with the Fast Multipole Method

In order to obtain the coarse grid matrix A_H with the explicit multiplication (4.24), we need direct access to the matrix A. As a result, the multigrid method cannot be trivially combined with a matrix-free matrix-vector multiplication like the fast multipole method (FMM) described briefly in section 2.7.

A naive approach to combine the two methods would be to compute matrix-vector multiplications with the coarse grid matrix by multiplying with each of the components of the product (4.24) separately. This would mean that, for each multiplication with a coarse grid matrix, we would have do perform a multiplication with the large matrix A, which will make the V-cycle very expensive.

It is our opinion that, since both multigrid and the FMM are based on multiple hierarchical levels, multigrid can be combined with the FMM in a more efficient way. We do not intend to present a full algorithm here, but only an idea how an efficient combination might be achieved.

A matrix-vector multiplication with A using the FMM can be decomposed in several



(a) When using this scheme, we have to do a full FMM cycle in order to apply a coarse grid matrix-vector multiplication. However, the top level can be eliminated by combining the linear operators of the top two levels.

(b) The combined diagram where the fine level currents and fields have disappeared.

FIGURE 4.9: Schematic representation of the first level coarse interaction matrix $A_c = RA_QP = RQ^{\mathrm{T}}AQP$ as needed in the multigrid algorithm. The M_ℓ's are multipole coefficients, the L_ℓ's are local expansion coefficients and $\mathbf{J}$ and $\mathbf{E}$ are the usual currents and electric fields, $\mathbf{J}_Q$ and $\mathbf{E}_Q$ with respect to the Q basis, and $\mathbf{J}_{Q_1}$ and $\mathbf{E}_{Q_1}$ the first coarse level variables.

steps, as shown in Figure 2.13. For a multiplication with the first coarse grid matrix A_H, this diagram is extended to the one shown in Figure 4.9(a). Just as in Figure 2.13, each line represents a linear operator. By combining operators, we can change Figure 4.9(a) into the diagram in Figure 4.9(b). In this new diagram, the intermediate stages with the fine grid variables have disappeared. The top line between $\mathbf{J}_{Q_1}$ and $\mathbf{E}_{Q_1}$ in Figure 4.9(b) corresponds to the product of the operators represented by the $\mathbf{J}_{Q_1} \rightarrow \mathbf{J}_Q \rightarrow \mathbf{J} \rightarrow \mathbf{E} \rightarrow \mathbf{E}_Q \rightarrow \mathbf{E}_{Q_1}$ line in Figure 4.9(a). The operator represented by the $\mathbf{J}_{Q_1} \rightarrow M_1$ line in Figure 4.9(b)is the product of the operators corresponding to the $\mathbf{J}_{Q_1} \rightarrow \mathbf{J}_Q \rightarrow \mathbf{J} \rightarrow M_1$ line in Figure 4.9(a). The same type of construction is used for the operator represented by the $L_1 \rightarrow \mathbf{E}_{Q_1}$ line in Figure 4.9(b). For further levels of coarser grid matrices, an analogous combination can be made.

All the operators in Figure 4.9(a) are sparse banded matrices, with interactions limited by physical distance. As a result, the combined operators in Figure 4.9(b) will also be sparse banded matrices, with interactions limited by physical distance. However, by combining operators, the bandwidth will increase and the matrices will have more elements per row.

By avoiding the fine level variables during a coarse grid matrix-vector multiplication, while still using sparse operators, the coarse grid matrix-vector multiplications can be applied efficiently. However, the combining of operators may require explicit storage of the FMM operators, which will require more memory than a standard FMM implementation. There are still a lot of details to be worked out and a lot of choices to be made in order to turn this idea into an efficiently working algorithm.

An alternative way of combining a fast matrix-vector multiplication with a fast iterative solver is the use of a wavelet basis. However, we have not done any research in this direction.

Chapter 5

Reuse of computational information

As we already discussed in section 2.1.1, the electronics in the apparatus that we are simulating is represented by voltage sources on the wires. The number of these sources can be rather large, depending on the system complexity, and 50 sources is not extreme. For each of these sources we need to calculate the response of the system, which means that we have to solve a linear system $Ax = b$. Fortunately, the matrix A is the same for all these sources; only the external field represented by b is different. We thus have to solve a linear system with many right-hand sides. When using a direct solver based on LU-decomposition, this is not a problem. Once we have made the LU-decomposition of the matrix, we can use that to find the solutions for all right-hand sides. This way, the incremental cost for one extra right-hand side are relatively small. In chapter 4 we described an efficient iterative solver that is much faster for the first right-hand side. However, there is no obvious way to get the solutions for the remaining right-hand sides at low incremental cost, and just repeating the whole process for the other right-hand sides is the easiest solution, leading to high incremental cost for extra right-hand sides.

As is explained in section 1.2, the iterative Krylov subspace solver GMRES that we have used builds a search space that depends on the right-hand side. For each new right-hand side we will have to build a new search space in order to use GMRES, and most of the CPU-time in the solve is consumed by the matrix-vector multiplications with the large dense matrix A, that are needed to build the search space.

In section 5.1, we will discuss a method that will allow us to iteratively solve the linear system for the second and further right-hand sides, more efficiently than for the first right-hand side by reusing computational information.

We also have to find the system response for a whole frequency range, but then the matrix A changes for the different frequencies while the right-hand sides b are the same for each frequency. In section 5.2, we will try to use the information we already have available from computation for other frequencies, when solving for the new frequency.

5.1 Multiple right-hand sides

When using an iterative method to solve the linear systems, the dense matrix-vector multiplications dominate the CPU-costs, and thus, only a reduction in the number of matrix-vector products can significantly decrease the work. The reuse of the GMRES

polynomial for multiple right-hand sides, as is done in some methods, or any other change in the GMRES polynomial, does not accomplish this. Since each right-hand side has its own optimal polynomial, which is used in GMRES, any change can only increase the order of the needed polynomial, and thus increase the number of matrix-vector multiplications needed. This can only save on the GMRES inner products, vector updates and the reduced system solve that are needed to determine the GMRES polynomial, but these are not the CPU-dominating factors here.

The best way to reduce the number of matrix-vector multiplications, is by sharing search space information between the solves. This way, the matrix-vector multiplications done for one right-hand side can also be used for the other right-hand sides.

This can be done in different ways. One option is to solve for all right-hand sides simultaneously. This is done by various so-called block Krylov subspace solvers. They generate a combined Krylov search space for all right-hand sides together, and try to find optimal solutions for all right-hand sides from this subspace. Applying a block method like this for 50 right-hand sides can quickly result in memory problems and a large amount of work needed to orthogonalise the basis. A block method that does not need to store the whole search space, like block-QMR [18], could be preferable. The disadvantage is that we lose the GMRES optimality. We cannot even use the efficient variants for complex symmetric systems, since our preconditioned matrix is not symmetric. Block-QMR would then require matrix-vector multiplications with the transpose of the matrix, which will be hard to construct for the multigrid preconditioner.

Another way, that is in some sense the opposite of the block methods, is to solve for all right-hand sides separately, and reuse search space information form previous right-hand sides for new right-hand sides. This is the approach we will follow.

To accomplish a reduction in the number of matrix-vector multiplications, we reuse the search space V_k from the previous solve. If we use this search space for the next solve, the search space for the new solve will not be a Krylov subspace, which means that we cannot use GMRES. We will do this using the GCR [17] based solver GMRESR [42], which allows the injection of an arbitrary search space and variable preconditioning. Vuik [43] also used GMRESR to inject search space information from previous right-hand side solves, but we will use a different method to select the information that is injected.

5.1.1 GMRESR

We solve the system $Ax = b$ with GMRESR, which means that, like in GMRES, we also build a search space V_k, but this will not necessarily be a Krylov subspace. This means that we cannot use the same V_k to store AV_k, as is done in GMRES (see equation (1.8)). In GMRESR, a separate basis for AV_k is stored in U_k, called the shadow space. The orthogonalisation of the basis is done such that $U_k^{\mathrm{H}} U_k = I$ while preserving $AV_k = U_k$. Like in GMRES, we choose the minimal residual solution from the search space V_k

(compare with (1.10))

$$\min_{x_k \in \langle V_k \rangle} \|r_0 - Ax_k\| = \min_{\alpha} \|r_0 - AV_k\alpha\| \tag{5.1a}$$

$$= \min_{\alpha} \|r_0 - U_k\alpha\| \tag{5.1b}$$

$$= \min_{\alpha} \|U_k^{\mathrm{H}} r_0 - \alpha\| \quad \Leftrightarrow \tag{5.1c}$$

$$x_k = V_k U_k^{\mathrm{H}} r_0 \ , \tag{5.1d}$$

which results for the residual r_k in

$$r_k = r_0 - Ax_k = r_0 - AV_k U_k^{\mathrm{H}} r_0 = r_0 - U_k U_k^{\mathrm{H}} r_0 \perp U_k \ . \tag{5.2}$$

The freedom to use any search space, allows us to start the iterative process with any pair (V,U), as long as $AV = U$ and $U^{\mathrm{H}}U = I$. We will call this *search space injection*. We are also free to expand the search space any way we like, but in order to get fast convergence, a sensible extension is required. Ideally, we would like to add the error $e_k = x - x_k$ to the search space V_k, such that $x \in V_{k+1}$. Obviously, the error is not available, but using a preconditioner $M_k \approx A^{-1}$, an approximation to the error $e_k = A^{-1}r_k \approx M_k r_k$ can be made. We gave the preconditioner an index k since it can be different in every step, which is called *variable preconditioning*. In the original article [42], this possibility was used to approximate the error with a few steps of GMRES, which explains the last R in GMRESR, which stands for recursive. The GMRESR algorithm with search space injection and variable preconditioning is shown in Figure 5.1. Note that the algorithm, as stated here, will breakdown if the extension vector for U_k is already in U_k. This breakdown is easily prevented by choosing another extension for V_k. In [42], one step of LSQR [31] is suggested to find an extension for the search space when breakdown is expected. However, for our problems, this situation has not occurred.

Unfortunately, we have to pay for the flexibility that GMRESR offers. The cost for one iteration consist of one application of the preconditioner, one matrix-vector multiplication with A, $k+1$ inner products, and $2(k+1)$ vector updates. Compared to GMRES, this is double the amount of vector updates, but we do not have to solve a small least squares system. We also have to store both V_K and U_k, which doubles the memory requirements compared with GMRES. However, since the number of iterations we will use is relatively small, and since our matrix-vector multiplication is the most expensive part in terms of CPU-time and memory requirements, this extra time and memory is relatively low for our application.

5.1.2 GMRESR and search space injection

In this section we will discuss some properties of GMRESR when it is used with a fixed preconditioner. First we will argue that, if no search space injection is used, GMRESR is a minimal residual Krylov subspace solver, like GMRES. Then we will show that search space injection is equivalent to an orthogonal projection of the linear system on the orthogonal complement of the injected shadow space $\langle U \rangle$. We can use this to remove small eigenvalues from the problem and get a better distribution of the remaining eigenvalues, which will often lead to better convergence.

Choose k, $V_k = [v_1, \ldots, v_k]$ and $U_k = [u_1, \ldots, u_k]$
such that $AV_k = U_k$ and $U_k^{\mathrm{H}} U_k = I$
$x_k = V_k U_k^{\mathrm{H}} b$
$r_k = b - U_k U_k^{\mathrm{H}} b$
while $||r_k||_2 >$ tol do
$k = k + 1$
$v_k^{(1)} = M_k r_{k-1}$
$u_k^{(1)} = A v_k^{(1)}$
for $i = 1 \ldots k-1$ do
$\alpha_i = u_i^{\mathrm{H}} u_k^{(i)}$
$u_k^{(i+1)} = u_k^{(i)} - \alpha_i u_i$
$v_k^{(i+1)} = v_k^{(i)} - \alpha_i v_i$
$u_k = u_k^{(k)} / ||u_k^{(k)}||_2$
$v_k = v_k^{(k)} / ||u_k^{(k)}||_2$
$x_k = x_{k-1} + v_k u_k^{\mathrm{H}} r_{k-1}$
$r_k = r_{k-1} - u_k u_k^{\mathrm{H}} r_{k-1}$

Main properties :
$x_k = V_k U_k^{\mathrm{H}} b = \arg\min_{x \in V_k} ||b - Ax||_2$
$r_k = (1 - U_k U_k^{\mathrm{H}}) b \perp U_k$

FIGURE 5.1: The GMRESR algorithm with search space injection and variable preconditioner M_k.

If a constant preconditioner is used, this preconditioner can be made implicit by using GMRESR for the linear system $(AM)y = b$, as described in section 1.2.4. We will assume here that preconditioning is implicit such that M_k in the GMRESR algorithm (Figure 5.1) is replaced by the identity.

If we start with $\langle U_1 \rangle = \langle Ab \rangle$, we can see by induction that we are actually building a Krylov subspace generated by A and Ab. We start with $\langle U_1 \rangle = \langle Ab \rangle = \mathcal{K}_1(A, Ab)$, and we suppose that

$$\langle U_k \rangle = \mathcal{K}_k(A, Ab) \ . \tag{5.3}$$

this also holds for some k. Using this, we see that

$$\begin{aligned} \langle U_{k+1} \rangle &= \langle U_k \rangle \oplus \langle A r_k \rangle \\ &= \langle U_k \rangle \oplus \langle A(I - U_k U_k^{\mathrm{H}}) b \rangle \\ &= \langle U_k \rangle \oplus \langle Ab - A U_k U_k^{\mathrm{H}} b \rangle \\ &\subset \langle U_k \rangle \oplus \langle A U_k \rangle \\ &= \mathcal{K}_k(A, Ab) \oplus A\mathcal{K}_k(A, Ab) \\ &= \mathcal{K}_{k+1}(A, Ab) \ . \end{aligned} \tag{5.4}$$

In general both sides are of dimension $k+1$, we have that $\langle U_{k+1} \rangle = \mathcal{K}_{k+1}(A, Ab)$, and

assumption (5.3) is correct by induction. The only exception occurs, if $Ar_k \in U_k$, in which case we must expand U_k with some other direction and this analysis is not valid. Unless this happens, we know that (5.3) holds, and since $AV_k = U_k$,

$$\langle V_k \rangle = \mathcal{K}_k(A, b) \ . \tag{5.5}$$

This shows that this search space in the same as the GMRES search space, and that, in exact arithmetic, we get the same results as for GMRES.

Now we can generalise this to the case where we use search space injection. We start with a V_0 and U_0, and a corresponding $r_0 = (I - U_0 U_0^{\mathrm{H}})b = P_0 b$. We use the notation $P_0 = (I - U_0 U_0^{\mathrm{H}})$ for the orthogonal projection on $\langle U_0 \rangle^{\perp}$. We find that

$$\begin{aligned} \langle U_1 \rangle &= \langle U_0 \rangle \oplus \langle A r_0 \rangle \\ &= \langle U_0 \rangle \oplus \langle P_0 A P_0 r_0 \rangle \\ &= \langle U_0 \rangle \oplus \mathcal{K}_1(PAP, PAPr_0) \ . \end{aligned} \tag{5.6}$$

Analogous to assumption (5.3), we assume here

$$\langle U_k \rangle = \mathcal{K}_k(P_0 A P_0, P_0 A P_0 r_0) \ , \tag{5.7}$$

which leads to

$$\begin{aligned} \langle U_{k+1} \rangle &= \langle U_k \rangle \oplus \langle A(I - U_k U_k^{\mathrm{H}})b \rangle \\ &= \langle U_k \rangle \oplus \langle P_0 A P_0 (I - U_k U_k^{\mathrm{H}})b \rangle \\ &= \langle U_k \rangle \oplus \langle P_0 A P_0 b - P_0 A P_0 U_k U_k^{\mathrm{H}} b \rangle \\ &\subset \langle U_k \rangle \oplus \langle P_0 A P_0 U_k \rangle \\ &= \mathcal{K}_k(P_0 A P_0, P_0 A P_0 b) \oplus P_0 A P_0 \mathcal{K}_k(P_0 A P_0, P_0 A P_0 b) \\ &= \mathcal{K}_{k+1}(P_0 A P_0, P_0 A P_0 b) \ , \end{aligned} \tag{5.8}$$

showing that the assumption (5.7) is correct, under the same "no potential breakdown" conditions as above. This shows that with the search space injection we actually have a minimal residual Krylov subspace method for the projected problem

$$P_0 A P_0 y = P_0 b \ . \tag{5.9}$$

This could have been expected by inspecting the GMRESR algorithm. In each step it projects the remaining problem onto the orthogonal complement of u_k, the new direction in U_k. Starting with an initial U_0 will project the system onto the orthogonal complement of U_0.

Another way of looking at search space injection, is that by injecting V_0 in the search space, the GMRESR process does not have to spend any interactions to find components of the solution that are in V_0, and this part of the problem is removed.

Note that P_0 projects on the orthogonal complement of U_0, and not on the orthogonal complement of V_0, as one might have hoped. This means that we cannot project on the orthogonal complement of any chosen subspace directly, but only on the orthogonal complement of the image under A of a chosen subspace. As a consequence, when we try to remove small eigenvalues from the problem, we need quite good approximations of the corresponding eigenvectors in order to make this strategy effective.

5.1.3 Selecting information

If we repeatedly inject the V and U from the previous solve in the new solve over 50 right-hand sides with an average of 10 iterations per right-hand side, this will lead to a 500 dimensional search and shadow space. This will lead to much work in the orthogonalisation and it is likely to lead to memory problems. These are the same problems that we expected with the block Krylov subspace solvers.

To prevent this, we want to reuse only the most relevant information from the previous solve(s). In section 1.2.3 we saw that the eigenvalues of the preconditioned matrix are an important factor in the convergence of the solver. In the ideal situation, we would like to inject the eigenvectors corresponding to the eigenvalues that cause slow convergence such that the projections in equation (5.9) would deflate them from the iteration matrix, leading to faster convergence.

Inspired by this ideal situation, we will try to select those directions from V that are close to the desired eigenvectors. To find these directions, we calculate Ritz pairs (see section 1.2.5) of A using V and U, and select the Ritz vectors that we expect to approximate the eigenvectors corresponding to the eigenvalues that cause the slow convergence, as will be explained below. These Ritz vectors and their image under A can then be injected in the next solve.

The GMRESR matrices V and U, with $AV = U$ and $U^{\mathrm{H}}U = I$, can be used to calculate harmonic Ritz pairs of A with respect to V by calculating the eigenpairs of $U^{\mathrm{H}}V$. This is done in the following way :

$$\frac{1}{\theta}y = U^{\mathrm{H}}Vy \quad \Leftrightarrow \tag{5.10a}$$

$$y = \theta U^{\mathrm{H}}Vy \quad \Leftrightarrow \tag{5.10b}$$

$$Uy = \theta UU^{\mathrm{H}}Vy = \theta Vy - \theta(I - UU^{\mathrm{H}})Vy \quad \Rightarrow \tag{5.10c}$$

$$AVy = \theta Vy + r \quad \Rightarrow \tag{5.10d}$$

$$Ax = \theta x + r \ , \tag{5.10e}$$

with $x = Vy$ and $r = -\theta(I - UU^{\mathrm{H}})x \perp U = AV$. Note that $x \in V$ and the eigenvector residual r is orthogonal to $U = AV$, which shows that the (x,θ) pairs we find this way are harmonic Ritz pairs of A w.r.t. V. In the first step we assume that $U^{\mathrm{H}}V$ is non-singular, thus $1/\theta \neq 0$. However, if $U^{\mathrm{H}}V$ is near singular, we might get inaccurate results. In practice we have not observed this, but if it would happen this would only imply that we cannot expect significant acceleration from the injection of these vectors.

The set of harmonic Ritz vectors x that we find using (5.10) spans the same space as V. However, we can now use the corresponding harmonic Ritz values θ to make a selection from this set to inject in the next solve. The harmonic Ritz pairs are asumed to approximate the eivenpairs of the matrix A in the relation $AV = U$. If we use explicit preconditioning, as done in the GMRESR algorithem in Figure 5.1, this will be the unpreconditioned matrix A_Q. Since we are interested in approximations to the eigenpairs of the preconditioned matrix, we have to use implicit preconditioning, in which case A corresponds to the preconditined matrix $A_Q M_Q$ where M_Q is the preconditioner for A_Q. This requirement to use implicit preconditioning makes it impossible to use a variable preconditioner.

We can now make a selection of the harmonic Ritz vectors that we expect to correspond to eigenvalues that delay the convergence. If the preconditioner is already very good, as we might expect for the multigrid preconditioner, we expect a cluster of eigenvalues in the neighbourhood of 1, and only a few eigenvalues further off that are responsible for the slow convergence. Since the Ritz values are supposed to approximate eigenvalues, we expect a comparable pattern for the Ritz values.

Since the GMRESR process behaves like a GMRES process, as we showed in section 5.1.2, we can apply the results from the convergence behaviour for GMRES of section 1.2.3 to GMRESR as well. Since our matrices are, in general, not very non-normal, we can think in terms of minimising (1.17). This means that, in order to get a small residual, the polynomial must be small for all the eigenvalues for which the eigevectors are represented in the initial residual r_0. In order to achieve this, the polynomial must have roots close to the outlying eigenvalues. Since the GMRES polynomial is the optimal polynomial for minimising the residual, there will be only one root for each outlying eigenvalue, since that is enough to get a small value for the polynomial in these points. Since the harmonic Ritz values are the same as the roots of the GMRES polynomial, we can thus expect that, when the linear solve has converged, there are harmonic Ritz values that approximate the outlying eigenvalues for which the eigenvectors are represented in the the initial residual. Our experimental results, for which one example is shown in Figure 5.2, confirm these ideas.

Since the eigenvalues that lie away from the cluster are the eigenvalues that can slow down the convergence, we will select the Ritz values that lie away from 1, and use the corresponding Ritz vectors for injection in the new search space. In Figure 5.2, we show an example of the Ritz values and corresponding residual norms we found for the test problem "complex" that we have also used in section 4.5. The top line in Figure 5.3 shows the corresponding convergence of the GMRESR solve. The preconditioner that was used was already very good, resulting in fast reasonably convergence. However, we see that the initial convergence speed is much lower than the final speed. This is due to outlying eigenvalues, so we will select the Ritz values that approximate these for injection of the corresponding Ritz vectors. Since there are not many of them, we can make a wide selection. We selected those values for which

$$|\theta - 1| > \frac{1}{2} \ , \tag{5.11}$$

as is also shown in Figure 5.2. Injecting the corresponding Ritz vectors results in the second convergence line in Figure 5.3. Effectively, we have skipped the initial slow convergence that was caused by the outlying eigenvalues, and immediately achieved the high final convergence rate. For this example, injecting the 6 selected Ritz vector saves us 7 iterations, which costs 21 matrix-vector multiplications. This is a cost reduction of almost 40%. As always, there is some optimum for the selection criterion. Since the extra cost of one more Ritz vector injection is relatively small compared with the potential gain, we expect that the optimum selection condition would make a wide selection. We have not tried any rigorous research to try find the optimum, but we have tried a few other selection criteria. For our problem, we found that the criterion (5.11) works fine.

After selection of the Ritz values, we can easily calculate a new pair (V_0, U_0) with $AV_0 = U_0$ and $U_0^{\mathrm{H}} U_0 = I$. To do this, we put the short vectors y of equation (5.10) that

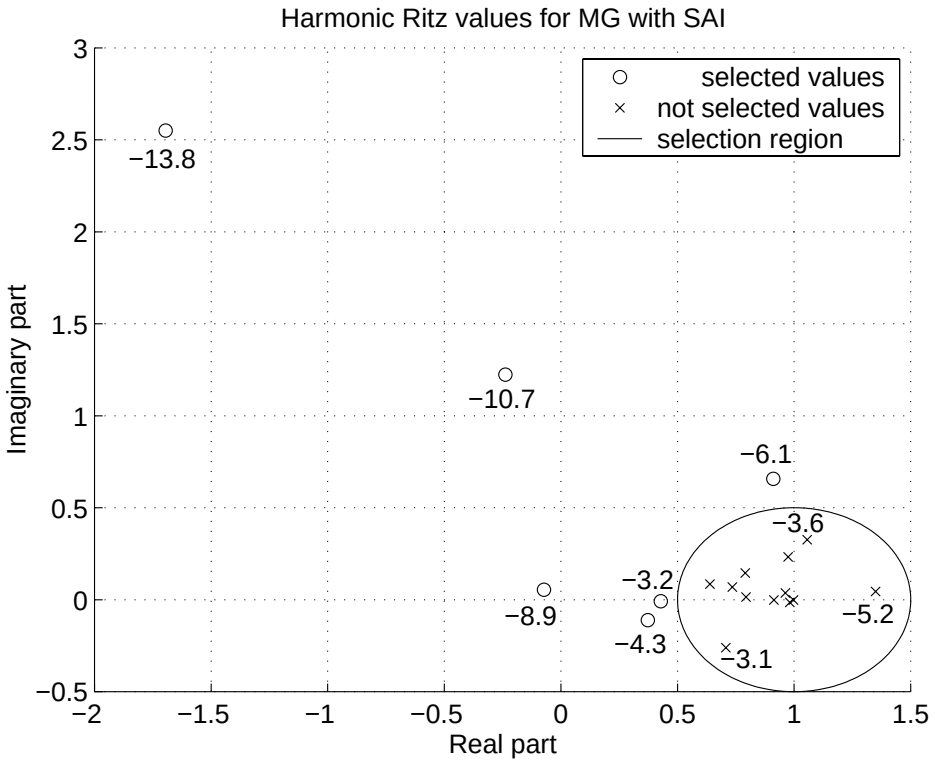


FIGURE 5.2: Harmonic Ritz values for the preconditioned matrix with MG and SAI smoother, for test problem "complex" at 100MHz with 5450 unknowns. The selection criterion selects all values outside the circle. The additional number show the $^{10}\log(\|r\|)$ where r is the eigenvector residual from equation (5.10), omitted values are larger than -2.

correspond to selected Ritz values θ in a small matrix Y. We now want to inject the search space $\langle VY \rangle$ and shadow space $\langle AVY \rangle = \langle UY \rangle$. By orthogonalising Y we get $\widetilde{Y}$, and we set $V_0 = V\widetilde{Y}$ and $U_0 = U\widetilde{Y}$. It is easily checked that $AV_0 = AV\widetilde{Y} = U\widetilde{Y} = U_0$ and $U_0^{\mathrm{H}}U_0 = \widetilde{Y}^{\mathrm{H}}U^{\mathrm{H}}U\widetilde{Y} = \widetilde{Y}^{\mathrm{H}}\widetilde{Y} = I$.

Figure 5.3 shows the convergence acceleration due to search space injection. It indicates that the injection of the Ritz vectors removes the stagnation, and convergence starts directly at approximately the same speed as at the end of the original convergence. This is the result of the fact that the eigenvalues corresponding to the selected Ritz values are effectively removed from the problem, which improves the spectral properties of the problem. This method will not remove the convergence problems observed for highly non-normal matrices. The Fourier analysis in section 3.3 shows that we do not have to expect very non-normal matrices, and indeed, we have not seen these problems in practice.

The extra work required for the injection is limited, since there are no extra matrix-vector multiplications involved. We use GMRESR to make the injection possible, while we would otherwise use GMRES, this doubles the number of vector updates in the solver. There is also work involved in calculating $W = U^{\mathrm{H}}V$. If the dimension of V and U is k, this will cost k^2 inner products. To calculate the new V_0 and U_0 we also need k_0^2 vector updates, if we selected k_0 Ritz values. The other operations involve small matrices only

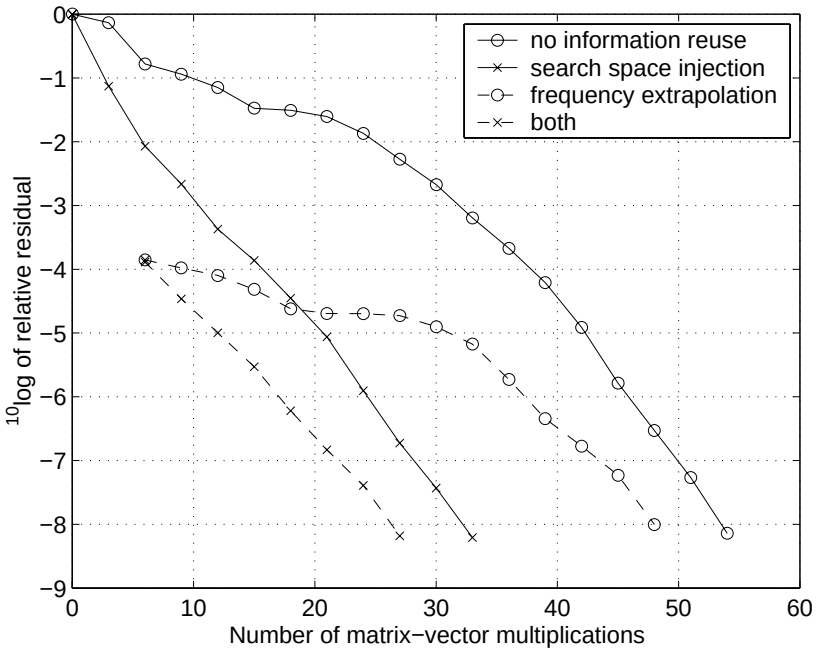


FIGURE 5.3: The result of reusing information for GMRESR preconditioned with MG with SAI smoother. All lines for 100MHz and 5450 unknowns. Reused frequency information from 3 previous frequencies (1MHz steps).

and will be relatively negligible. Note that the inner products and the vector updates can be combined in efficient BLAS-3 operations [16]. In practice, the work done for the selection will be small compared with one matrix-vector multiplication.

We can reuse the V_0 and U_0 calculated from the solve for the first right-hand side for all the following right-hand sides, but it is also possible to repeat the selection after each solve, in order to capture any directions that were missed in the first solve. In this case, we can save k_0^2 inner products for the next selection, by calculating $W_0 = U_0^{\mathrm{H}} V_0 = S^{\mathrm{H}} W S$.

The Ritz vector injection improves the robustness of the method. The removal of the stagnation phase in the convergence, as is seen in Figure 5.3, will reduce the CPU-time needed for the solution of the system and make it less dependent on the problem. Only for the first right-hand side we need to overcome the stagnation phase, but the other right-hand sides require an almost constant (small) number of iterations.

With this injection method, we can even remove the long stagnation from the SAI preconditioned solve, as is shown in Figure 5.4. However, to achieve this we first have to go through the stagnation phase in a (non-restarted) GMRESR. We then use the same selection criterion (5.11) for the Ritz value selection, which results in the injection of 91 Ritz vectors. By doing so, the second and further right-hand sides can be solved more quickly than with search space injection for the MG preconditioning, because the iterations cost only one matrix-vector multiplication per iteration. However, overcoming the stagnation in the SAI preconditioned solve for the first right-hand side is so expensive that this strategy will not be feasible, especially since the length of the stagnation phase is strongly dependent on the problem size.

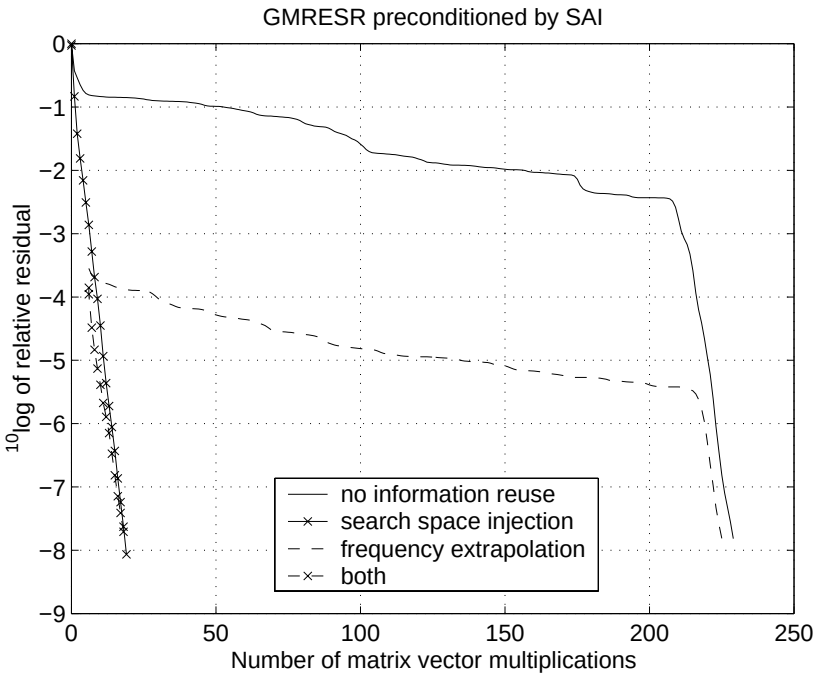


FIGURE 5.4: The result of reusing information for GMRESR with SAI preconditioning. All lines for 100MHz and 5450 unknowns. Reused frequency information from 3 previous frequencies (1MHz steps).

5.2 Frequency extrapolation

We have described how to reuse information obtained with GMRESR for previous right-hand sides, but there is more information that can potentially be reused. For many different frequencies, with each a different matrix A_Q, the same right-hand sides have to be solved. Since A_Q, and thus the solutions, change continuously with the frequency, we may be able to reuse information from the nearby frequencies.

A common technique to get the solution for a problem for a range of parameter values is by using continuation techniques, see for instance [44]. In general, this will increase the number of frequency steps in order to assure that the solution does not change too much per step. For our type of problems, this is not feasible, because for each frequency a new interaction matrix A (equation (2.33)) has to be calculated, which is very expensive. Using extra small frequency steps in order to accelerate the convergence of the iterative solver is unlikely to be advantageous.

What we can do is to try to derive a good initial guess from what we already know. The simplest way of reusing this information is to use the solution from the previous frequency step as an initial guess for the current solve. One step further would be to consider a short frequency history and extrapolate from that for an initial guess. This can be done at almost no cost, just a few vector updates, but it does not help much.

Another idea is to determine the optimal initial guess in the space spanned by the previous frequency solutions. This strategy is the "marching-on-in-frequency" approach described in [32]. In this case the "optimal" initial guess is found by minimising the

residual for the initial guess over the search space generated by the previous frequency solutions. To do this, we need to know the image under the new frequency matrix A_Q of these previous solutions, which costs a matrix-vector multiplication per direction. More explicitly: if our set of previous solutions is stored in the matrix X, we choose our initial guess $X\alpha$, where α minimises

$$\min_{\alpha} \|b - A_Q X\alpha\| \quad . \tag{5.12}$$

The corresponding residual is always less than or equal to the initial residual for standard extrapolation with a predetermined fixed α.

The above minimisation can also be included in the GMRESR process, by injecting X in the search space V. Since X contains approximate solutions for $A_Q x_Q = b_Q$ (3.43), while V would normally be the search space for the solution y of the preconditioned system $A_Q M_A y = b_Q$, injection of X in V would not work. By using left preconditioning, V becomes the search space for the solution X_Q of $M_Q A_Q X_Q = M_Q b_Q$, and X can be injected. In order to inject X in V, we would need to calculate the corresponding $U = M_Q A_Q X$. If we use multigrid preconditioning, this costs 3 matrix-vector multiplications per column of X. This is three times the cost of the initial residual minimisation in equation (5.2). Consequently, we do not inject X in the GMRESR process, but calculate the minimising initial guess x_0 using (5.2) and solve $A_Q M_Q y = b - A_Q x_0$, with $x = x_0 + M_Q y$.

The capacitive and inductive parts of the solution behave in a different way when the frequency changes, which means that the optimal coefficients α may differ for the capacitive and inductive parts. Fortunately, in Chapter 3, we have introduced a basis transformation that separates the important inductive terms from the capacitive terms. With respect to the new basis, a solution vector consists of two parts (x_l, x_c) that correspond to the inductive and capacitive currents. This can be used for the determination of separate factors for each of the two parts, by choosing the initial guess $(X_l\alpha_l, X_c\alpha_c)$, where α_l and α_c minimise

$$\min_{\alpha_l,\alpha_c} \left\| b - A \begin{pmatrix} X_l\alpha_l \\ X_c\alpha_c \end{pmatrix} \right\| \quad , \tag{5.13}$$

or, equivalently,

$$\min_{\alpha_l,\alpha_c} \left\| b - A \begin{pmatrix} X_l & 0 \\ 0 & X_c \end{pmatrix} \begin{pmatrix} \alpha_l \\ \alpha_c \end{pmatrix} \right\| \quad . \tag{5.14}$$

This will require twice as many matrix-vector multiplications, but will give a better result than the coupled minimisation.

Figure 5.5 shows the reduction of the initial residual due to frequency extrapolation using 3 previous frequencies for the test problem "complex". Experiments have shown that, for this problem, exploiting a longer frequency history did not significantly decrease the residual any further. The graphs show that, as we could have expected, the initial residual will not decrease below the relative residual stop criterion of the previous solutions, which was 10^{-8} for this example.

To see the result of this initial residual reduction, the third line (--○--) in Figure 5.3 shows the convergence graph for 100 MHz, using three previous frequencies with 1 MHz step size. The residual minimisation costs 6 matrix-vector multiplications, so the convergence graph starts with a horizontal offset of 6 matrix-vector multiplications. In this case the initial residual was reduced by approximately 3.5 orders of magnitude, but due

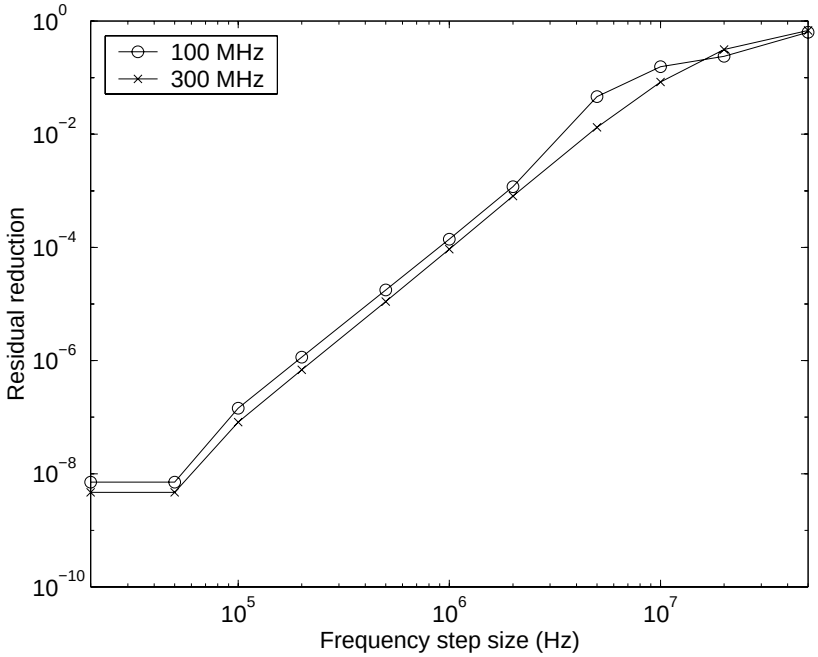


FIGURE 5.5: The result of minimising the initial residual using 3 previous frequencies and separated capacitive and inductive parts for test problem "complex" at 100 and 300 MHz.

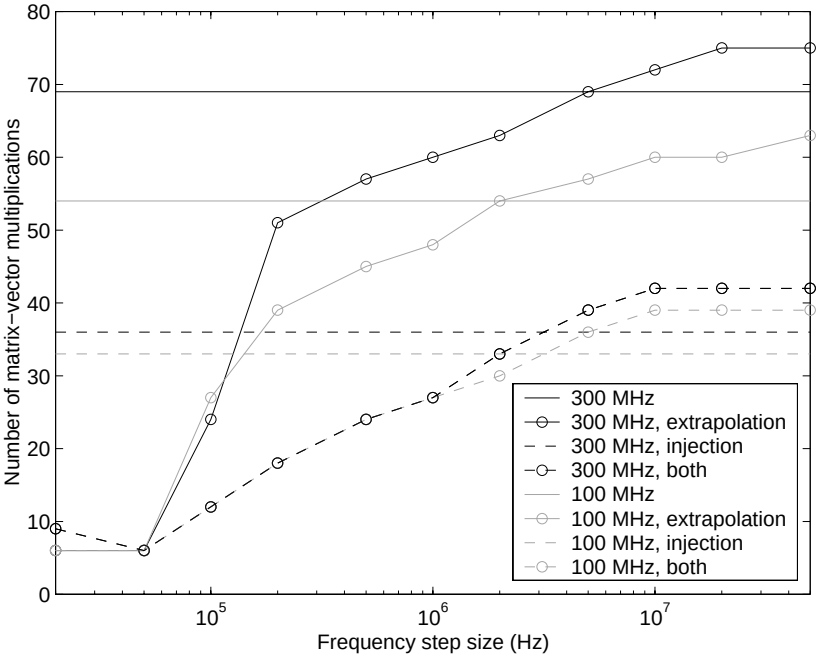


FIGURE 5.6: The number of matrix-vector multiplications needed for different combinations of injecting Ritz vectors and frequency extrapolation by minimising the initial residual, using 3 previous frequencies and separated capacitive and inductive parts for test problem "complex" at 100 and 300 MHz.

to the stronger initial stagnation, the final gain is only 2 iterations. The stronger initial stagnation shows that the new initial residual has relatively stronger components in the eigendirections that correspond to the eigenvalues that cause the stagnation. This means that the initial guess reduced the norm of the initial residual, but changed the direction for the worse.

In order to try to prevent this stagnation, we can combine the initial guess calculated from the previous frequency solutions with search space injection from previous right-hand side solves. For the second and further right-hand sides, it is possible to combine the initial guess strategy with the Ritz vector injection of section 1.2.5, in order to overcome initial stagnation. In this way, we reuse both the solutions of the previous frequencies, and the Ritz vectors of the previous right-hand side solves (at the current frequency). The results for our test problem are also shown in the bottom line (--×--) of Figure 5.3. It can be seen that the stagnation has been removed and convergence rate is almost as fast as for the second line (—×—). For these high convergence rates, the cost of the initial residual minimisation (equivalent with 2 iterations) is so high that the total savings amount to the cost of only 2 iterations. However, the combined strategies for reusing information result in a reduction of the number of iterations with a factor of 2. The frequency step size for this example was chosen quite small, for a much larger step size the advantage of the frequency extrapolation disappears.

Figure 5.6 shows the effect of different combinations of reusing information on the number of matrix-vector multiplications needed for one solve. Here we can see that for the large frequency steps, the extrapolation only increases the work, but for the very small frequency steps, it can help a lot. We can also see that, although the 300 MHz problem is harder to solve without the reusing of information, this difference is strongly reduced by search space injection. Also note that for the smallest frequency step sizes, the initial residualreduction by the frequency extrapolation is so strong, that only one or even no additional iterations are required.

If we apply the initial residual minimisation to the SAI preconditioned solve, as shown in Figure 5.4, the stagnation for the first right-hand side is so long that the reduction of the initial residual hardly helps. If, for the second right-hand side, search space injection is used, then the convergence rate is so high, that the costs of initial residual minimisation the are too high to get an overall gain with respect to search space injection only.

So far, for the first right-hand side we could not use Ritz vectors to overcome the stagnation. We have tried to inject harmonic Ritz vectors from previous frequencies, but this gave no significant improvement. The problem is that these are Ritz vectors for the previous preconditioned matrix, and are inaccurate for the new one. It appears that the outlying eigenvalues and vectors change significantly with the frequency, so that the injection of the old Ritz vectors does not help the new solve. If the frequency step is chosen so small that injecting the old Ritz vectos improves convergence, the matrix has changed so little, that it was not necessary to recalculate the solution for this frequency.

5.3 Conclusions

When solving for multiple right-hand sides, it can be effective to reuse available information about the matrix. Our strategy of using GMRESR and injecting selected harmonic

Ritz vectors from the previous solution process has shown to work well for our type of problem. The initial stagnation phase, observed when using standard GMRES(R), could be removed, leading to a fast and less problem dependent convergence.

The reusing of information from previous frequencies is more problematic. When using small frequency steps, a small gain can be achieved, also in combination with the injection of Ritz vectors from other right-hand sides. When the frequency step size is larger, this strategy will not lead to savings in computational work. The problem is that, in order to get a reduction of the initial residual, we have to invest some matrix-vector multiplications, which may not be compensated by the reduction in the number of iterations. As a result, this technique can be useful when using small frequency steps, but will not be helpful when using larger steps.

Chapter 6

Algebraic Multigrid

We have studied algebraic multigrid for application to transport equations on severely distorted meshes, and tried a different approach to the construction of the interpolation.

Since this new approach can be more easily applied to various types of matrices, we could also apply this method, with some adaptations, to the electromagnetic boundary integral problem. In this context, we are not aiming to get better convergence results than the geometric multigrid we used in chapter 4. A working algebraic multigrid method could be easier to implement and easier to integrate into existing codes, since it uses only the matrix equation, and does not require knowledge of the geometry. This also means that it cannot use knowledge of the geometry, and thus it has a disadvantage with respect to convergence speed if compared with geometric multigrid.

As we shall see, the convergence results of the new method are not yet very convincing, but we think that further exploration of the ideas behind the method is of interest, and makes it worthwhile to describe the current status.

6.1 Introduction

The geometric multigrid method, as described very shortly in section 4.1, is designed to solve *continuous* equations. It exploits several levels of increasingly coarser discretisations to achieve fast convergence. On each level there are two basic ingredients, the smoothing or relaxation of the error and the coarse grid correction that recursively uses the coarser levels. There is only limited freedom in choosing the coarse grid correction, since it is mostly determined by the geometry of the problem. The smoothing step should damp the parts of the error that are not damped by the coarse grid correction, which are the geometrically non-smooth errors. To achieve this, the smoother has to be adapted to complement the coarse grid correction.

The algebraic multigrid (AMG) method applies the same basic methods to solve a system of linear algebraic equations, i.e. a *matrix* equation, and does that fully automated. In order to do this, the matrix is regarded as if it represents a problem on a grid. Each point in this grid represents one unknown and the corresponding equation. In order to stress that this is a hypothetical grid, we will refer to this as "grid". The AMG method automatically determines several levels of coarsenings for the "grid" and suitable coarse "grid" correction mechanisms for these levels. In contrast with geometric

multigrid, the AMG smoothing step is considered to be fixed. The errors that are not damped by the smoother are called algebraically smooth errors. Since there is no relation to a geometry, this should not be seen as geometrically smooth, but as the type of error that is not damped by the smoother. The coarse "grid" correction must be chosen to complement the smoother and damp the algebraically smooth errors.

Ruge and Stüben have developed a method to generate coarse "grids" and a coarse "grid" correction [34]. For a more introductory text, see [39]. Their method is often used with good results, but it has some limitations. For this reason there are many variations on the Ruge-Stüben AMG. In this chapter, we will add another variant to the list.

In order to generate a coarse "grid" projection that is complementary to the smoother, it is necessary to know the form of the algebraically smooth errors. Ruge and Stüben use only the coefficients in the original equations (the matrix elements) to generate the coarse "grid" projection, thereby limiting the class of matrices and smoothers for which the method is effective. Their method is limited to real valued matrices with positive diagonal elements and negative off-diagonal elements. The method can still be used if there are some small positive off-diagonal elements, but large positive off-diagonal elements will strongly reduce the effectiveness. Another disadvantage of the Ruge-Stüben approach is that there is no obvious generalisation to complex matrices.

Since, in general, the form of the algebraically smooth error will not only depend on the original matrix equation, but also on the smoother, we propose to use them both in finding the behaviour of these algebraically smooth errors. We will propose a coarse "grid" correction algorithm using both the original equations and the smoother.

6.2 Algebraic multigrid framework

In this section we will describe the general AMG framework for solving linear matrix equations of the form $Ax = b$. Like multigrid, AMG is an iterative method, where each iteration is designed to reduce the error of an approximate solution. We will use the simplest variant, the so-called V-cycle iteration. The AMG V-cycle is the same as the multigrid V-cycle described in section 4.1. The only difference is in the way the coarse "grid" projections and the coarse "grid" matrix are constructed.

Each iteration in AMG starts with one or more smoothing steps, the so-called pre-smoothing :

$$\begin{aligned} x &\leftarrow x + Mr \\ e &\leftarrow e - Mr = e - MA\,e \equiv S_e e \ . \end{aligned} \tag{6.1}$$

In each smoother step, the error is multiplied with the smoother error operator

$$S_e = I - MA \ , \tag{6.2}$$

where M is called the smoother. M must be chosen so that S_e damp certain components of the error e. The most commonly used smoother is the Gauss-Seidel relaxation, in which case M is the inverse of the lower triangular part of A, including the diagonal.

The next step is the coarse "grid" correction. First a coarse "grid" must be chosen. Often this is a subset of the unknowns that are marked as coarse "grid" variables, but

this may also represent some other subspace. The residual must be projected on the coarse "grid", using the restriction operator R (6.3a). Next, the coarse "grid" error can be (approximately) computed by solving the coarse "grid" equation (6.3b), which is done by applying the V-cycle to the coarse "grid" equations, resulting in a recursive algorithm. The coarse error can be interpolated to the fine "grid" with the interpolation operator P and can be used to update the approximate solution (6.3c).

$$r_c = Rr \tag{6.3a}$$
$$e_r \approx (RAP)^{-1} r_c \tag{6.3b}$$
$$x \leftarrow x + Pe_c \tag{6.3c}$$
$$e \leftarrow e - Pe_c \approx e - P(RAP)^{-1}Rr \equiv K_e e \ . \tag{6.3d}$$

If the coarse "grid" solve (6.3b) is exact, the error operator associated with this coarse "grid" correction is given by

$$K_e = I - P(RAP)^{-1}RA \ . \tag{6.4}$$

After the coarse "grid" correction, again a number of smoother steps are applied, the so-called post-smoothing. In practice the coarse "grid" matrix RAP is still much too large to use a direct solver for the coarse "grid" equation (6.3b). Therefore, the coarse "grid" error is approximated by another AMG iteration for this smaller system, using smoothing and a coarse "grid" correction on an even coarser "grid". This is done recursively until the coarse "grid" system is small enough to be solved directly. This leads to the multi-level character of the AMG.

If we have only two levels, and if the coarse system (6.3b) is solved exactly on the second level, we speak of a 2-level method. For each V-cycle of the 2-level method, the error is multiplied with the combined error operator

$$T_e = S_e^{\beta} K_e S_e^{\alpha} \ , \tag{6.5}$$

where we used α pre-smoothing steps and β post-smoothing steps.

A pure AMG solver will repeatedly apply V-cycles to reduce the error :

$$e_k = T_e^k e_0 = U\Lambda^k U^{-1} e_0 \ , \tag{6.6}$$

with U an eigenbasis of T_e (assuming that this exists) and Λ a diagonal matrix with the eigenvalues $\lambda_1, \ldots, \lambda_n$ of T_e on the diagonal. This gives the error estimation

$$\|e_k\| = \|U\Lambda^k U^{-1} e_0\| \leq \kappa(U)(\max_i |\lambda_i|)^k \|e_0\| \ . \tag{6.7}$$

Equation (6.7) shows that all eigenvalues of T_e should be small in order to have fast convergence. This means that the smoothing steps and the coarse "grid" correction should complement each other: errors that are not damped by the smoother should be damped by the coarse "grid" correction and vice versa. In the AMG context, the smoother is treated as a given fixed operator, which means that the coarse "grid" correction must be adapted to make sure that all eigenvalues of T_e are small. This is done by choosing appropriate prolongation and restriction operators P and R.

The modes that are not damped by the smoother, the so-called algebraically smooth errors, should be damped by the coarse "grid" correction. From equation (6.4), we can see that K_e is a skew projection and has only eigenvalues 0 and 1. The corresponding eigenspaces are

$$E_0^{K_e} = \langle P \rangle \quad \text{and} \quad E_1^{K_e} = \langle (RA)^{\mathrm{T}} \rangle^{\perp} \ , \tag{6.8}$$

respectively, where $\langle \cdot \rangle$ denotes the span of the columns of the matrix. This can be verified by using

$$K_e P = P - P(RAP)^{-1} RAP = 0 \tag{6.9}$$

and

$$RA\langle (RA)^{\mathrm{T}} \rangle^{\perp} = 0 \ . \tag{6.10}$$

Since K_e has to damp the algebraically smooth errors, these errors should ideally be contained in the null space $E_0^{K_e} = \langle P \rangle$. To achieve this, we try to construct P such that this is the case. In section 6.6 we will say something about the choice of the restriction operator R.

6.3 The Ruge-Stüben approach

Within the general framework described in the previous section, the only thing that is still undetermined, is the construction of the projection matrices P and R. This has to be done at each level, but at each level the same method is used. In this section we will briefly describe the method that Ruge-Stüben use to construct the interpolation P and restriction R at each level.

First a subset C of the unknowns is selected for the coarse "grid", while the rest of the unknowns is called the fine "grid" subset F. This so-called coarse fine split is based on the notion of strong dependence. Ruge and Stüben define unknown i to be strongly dependent on the unknowns j in the set

$$S_i = \{ j \mid -A_{ij} \geq \theta \max_k -A_{ik} \} \ , \tag{6.11}$$

where $\theta < 1$ is some parameter that is often chosen to be $1/4$, and the minus signs appear because Ruge and Stüben assume that the important off-diagonal elements are negative. This definition is used to generate a coarse fine split such that every fine point depends strongly on at least one coarse point while trying to prevent strong coarse-coarse dependence. For an exact description of the coarse fine split, see section 4.6.2 of [34].

We will now assume that the equations and unknowns are ordered such that all the coarse unknowns and corresponding equations appear first

$$A = \begin{pmatrix} A_{CC} & A_{CF} \\ A_{FC} & A_{FF} \end{pmatrix} \ . \tag{6.12}$$

The interpolation P has to construct an error on the fine level from the error on the coarse level ($e = Pe_c$). Since the coarse level unknowns are the subset C of the fine level unknowns, the interpolation can just copy the coarse level errors to this subset of the fine level, $e_C = e_c$, leading to

$$P = \begin{pmatrix} I_{CC} \\ P_{FC} \end{pmatrix} \ . \tag{6.13}$$

The error on the fine "grid" unknowns F are interpolated from the coarse "grid" errors e_c by P_{FC}. This P_{FC} should be chosen such that the total complete fine level error e is algebraically smooth. The same structure will be assumed for the restriction

$$R = \big(I_{CC}, R_{CF}\big) \quad . \tag{6.14}$$

Ruge and Stüben devised a method to construct a P_{FC} using only information from A. The method is based on the assumption that, for a smooth error e,

$$A_{ii}e_i \approx -\sum_{j\neq i} A_{ij}e_j \quad . \tag{6.15}$$

Note that this can be interpreted as $Ae \approx 0$. In itself, this does not imply algebraic smoothness of the error e. However, if P is constructed to make Ae small, i.e., $AP \approx 0$, then the image space of this P will approximately contain the eigenvectors corresponding to the smallest eigenvalues of A. When using a Jacobi or Gauss-Seidel smoother, these directions are generally damped the least, and are thus algebraically smooth. Therefore, we might expect that $\langle P \rangle$ approximately contains the algebraically smooth errors. A much more detailed argumentation for (6.15) can be found in [34].

Relation (6.15) will be used to interpolate the fine "grid" errors e_i for $i \in F$ from the coarse errors e_j with $j \in C$. Using only the rows of $Ae \approx 0$ with a fine variable on the diagonal gives $A_{FF}e_F + A_{FC}e_C \approx 0$, which leads to the interpolation formula

$$e_F = -A_{FF}^{-1}A_{FC}e_C \quad , \tag{6.16}$$

which means $P_{FC} = -A_{FF}^{-1}A_{FC}$. If if this interpolation is used, in combination with $R = P^{\mathrm{T}}$ for symmetric A, the coarse "gird" projection will be an exact Schur decomposition of (6.12). Since this P is likely to be dense and expensive to compute, this is not practical. Instead, the relation (6.15) is used to construct an approximation to (6.16).

In (6.15), e_i depends on all the other error values e_j for which $A_{ij} \neq 0$, but the values e_j for which A_{ij} is large, i.e., $j \in S_i$, are the most important ones. The remaining connected errors are called weakly connected and are given by

$$W_i = \{j \neq i \mid A_{ij} \neq 0\} \setminus S_i \quad . \tag{6.17}$$

In order to remove the weakly connected error values from (6.15), they are replaced by the error e_i. This leads to the interpolation formula

$$\Big(A_{ii} + \sum_{j\in W_i} A_{ij}\Big)e_i = -\sum_{j\in S_i} A_{ij}e_j \quad , \tag{6.18}$$

which shows that the weak elements of A are lumped on the diagonal. This interpolation formula cannot yet be used since e_i may still depend on some strongly connected fine "grid" errors e_j with $j \in F_i \equiv S_i \cap F$. These unknown error values are replaced by the weighted average of the coarse errors that are strongly connected to both e_j and e_i :

$$e_j = \frac{\displaystyle\sum_{k\in C_i\cap C_j} A_{jk}e_k}{\displaystyle\sum_{k\in C_i\cap C_j} A_{jk}} \qquad \text{for} \qquad j \in F_i \quad . \tag{6.19}$$

Substitution of (6.19) in (6.18) expresses the fine "grid" errors e_i as linear functions of the strongly connected coarse "grid" errors e_j with $j \in C_i$. This defines an interpolation P_{FC}, with a sparsity pattern corresponding to the strong elements in A_{FC}. In the context of approximating (6.16), the expression (6.19) moves the strong elements of A_{FF} to the already existing strong elements of A_{FC}, thereby making the remainder of A_{FF} diagonal and thus easily invertible. Note that the replacement (6.19) can lead to breakdown due to devision by zero if not all strong off-diagonal elements of A_{FC} have the same sign. However, the definition (6.11) of strong connectivity ensures this.

For symmetric matrices, Ruge and Stüben choose the transpose of the interpolation matrix for the restriction, $R = P^{\mathrm{T}}$, and thereby approximating the Schur decomposition. See section 6.6 for some more thoughts about the restriction.

6.4 An alternative approach

If the problem matrix A and the smoother are not of the standard type, the arguments for using relation (6.15) to describe the behaviour of algebraically smooth errors are not valid. As an example, we consider the capacitive part of the transformed electromagnetic boundary element matrix (3.33). The eigenvectors corresponding to the small eigenvalues for this matrix are highly oscillatory, as follows from the Fourier analysis in section 3.3.4. However, in section 4.2.2 we argue that a smoother based on a truncated interaction, like the SAI smoother described in section 4.2.1, will damp the highly oscillating modes. In this situation the eigenvectors corresponding to the smallest eigenvalues of A will not be algebraically smooth, and the Ruge-Stüben approach cannot be expected to work well.

We propose to follow a different approach, and construct the interpolation operator P by using the smoother error operator S_e directly. We will adopt the structure (6.13) for P and also use the same sparsity pattern for P_{FC} as in the Ruge-Stüben approach.

In section 6.2 we argued that $\langle P \rangle$ should contain the algebraically smooth errors. To achieve this, we first have to investigate the behaviour of the algebraically smooth errors. Next, we need to construct a sparse P_{CF} such that algebraically smooth errors are approximately contained in the span of the columns of P.

We assume that for algebraically smooth errors, there is a strong linear correlation between each fine "grid" error value and the error values for the coarse unknowns that are strongly connected to to this fine unknown, as defined in (6.11). We will call this correlation the local behaviour of the algebraically smooth errors. This local behaviour for an algebraically smooth error e can be expressed in the following way

$$e_i \approx \sum_{j \in C_i} P_{ij} e_j \quad , \tag{6.20}$$

or in matrix notation

$$e_F \approx P_{FC} e_C \quad . \tag{6.21}$$

This is the form of interpolation that is also used in the Ruge-Stüben approach.

The errors that are damped the least by the smoother, are the eigenvectors that correspond to the largest eigenvalues of S_e, which means that these eigenvectors are algebraically smooth, and the local behaviour is reflected in these eigenvectors. These eigenvectors often have a different global shape but the same local behaviour.

We consider the simple example of a regular finite difference discretisation of the Poisson operator on the square with Dirichlet boundary conditions in combination with a (damped) Jacobi smoother. For this case, the eigenvectors that correspond to the largest eigenvalues of S_e are identical to the eigenvectors that correspond to the smallest eigenvalues of A. They correspond to the well known smooth functions with value zero on the sides of the square and one maximum in the centre for the first eigenvector, one maximum and one minimum for the second and third eigenvectors respectively, etc. In this example, the algebraically smooth local behaviour can be characterised as geometrical smoothness. We clearly see that the eigenvectors that correspond to the largest eigenvalues of S_e are different global modes with the same local behaviour (geometric smoothness). In this example, the algebraically smooth local behaviour is the same over the whole domain. In general this is not necessary and we allow different interpolation coefficients for different fine unknowns.

The presence of a common local behaviour for algebraically smooth errors is not uncommon for matrices that originate from continuous equations in combination with a common smoother like Gauss-Seidel. We will use this local behaviour for the construction of the interpolation matrix P.

6.5 The interpolation

Suppose we have one algebraically smooth error vector s at our disposal. In section 6.5.3, we will discuss a method to obtain such a smooth vector. We will try to use this vector to identify the local behaviour and construct the interpolation P. The basic idea is to make the interpolation exact for s. By this, we mean that $Ps_C = s$, which implies that

$$P_{FC} s_C = s_F \ , \tag{6.22}$$

or, equivalently,

$$\sum_{k \in C_i} P_{ik} s_k = s_i \ . \tag{6.23}$$

In this section, $i \in F$ and $k, l \in C_i$. Note that, in general, the condition (6.22) does not define P_{FC} uniquely.

6.5.1 Linear interpolation

We start with a "linear" interpolation and then adapt this to make it exact for s. The linear interpolation computes the fine errors by averaging the strongly connected coarse errors weighted by the absolute value of the connecting element in A, which leads to

$$\bar{P}_{ik} = \frac{|A_{ik}|}{\sum_l |A_{il}|} \qquad i \in F \ , \quad k, l \in C_i \ . \tag{6.24}$$

The interpolation is linear in the sense that, if there is an underlying geometry, and if the absolute matrix elements $|A_{ik}|$ are inversely proportional to the distances $|\mathbf{x}_i - \mathbf{x}_k|$

between the location of the unknowns, and if the directions in the grid have a symmetry such that

$$\sum_{k\in C_i} \frac{\mathbf{x}_k - \mathbf{x}_i}{|\mathbf{x}_k - \mathbf{x}_i|} = 0 \ , \tag{6.25}$$

then geometrically linear functions will be interpolated exactly by this $\bar{P}$. To see this, let $f_i = f(\mathbf{x}_i)$, with $f(\mathbf{x})$ a first order polynomial of $\mathbf{x}$, then

$$\sum_k \bar{P}_{ik} f_k = f\Big(\sum_k \bar{P}_{ik}\mathbf{x}_k\Big) = f\Big(\mathbf{x}_i + \sum_k \bar{P}_{ik}(\mathbf{x}_k - \mathbf{x}_i)\Big) \ , \tag{6.26}$$

where we used that $\sum_k \bar{P}_{ik} = 1$. By using definition (6.24) and the symmetry condition (6.25), we find that

$$\sum_k \bar{P}_{ik}(\mathbf{x}_k - \mathbf{x}_i) = \sum_k \frac{|A_{ik}|}{\sum_l |A_{il}|}(\mathbf{x}_k - \mathbf{x}_i) = \frac{1}{\sum_l 1/|\mathbf{x}_i - \mathbf{x}_l|} \sum_k \frac{\mathbf{x}_k - \mathbf{x}_i}{|\mathbf{x}_k - \mathbf{x}_i|} = 0 \ . \tag{6.27}$$

Substituting this in (6.26) shows that the interpolation is exact :

$$\sum_k \bar{P}_{ik} f_k = f(\mathbf{x}_i) = f_i \ . \tag{6.28}$$

Note that the constant vector $(1, 1, \ldots, 1)^{\mathrm{T}}$ is always interpolated exactly by $\bar{P}$.

The linear interpolation (6.24) is sometimes close to the Ruge-Stüben interpolation, which uses

$$\widetilde{P}^{\mathrm{RS}}_{ik} = \frac{A_{ik}}{-A_{ii}} \tag{6.29}$$

as the basic interpolation, see equation (6.15) and further. If A is real valued, and if the strong off-diagonal elements A_{ik} have the same sign, and if $\sum_j A_{ij} = 0$, then $\widetilde{P}^{\mathrm{RS}} = \bar{P}$.

6.5.2 Adaptation

We adapt the linear interpolation $\bar{P}$ to make it exact for an algebraically smooth error s. The difference between $s_{_F}$ and $\bar{s}_{_F} \equiv \bar{P}_{_{FC}} s_{_C}$ is distributed over the corresponding entries of $P_{_{FC}}$ with some scaling factors w_{ik},

$$P_{ik} = \bar{P}_{ik} + \frac{w_{ik}}{\sum_l w_{il}} \, \frac{s_i - \bar{s}_i}{s_k} \ . \tag{6.30}$$

This P will interpolate s exactly, which can be seen by verifying relation (6.23),

$$\sum_k P_{ik} s_k = \bar{s}_i + \frac{\sum_k w_{ik}}{\sum_l w_{il}} \frac{s_i - \bar{s}_i}{s_k} s_k = \bar{s}_i + (s_i - \bar{s}_i) = s_i \ . \tag{6.31}$$

We can still choose the scaling factors w_{ik}. We want to distribute the correction of P over the available elements of P, weighted only by the connection strength $|A_{ik}|$. To achieve this, we choose

$$w_{ik} = |s_k| \, |A_{ik}| \ . \tag{6.32}$$

The $|s_k|$ will cancel the norm of the $1/s_k$ in the correction (6.30), preventing that elements P_{ik} corresponding to small s_k are changed more than those with large s_k. By performing this cancellation explicitly, this also avoids breakdown for $s_k = 0$.

If, for some row i of P, the sign of the s_k (for complex values the argument of the s_k) in equation (6.30) are all the same, then

$$P_{ik} = \bar{P}_{ik} + \frac{|s_k|\,|A_{ik}|}{\sum_l |s_l|\,|A_{il}|}\frac{s_i - \bar{s}_i}{s_k} \tag{6.33a}$$

$$= \frac{|A_{ik}|}{\sum_{l'} |A_{il'}|} + \frac{|A_{ik}|}{\sum_l |A_{il}|s_l}(s_i - \bar{s}_i) \tag{6.33b}$$

$$= |A_{ik}|\,\mu_i = \nu_i \bar{P}_{ik} \quad . \tag{6.33c}$$

In (6.33c), we collected all terms that are independent of k in μ_i, and using the definition of $\bar{P}$ in (6.24), we see that the ith row of P is a scaling of the corresponding row in $\bar{P}$. Since P interpolates s exactly, the scaling factor ν_i must be equal to $s_i/\bar{s}_i$. Some further manipulation with (6.33b), confirms this. Thus, if $\text{sign}(s_k)$ is the same for all $k \in C_i$, then

$$P_{ik} = \frac{s_i}{\bar{s}_i}\bar{P}_{ik} \quad , \tag{6.34}$$

which is much simpler than (6.30). However, we do not use the scaling (6.34) for the correction of $\bar{P}$, since this breaks down if $\bar{s}_i = 0$, which can only happen if there are corresponding values s_k with opposite sign. We can use the scaling (6.34) to get a better feeling for what the correction (6.30) does if $\text{sign}(s_k)$ is the same for all $k \in C_i$.

6.5.3 Algebraically smooth vector

To be able to apply the correction (6.30), we have to find an algebraically smooth vector representative for the algebraically smooth local behaviour. In section 6.4, we argued that the eigenvectors corresponding to large eigenvalues of S_e are algebraically smooth, which makes these eigenvectors good candidates for the smooth vector s. We approximate the eigenvector corresponding to the largest eigenvalue of S_e with the Arnoldi method [35, 41]. If the two largest eigenvalues are well separated (in a relative sense), this will converge quickly, but if the separation is small, convergence will be slow. However, we do not necessarily need the eigenvector corresponding to largest eigenvalue. A linear combination of the eigenvectors corresponding to the large eigenvalues is also algebraically smooth. This means that usually a few steps of Arnoldi suffice to get a smooth vector s. The interpolation based on this s will be called the *eigen interpolation* in this thesis.

If, for some problem, we may expect geometrically smooth behaviour, we can also select the constant vector $(1, 1, \ldots, 1)^{\mathrm{T}}$ for the algebraically smooth vector s. This vector is already interpolated exactly by the linear interpolation $\bar{P}$, so there will be no correction and the interpolation will be the linear interpolation (6.24).

6.6 The restriction

In section 6.2, we saw that the 2-level coarse "grid" error operator

$$K_e = I - P(RAP)^{-1}RA \tag{6.35}$$

is a skew projection with eigenvalues 0 and 1, and eigenspaces

$$E_0^{K_e} = \langle P \rangle \quad \text{and} \quad E_1^{K_e} = \langle (RA)^{\mathrm{T}} \rangle^{\perp} \ , \tag{6.36}$$

respectively. If the angle between the eigenspaces is small, the projection is very skew, and K_e will have large singular values, which means that some errors will lead to a larger error after the coarse "grid" correction. For an optimal error reduction, we would like K_e to have only singular values 0 and 1 as well. To accomplish this, the projection K_e must be an orthogonal projection, which means that the eigenspaces $E_0^{K_e}$ and $E_1^{K_e}$ have to be orthogonal, leading to the requirement

$$\begin{aligned} \langle (RA)^{\mathrm{T}} \rangle^{\perp} \perp \langle P \rangle \quad &\Leftrightarrow \quad \langle (RA)^{\mathrm{T}} \rangle = \langle P \rangle \quad \Leftrightarrow \\ A^{\mathrm{T}} \langle R^{\mathrm{T}} \rangle = \langle P \rangle \quad &\Leftrightarrow \quad \langle R^{\mathrm{T}} \rangle = A^{-\mathrm{T}} \langle P \rangle \ , \end{aligned} \tag{6.37}$$

where we used that R^{T} and P have the same dimensions.

Note that the eigenspaces $E_0^{K_e}$ and $E_1^{K_e}$ together fully determine K_e. This means that changing P and R such that $\langle P \rangle$ and $\langle R^{\mathrm{T}} \rangle$ are unchanged, will not alter K_e. The requirement that $\langle P \rangle$ contains the smooth errors, together with the orthogonality condition (6.37), is all that matters for the 2-level method. The remaining freedom can be used for practical aspects like sparseness of P and R and for getting a convenient coarse "grid" operator RAP for multilevel algebraic multigrid.

The question remains what to do with condition (6.37). In general, multiplication with A^{-1} will change the direction of a vector toward the eigenvectors corresponding to the small eigenvalues of A. If these eigenvectors of A correspond to the algebraically smooth vectors (the eigenvectors corresponding to the large eigenvalues of S_e that we tried to capture in $\langle P \rangle$), then multiplication with A^{-1} will make a vector more algebraically smooth. If $A = A^{\mathrm{T}}$, then the ideal $\langle R^{\mathrm{T}} \rangle = A^{-\mathrm{T}} \langle P \rangle = A^{-1} \langle P \rangle$ is more smooth than $\langle P \rangle$. Since we already did our best to capture the smooth vectors in $\langle P \rangle$, we have nothing better than to choose than $\langle R^{\mathrm{T}} \rangle = \langle P \rangle$, or $R^{\mathrm{T}} = P$. This is a very common choice for symmetric matrices, and has the advantage that the coarse "grid" matrix RAP is again symmetric.

The assumption that the eigenvectors corresponding to the small eigenvalues of A correspond to the algebraically smooth vectors is not unnatural, since standard smoothers like the Jacobi or Gauss-Seidel iterations usually damp these eigenvectors the least. However, if this assumption is not valid, choosing $R^{\mathrm{T}} = P$ might not be a wise choice. Because of a lack of better ideas, we will still use $R^{\mathrm{T}} = P$ if we know that this assumption is not valid, like in the case of the electromagnetic boundary element matrices.

6.7 Experimental results for transport problems

In this section, we present some results for linear systems resulting from diffusive transport equations. The matrices were generated with the Augustus package [23]. The

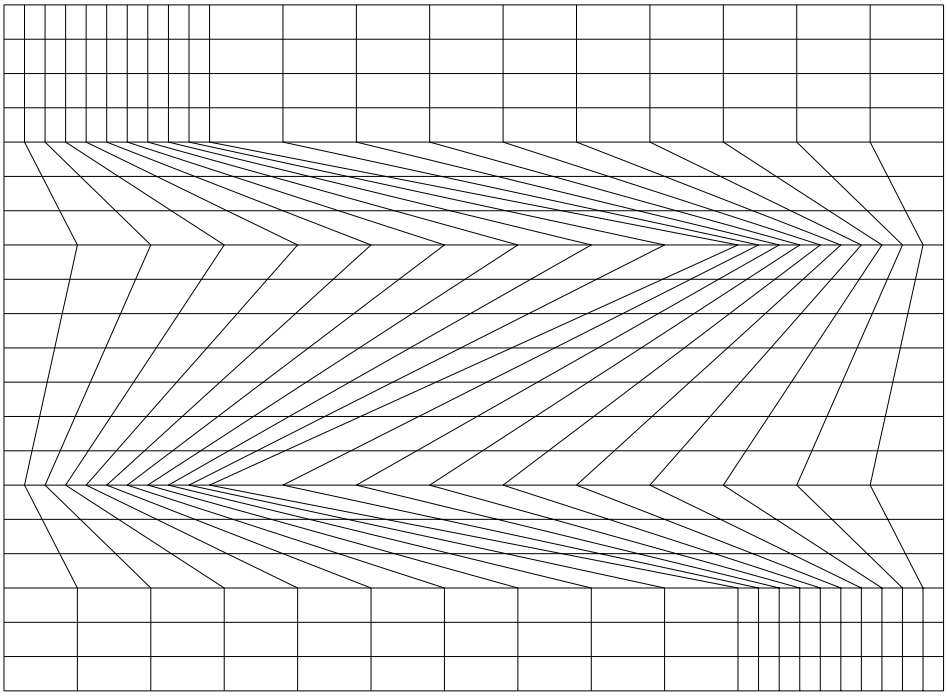


FIGURE 6.1: The 2-dimensional 20×20 Kershaw mesh.

equations are of the type

$$\begin{aligned} \alpha \frac{\partial \Psi}{\partial t} + \nabla \cdot \mathbf{F} + \sigma \rho &= S \\ \mathbf{F} &= -D \nabla \rho \ , \end{aligned} \tag{6.38}$$

with $\mathbf{F}$ the flow, Ψ the density, D the diffusion coefficient, α the time derivative coefficient, σ the removal coefficient and S a source term. These equations are discretised on a finite element grid with a local support-operator diffusion discretisation scheme [29], leading to a symmetric matrix. The numerical unknowns represent the densities Ψ at the cell centres and the cell faces. The matrices we used do not come from real applications, but are used as representative test problems. These test problems have a square domain with reflective, absorbing and fixed flow boundary conditions on the different sides.

We use two different sets of matrices. The first set comes from a problem involving two materials resulting in a jump in the diffusion coefficient of a factor 10^6 while using a moderately distorted regular grid. This grid is based on a regular square grid, which grid points are randomly moved over a maximum distance of one quarter of the grid size. For this problem set, all diagonal elements are positive and all off-diagonal elements are negative.

The second set comes from the constant coefficient problem, discretised on an extremely distorted mesh with small angels. We used a Kershaw mesh [27], for which an example can be seen in Figure 6.1. The small angles in the mesh lead to large positive off-diagonal elements in the matrix.

	two materials			Kershaw mesh		
size	320	1200	11000	320	1200	11000
Linear	0.401	0.500	0.581	0.42	0.57	0.75
Eigen Arnoldi	0.015	0.034	0.125	0.42	0.61	0.74
Eigen exact	0.015	0.032	0.105	0.42	0.62	0.75
R-S	0.015	0.026	0.086	0.41	0.59	0.75

TABLE 6.1: Measured asymptotic convergence factors for transport problem using 2 Gauss-Seidel pre-smoothing and post-smoothing steps.

For these problems, we have tried to measure the asymptotic convergence factor

$$c = \lim_{k\to\infty} \frac{\|r_{k+1}\|}{\|r_k\|} \quad . \tag{6.39}$$

Using (6.6), we see that

$$r_k = Ae_k = AT_e^k e_0 = AT_e^k A^{-1} r_0 = T_r^k r_0 \quad , \tag{6.40}$$

where we used the V-cycle error operator T_e of (6.5) and define the accompanying V-cycle residual operator $T_r \equiv AT_eA^{-1}$. This shows that the asymptotic convergence factor is equal to the largest eigenvalue of T_r, and since the spectra of T_r and T_e are equal, we get

$$\begin{aligned} c &= \lim_{k\to\infty} \frac{\|r_{k+1}\|}{\|r_k\|} = \max_{\lambda\in\sigma(T_r)} |\lambda| \\ &= \lim_{k\to\infty} \frac{\|e_{k+1}\|}{\|e_k\|} = \max_{\lambda\in\sigma(T_e)} |\lambda| \quad . \end{aligned} \tag{6.41}$$

To approximate this, we iterated AMG until convergence ($\|r_k\|/\|r_0\| < 10^{-6}$) with a maximum of 20 iterations and took the ratio between the last two residual norms as an approximation for the asymptotic convergence factor.

Table 6.1 shows these measured asymptotic convergence factors for the two test problems for different matrix sizes and interpolation methods. All methods used the standard Ruge-Stüben coarse fine split, the same sparsity pattern for P_{FC}, and used $R^{\mathrm{T}} = P$. The "Linear" interpolation is defined in (6.24). The "Eigen Arnoldi" interpolation uses the correction (6.30), where the algebraically smooth error vector s is found with 8 steps of Arnoldi on S_e. The "Eigen exact" interpolation uses the same correction (6.30), but now the algebraically smooth error vector s is a high accuracy approximation to the eigenvector corresponding to the largest eigenvalue of S_e, which we computed with the Jacobi-Davidson (JD) method [38]. Since all experiments were done in MATLAB [40], we could use the MATLAB JDQR package [37]. This method is intended only as a diagnostic tool since running the JD process to full convergence is often more expensive than solving the linear system. The "R-S" method stands for the Ruge-Stüben interpolation.

From Table 6.1 we can see that, for the two material problem, the Ruge-Stüben approach works very good, and there is no direct need for an alternative method. The eigen correction (6.30) works very nice and is almost as effective as the Ruge-Stüben

method. The Kershaw mesh problems appear to be much harder, also for the Ruge-Stüben approach. For these matrices, there was no significant difference between the different methods. The correction (6.30) had no significant effects when using this mesh.

The absence of significant differences between the "Eigen Arnoldi" and the "Eigen exact" methods, shows that the "Eigen Arnoldi" method did not suffer much from the lack of precision in the eigenvector approximation. However, for increasing problem size, the increasing lack of accuracy of the Arnoldi result may eventually have a negative effect on the convergence factor. The small difference between the "Eigen Arnoldi" and the "Eigen exact" method for the largest problem size, as seen for the two material problem, may be an indication of this effect.

6.8 Adaptation for A_Q

In this section we describe the extra adaptations that we have made in order to make the AMG algorithms practical for the linear system (3.43). Problems arise due to the fact that the system matrix A_Q (3.33) is complex valued, dense, not explicitly available, has two different types of variables, and is singular.

The definition (6.11) for strong dependence cannot be used for complex matrices, so we use a variation of this definition

$$\widetilde{S}_i = \{j \mid |A_{ij}| \geq \theta \max_k |A_{ik}|\} \quad . \tag{6.42}$$

The weighted average used in equation (6.19) should not be used for this type of matrices either. The numerator might become small, leading to very large elements in P. To prevent this, we replace equation (6.19) with

$$e_j = \frac{\sum_{k \in C_i \cap C_j} |A_{jk}| e_k}{\sum_{k \in C_i \cap C_j} |A_{jk}|} \qquad \text{for} \qquad j \in F_i \quad . \tag{6.43}$$

Although the AMG algorithms were not specifically designed for linear systems with dense matrices, it is possible to apply AMG to dense linear systems. Since the size of the matrix elements in A_Q does not decrease very rapidly for increasing distance, the often used value $\theta = 1/4$ will result in very large sets $\widetilde{S}_i$ (6.42). In the coarse fine split, this will result in a very small number of coarse variables C, which will lead to very bad convergence. This forces us to choose a larger value for θ. We have used $\theta = 3/4$.

The matrix A_Q is not stored explicitly, only a small part A_Q^{sp}, corresponding to the small distance interactions, is computed explicitly for the construction of the smoother, as is described in section 4.2. We will use this sparse A_Q^{sp} to determine the strong dependencies S_i and thereby the coarse fine split. For the Ruge-Stüben approach, we also use this sparse matrix to determine the prolongation and restriction operators P and R. For the "eigen" interpolation, the full matrix A_Q is used in the Arnoldi iteration.

An extra complication is due to the fact that we have two distinct types of variables, the loop variables and the quasi-charge variables (see section 3.2). This can complicate

the selection of a good coarse fine split and good interpolation and restriction operators. To resolve this, we constructed independent coarse fine splits and interpolation and restriction operators for the different types of variables, as we did for the geometric multigrid in section 4.3. This is also suggested in [34, section 4.7.1]. This is easily implemented by temporarily ignoring all interaction between the loop and quasi-charge variables in A_Q^{sp}. In the case of the eigen interpolation, we determine separate approximate eigenvectors corresponding to the diagonal blocks of A_Q belonging to the different types of variables.

The variables corresponding tot the global loops will always be selected in the coarse sets C, and thus be resolved in the direct solve on the coarsest level. This is the same as we did for geometric multigrid. We implemented this by making each global loop a separate variable type, an treating them in the way we described in the previous paragraph.

Experiments showed, that the coarse fine split for the quasi-charge variables was very poor. This could be traced back to the fact that we truncate the quasi-charge basis functions at some distance (see Figure 3.5), which results in an interaction strength that does not decrease monotonically with increasing distance. This may lead to situations where neighbouring variables are not considered to be strongly connected in the sense of (6.42). In order to prevent this, we temporarily replaced the elements of A_Q^{sp} in the capacitive diagonal block with the corresponding elements in the electrostatic matrix D, defined in equation (3.39). This substitution is motivated by the fact that the quasi-charge currents in the basis Q are designed to let the capacitive block of A_Q approximate D, as described in section 3.2.

Note that the above alterations to A_Q^{sp} only influence the construction of P and R. In the AMG iteration the full matrix A_Q is used as the linear system matrix and the unchanged A_Q^{sp} is used for the smoother.

6.9 Experimental results for electromagnetic boundary integral equations

With the adaptations described in section 6.8, we have applied AMG for the "complex" test problem (Figure 4.5) that we also used in the geometric multigrid experiments of section 4.5. Due to technical limitations of our implementation, we were unable to use the same range of problem sizes that we used for geometric multigrid. Since our implementation is not optimised, we have not measured CPU-times, but only convergence factors and iterations counts.

Table 6.2 shows the measured asymptotic convergence factors (see section 6.7) for the "complex" test problem of several sizes at three frequencies using different interpolation methods. The "Smoother only" method indicates that the course "grid" correction step was skipped, which means that we only do two smoother steps: one pre-smoothing and one post-smoothing step. These results are shown in order to be able to see the effect of the different coarse "grid" corrections. The convergence factors larger than 1 show that for almost all cases AMG is diverging.

Although AMG itself is diverging, the AMG V-cycle might still be a good precondi-

	Asymptotic convergence					MG prec. GMRES iterations				
problem size	321	673	1431	2103	2901	321	673	1431	2103	2901
frequency	30 MHz									
Smoother only	1.00	1.00	1.00	0.98	1.04	24	34	54	70	>70
Adapted R-S	1.00	1.00	1.00	0.98	1.05	22	32	49	66	>70
Linear	1.79	1.61	1.13	1.01	1.07	11	15	21	34	34
Eigen Arnoldi	3.06	1.32	1.04	1.00	1.06	11	15	22	36	38
Geometric MG						9	10	13	14	15
frequency	100 MHz									
Smoother only	1.14	1.08	1.04	1.01	1.04	26	37	57	>70	>70
Adapted R-S	1.17	1.08	1.06	1.02	1.04	23	35	53	70	>70
Linear	2.05	2.19	2.56	1.33	3.29	13	17	23	38	37
Eigen Arnoldi	2.78	1.53	1.62	1.34	2.05	18	26	25	39	53
Geometric MG						9	10	14	16	16
frequency	300 MHz									
Smoother only	3.80	1.43	1.18	1.10	1.08	65	63	>70	>70	>70
Adapted R-S	4.14	6.71	4.97	1.12	124	55	49	>70	>70	>70
Linear	2.06	82.7	5.61	9.76	1.90	41	35	34	49	45
Eigen Arnoldi	29.7	6.85	29.0	11.0	2.08	46	46	61	>70	>70
Geometric MG						24	19	17	17	18

TABLE 6.2: Measured asymptotic convergence factors and the number of multigrid preconditioned GMRES iterations needed for a relative residual less than 10^{-8} for different size discretisations of the test problem "complex" at different frequencies.

tioner for GMRES. The number of V-cycle preconditioned GMRES iterations needed for a relative residual less than 10^{-8} is also shown in Table 6.2. Comparison of the "Smoother only" and the "Adapted G-S" results shows that the adapted Ruge-Stüben coarse "grid" correction does not contribute much to the convergence speed. For these problems, the linear interpolation is the best of the AMG variants. The "eigen" correction to the linear interpolation improves the asymptotic convergence factor for some cases, but as a preconditioner it is worse for almost all cases. We have also done experiments with more accurate approximations to the eigenvector corresponding to the largest eigenvalue of S_e (obtained with the JDQR MATLAB package [37]). These experiments showed that the increased accuracy leads to an equal or even larger number of iterations. This indicates that the negative results of the "eigen" correction are not due to a lack of accuracy of the approximation of the smooth eigenvector.

6.10 Discussion

The Ruge-Stüben approach for AMG gives good results for a certain class of problems. An example of such a problem is the two material problem with the slightly distorted regular mesh described in section 6.7. However, we saw that the same equations, discretised on the very distorted Kershaw mesh, lead to much worse results for Ruge-Stüben AMG.

It is even impossible to apply the Ruge-Stüben approach to complex matrices without adaptation, and we saw that for the electromagnetic boundary integral equations this adapted Ruge-Stüben AMG does not work very well either.

In the Ruge-Stüben approach for AMG, the coarse "grid" correction is independent of the smoother. Instead, some assumptions are made about the behaviour of algebraically smooth errors. For the class of problems it was developed for, these are valid assumptions, but they do not apply for other problems.

Our idea was to construct an alternative coarse "grid" correction that also depends on the smoother, and might thus work for a larger class of problems. We implemented this idea by using a linear interpolation and correcting this using a known smooth vector, for which we used an approximation to the eigenvector corresponding to the smallest eigenvalue of the smoother error operator S_e.

Unfortunately, this method did not give the desired results. For the transport problems of section 6.7, the results obtained with this new eigen interpolation were approximately the same as for the Ruge-Stüben approach. For the electromagnetic boundary integral problems, the linear interpolation worked best, and the eigen correction only deteriorated the convergence. For these dense and complex valued problems, the positive result is that the linear interpolation leads to a reasonable AMG based preconditioner.

We are of the opinion that there exist other implementations of this idea that lead to better results. There may be better correction methods, although we already tried several with the same or worse results. More improvement may be in the identification of the local algebraically smooth behaviour. Other improvements may result from a better (smoother dependent) coarse fine split.

Bibliography

[1] E. Anderson, Z. Bai, C. Bischof, L. S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*, SIAM, Philadelphia, 1999. Html version available at `http://www.netlib.org/lapack/lug/`.

[2] D. M. Arnold, R. S. Falk, and R. Winter, *Preconditioning in $H(\mathrm{div})$ and applications*, Math. Comp., 66 (1997), pp. 957–984.

[3] C. A. Balanis, *Antenna theory; analysis and design*, Harper & Row series in electrical engineering, Harper & Row, New York, 1st edition ed., 1982.

[4] ——, *Advanced Engineering Electromagnetics*, John Wiley & Sons, Nov. 1990.

[5] J. E. Barnes and P. Hut, *A hierarchical $\mathcal{O}(N \log N)$ force-calculation algorithm*, Nature, 324 (1986), pp. 446–449.

[6] R. Barret, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, C. Romine, and H. A. van der Vorst, *Templates for the Solution of Linear Systems : Building Blocks for Iterative Methods*, SIAM Publications, 1993.

[7] M. Benzi and M. Tuma, *A sparse approximate inverse prteconditioner for nonsymmetric linear systems*, SIAM Journal of Scientific Computing, 19 (1998), pp. 968–994.

[8] ——, *A comparative study of sparse approximate inverse preconditioners*, Applied Numerical Mathematics: Transactions of IMACS, 30 (1999), pp. 305–340.

[9] J. R. Bergervoet, *Some experiments in solving large electromagnetic systems*. Unpublished.

[10] J. R. Bergervoet, G. Maas, and M. J. C. M. van Doorn, *The common-mode skeleton model for assessment of electromagnetic compatibility at the system-level*, in Proc. 12th Int. Zürich Symp. on EMC, 1997.

[11] L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. C. Whaley, *ScaLAPACK Users' Guide*, SIAM Publishing, Philadelphia, 1997.

[12] A. Brandt, W. Joppich, J. Linden, G. Lonsdale, and A. Schuller, *Multigrid Course*, GMD–690, Gesellschaft für Mathematik und Datenverarbeitung, St. Augustin, 1992.

[13] T. C. Chan and H. van der Vorst, *Approximate and incomplete factorizations*, in Parallel Numerical Algorithms, D. E. Keyes, A. Samed, and V. Venkatakrishnan, eds., ICASE/LaRC Interdisciplinary Series in Science and Engineering, Kluwer Academic, Dordecht, 1997, pp. 167–202.

[14] R. Coifman, V. Rokhlin, and S. Wandzura, *The fast multipole method for the wave equation: A pedestrian prescription*, IEEE Antennas and Propagation Magazine, 35 (1993), pp. 7–12.

[15] D. Colton and R. Kress, *Integral equation methods in scattering theory*, John Wiley & Sons, New York, 1983.

[16] J. J. Dongarra, J. Du Croz, S. Hammarling, and I. S. Duff, *A set of level 3 basic linear algebra subprograms*, ACM Transactions on Mathematical Software, 16 (1990), pp. 1–17. Software available at `http://www.netlib.org/blas/`.

[17] S. C. Eisenstat, H. C. Elman, and M. H. Schultz, *Variational iterative methods for nonsymmetric systems of linear equations*, SIAM J. Numer. Anal., 20 (1983), pp. 345–356.

[18] R. W. Freund and M. Malhotra, *The block-QMR method for the solution of multiple radiation and scattering problems in structural acoustics*, in 15th IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics, Vol. 3, Computational Physics, Chemistry and Biology, A. Sydow, ed., Wissenschaft und Technik Verlag, 1997, pp. 461–466.

[19] L. Greengard and V. Rokhlin, *A fast algorithm for particle simulations*, J. Comput. Phys., 73 (1987), pp. 325–348.

[20] M. J. Grote and T. Huckle, *Parallel preconditioning with sparse approximate inverses*, J. Sci. Comp., 18 (1997).

[21] W. Hackbusch, *Multi-grid methods and applications*, Springer Verlag, Berlin, 1985.

[22] ——, *Integral equations: theory and numerical treatment*, Birkhäuser, Basel, 1995.

[23] M. L. Hall. The Augustus Code Package, information available at `http://www.lanl.gov/Augustus/`.

[24] R. Hiptmair, *Multigrid method for $H(\mathrm{div})$ in three dimensions*, Elect. Trans. Num. Anal., 6 (1997).

[25] International Electrotechnical Commission, *Electromagnetic compatibility, the role and contribution of IEC standards*, 1999. Available at `http://www.iec.ch/Onlinepubs/EMC.pdf`.

[26] J. Jackson, *Classical Electrodynamics*, John Wiley & Sons, New York, 1975.

[27] D. S. Kershaw, *Differencing of the diffusion equation in Lagrangian hydronamics codes*, Journal of Computational Physics, 39 (1981), p. 375.

[28] J. A. Meijerink and H. A. van der Vorst, *An iterative solution method for linear systems of which the the coefficient matrix is a symmetric M–matrix*, Math. Comp., 31 (1977), pp. 148–162.

[29] J. E. Morel, M. L. Hall, and M. J. Shashkov, *A local support-operators diffusion discretization scheme for hexahedral meshes*, Submitted to the Journal of Computational Physics, (Summer 1999).

[30] C. C. Paige, B. N. Parlett, and H. A. van der Vorst, *Approximate solutions and eigenvalue bounds from Krylov subspaces*, Numerical linear algebra with applications, 2 (1995), pp. 115–133.

[31] C. C. Paige and M. A. Saunders, *LSQR : An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software, 8 (1982), pp. 43–71.

[32] Z. Q. Peng and A. G. Tijhuis, *Transient scattering by a lossy dielectric cylinder: Marching-on-in-frequency approach*, Journal of Electromagnetic Waves and Applications, 7 (1993), pp. 739–763.

[33] R. Rietman, *A common-mode skeleton model for EMC simulations.* To be published in SCEE-2000 proceedings.

[34] J. W. Ruge and K. Stüben, *Algebraic multigrid*, in Multigrid Methods, S. F. McCormick, ed., vol. 3 of SIAM Frontiers on Applied Mathematics, SIAM, 1987, ch. 4, pp. 73–130.

[35] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press series in algorithms and architectures for advanced scientific computing, Manchester University Press, 1992.

[36] Y. Saad and M. H. Schultz, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.

[37] G. L. G. Sleijpen. A Matlab implementation of the JDQR algorithm, available at `http://www.math.uu.nl/people/sleijpen/`.

[38] G. L. G. Sleijpen and H. A. Van der Vorst, *A Jacobi-Davidson iteration method for linear eigenvalue problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 401–425.

[39] K. Stüben, *Algebraic multigrid (AMG): An introduction*, GMD Report 53, GMD German National Research Center for Information Technology, 1999.

[40] The MathWorks, Inc. The commercial software package Matlab, information available at http://www.mathworks.com/products/matlab/.

[41] H. A. van der Vorst, *Computational Methods for Large Eigenvalue Problems*, Submitted for publication with Elsevier - North Holland.

[42] H. A. van der Vorst and C. Vuik, *GMRESR: a family of nested GMRES methods*, Numerical Linear Algebra with Applications, 1 (1994), pp. 369–386.

[43] C. Vuik, *Fast iterative solvers for the discretized incompressible navier-stokes equations*, Int. J. Num. Methods Fluids, 22 (1996), pp. 195–210.

[44] J. W. M. Coughran, M. R. Pinto, and R. K. Smith, *Continuation methods in semiconductor device simulation*, Journal of Computational and Applied Mathematics, 26 (1989), pp. 47–65.

[45] P. Wesseling, *An Introduction to Multigrid Methods*, John Wiley & Sons, Chichester, 1992.

Samenvatting

In deze samenvatting zal ik proberen om, aan de hand van de titel, uit te leggen waar dit proefschrift over gaat. Als eerste is in de titel te vinden dat het onderzoek te maken heeft met *methodes* voor het *oplos*sen van *elektromagnetische vergelijkingen*. Ik zal eerst uitleggen wat dit voor problemen zijn, en het vervolgens hebben over de methodes die we voorstellen om deze op te lossen.

Elektrische ladingen en stromen (bewegende elektrische ladingen) wekken een elektromagnetisch (EM) veld op. Onder bepaalde omstandigheden wordt dit ook wel elektromagnetische straling genoemd. Dit EM veld kan op zijn beurt weer ladingen en stromen genereren in een ander elektrisch geleidend voorwerp. Dit wordt ook wel inductie genoemd. Dit effect is al heel lang bekend en is de basis voor zeer veel toepassingen, zoals de radio en de inductiekookplaat.

Dit verschijnsel heeft ook negatieve effecten. Zo kunnen elektrische apparaten die elektromagnetische straling produceren, onbedoeld stromen opwekken in andere apparaten. Dit kan er voor zorgen dat dit andere apparaat slechter of helemaal niet meer werkt. Als voorbeeld zou men kunnen denken aan een videorecorder die vlak bij de televisie staat. Als de videorecorder een te sterk elektromagnetisch veld produceert kan dit onbedoelde stroompjes opwekken in de televisie, met het gevolg dat het televisiebeeld gestoord wordt. Serieuzere probleem zouden kunnen ontstaan in een ziekenhuis, waar b.v. in de operatiekamer veel vitale machines dicht bij elkaar staan. De hart-longmachine mag er niet voor zorgen dat de hartmonitor niet meer goed werkt.

De elektromagnetische interactie tussen elektrische apparaten en hun omgeving is onderdeel van elektromagnetische compatibiliteit (EMC). Een apparaat wordt elektromagnetisch compatibel genoemd als het in zijn omgeving goed kan functioneren en bovendien het goed functioneren van andere apparaten in zijn omgeving niet stoort. De Internationale Elektrotechnische Commissie (IEC) publiceert EMC standaards, waarin deze definitie van elektromagnetische compatibiliteit wordt vastgelegd in meetbare eisen voor een apparaat. Deze standaards worden onder andere gebruikt voor nationale en ook Europese wetgeving. Dit betekent dat alle elektrische apparatuur moet voldoen aan bepaalde EMC standaards.

Een belangrijk onderdeel van deze standaards is, dat het bij normaal gebruik door het apparaat opgewekte elektrisch veld gebonden is aan bepaalde maxima. Voor moderne elektrische apparatuur, waarin steeds meer en snellere digitale componenten gebruikt worden, wordt het steeds moeilijker om aan deze EMC standaards te voldoen. Het is daarom belangrijk om bij het ontwerpen van nieuwe elektronische apparatuur rekening te houden met deze EMC standaards. Als een eerste ontwerp gereed is moet er een prototype gemaakt worden zodat gemeten kan worden of het ontwerp voldoet aan de

benodigde EMC standaards. Als dit niet het geval is moet het ontwerp aangepast worden waarna er weer een prototype gemaakt moet worden, enzovoort. Het maken van een prototype en het testen hiervan is een tijdrovend proces. Het ontwerpproces kan aanzienlijk versneld worden als de ontwerper door middel van computer simulaties kan controleren of een ontwerp voldoet. In het ideale geval zou de ontwerper met een druk op de knop een nauwkeurige voorspelling van het gegenereerde elektrische veld krijgen. Echter, om met behulp van computer simulaties een hoge precisie te krijgen, moeten er heel grote problemen opgelost worden, en dat kost veel tijd. Voor complexe ontwerpen leidt dit er toe dat veel te lang gerekend moet worden om een redelijke voorspelling te maken.

Dit onderzoek heeft tot doel gehad om bij te dragen aan de ontwikkeling van methodes om deze computer simulaties zo snel te maken dat ze voor complexe ontwerpen toch praktisch toepasbaar zijn.

Bij het uitvoeren van dit soort simulaties wordt er als eerste een vereenvoudigd computermodel opgesteld van het ontwerp. Dit houdt o.a. in dat alle niet geleidende onderdelen (zoals alle plastic delen) weggelaten worden. Het is natuurlijk de bedoeling dat de vereenvoudigingen het probleem makkelijker maken maar de uitkomst niet of slechts zeer weinig beïnvloeden. In onze situatie bestaat het vereenvoudigde model uit een aantal vlakke rechthoekige dunne geleidende platen en dunne geleidende draden. Op de punten waar de draden met de platen verbonden zijn, zijn in het model spanningsbronnetjes opgenomen die de elektrische componenten in het ontwerp vervangen.

Voor het berekenen van het elektromagnetisch gedrag hebben we gebruik gemaakt van de *randintegraal* methode. Hierbij wordt door middel van integralen over het oppervlak van de geleider (de rand) het elektrisch veld uitgedrukt als functie van de ladingen en stromen in de geleider. Deze ladingen en stromen worden berekend met behulp van een randintegraalvergelijking over het oppervlak van de geleider.

Het meest tijdrovende gedeelte van deze aanpak is het berekenen van de ladingen en stromen door het oplossen van de randintegraalvergelijking. Hiervoor wordt eerst het oppervlak van de geleider gediscretiseerd, wat betekent dat de geleidende platen in kleine rechthoekjes worden verdeeld en de draden in kleine stukjes worden opgedeeld. Hierdoor kan de integraalvergelijking geschreven worden als een (zeer) groot stelsel “gewone” lineaire vergelijkingen. Dit stelsel is een vol stelsel, wat betekent dat alle onbekenden in alle vergelijkingen voorkomen.

Op de middelbare school leert men al hoe dit soort stelsels van lineaire vergelijkingen opgelost kunnen worden. Deze methodes zijn alleen niet meer erg praktisch voor stelsels van meer dan duizend vergelijkingen en onbekenden. We lossen deze stelsels vergelijkingen op met *iteratieve oplosmethodes*. Dit zijn methodes die uitgaan van een eerste schatting voor de oplossing en deze stap voor stap iets corrigeren zodat hij beter wordt. Door de correctiestap te herhalen (itereren) hopen we dat de benaderende oplossing steeds beter wordt en convergeert naar de echte oplossing. De iteratie wordt herhaald totdat de benaderende oplossing goed genoeg is. Deze methodes werken doorgaans het best voor stelsels vergelijkingen met bepaalde eigenschappen. In hoofdstuk 3 laten we zien dat het stelsel lineaire vergelijkingen dat wij moeten oplossen deze eigenschappen niet heeft. Door naar de originele integraalvergelijking te kijken constateren we dat het

probleem ligt in het verschillende gedrag van twee delen van de vergelijking, het capacitieve en het inductieve effect. In dit hoofdstuk wordt vervolgens een transformatie geïntroduceerd die deze twee effecten splitst zodat ze apart te behandelen zijn.

Deze splitsing maakt de weg vrij voor het verbeteren van de correctie stap in de iteratieve oplosmethode, zodat deze sneller kan convergeren. Deze techniek heet *preconditionering* en gebruikt doorgaans een alternatieve methode om het lineaire stelsel snel benaderend op te lossen. Hiervoor bestaan een aantal technieken, maar voor veel klassen van problemen voldoen deze niet zonder aanpassing. In dat geval moet, voor een goede convergentie, een van deze methodes aangepast worden of zelfs iets geheel nieuws bedacht worden.

In hoofdstuk 4 beschrijven we de toepassing van geometrisch multigrid als preconditionering voor ons stelsel lineaire vergelijkingen. Deze methode maakt gebruik van een serie van steeds grover wordende discretisaties om met relatief weinig rekentijd een benaderende oplossing van het probleem te maken. Gebruik hiervan, samen met de transformatie uit Hoofdstuk 3, zorgt voor een redelijk snelle convergentie van de iteratieve oplosmethode. Een nadeel van geometrisch multigrid is dat het maken van de grovere discretisaties lastig kan zijn, vooral voor een model met een ingewikkelde geometrie. In hoofdstuk 6 hebben we daarom ook een variant hierop, genaamd algebraïsch multigrid, bekeken. De met algebraïsch multigrid behaalde resultaten zijn echter niet zo goed als die voor geometrisch multigrid.

Voor het bepalen van het elektromagnetische gedrag van het model, moeten voor een groot aantal verschillende frequenties een aantal van dit soort stelsels van lineaire vergelijkingen opgelost worden. In hoofdstuk 5 maken we gebruik van het feit dat deze stelsels van lineaire vergelijkingen sterk op elkaar lijken en laten we zien hoe hiermee de benodigde rekentijd verder gereduceerd kan worden.

Dankwoord

Allereerst wil ik de mensen bedanken die direct betrokken waren bij dit project. Henk van der Vorst bedank ik voor zijn begeleiding, zijn vele ideeën, de prettige samenwerking en de vrijheid die hij mij gegeven heeft. Ook voor zijn uitvoerige commentaar op het manuscript en zijn pogingen om mijn schrijfstijl te verbeteren ben ik hem dankbaar. Aan de Philips kant van het project wil ik graag Wil Schilders, Jos Bergervoet en Ronald Rietman bedanken voor onze leerzame en vruchtbare discussies. Ik wil met name de fysici bedanken voor het delen van hun inzicht in de praktische kant van de problemen.

I would like to thank Jane Cullum for inviting me for a three month stay at the Los Alamos National Laboratory. I enjoyed our collaboration. There are a lot of people I have to thank for the good time I had in Los Alamos outside the lab. Among them are the house mates I had during that time, the LAFC fencers and my pool friends.

Graag wil ik ook alle collega's bedanken voor de goede sfeer op het instituut. De numerieke AIO's wil ik bedanken voor het gewillig luisteren naar verhalen over draadjes en lusjes. In het bijzonder wil ik mijn HPCN lotgenoot Wim bedanken voor zijn interesse in de EMC kant van het project en Ellen voor veel gezelligheid en een geslaagd eerste bezoek aan de VS. Voor de grote hoeveelheid dagelijkse gezelligheid tijdens en na het werk dank ik mijn collega's van "de zevende" en de Bastaardgangers. Met name bedank ik kamergenoten Luis en Mischja en *next nearest neighbours* Lennaert en Theo voor het overschot aan gezelligheid, een luisterend oor en lekker eten.

Special thanks to Elisabeth, whom I thank for her support by means of many long e-mails and for livening our stay in Nijmegen and Copper Mountain.

Verder wil ik de schermers bij Pallós bedanken voor een ruime hoeveelheid sportiviteit en gezelligheid tijdens en na het schermen, in het bijzonder Nick, Gert en Eric voor verscheidene bierproef- en spelavonden.

Ingrid wil ik graag bedanken voor haar steun, gezelschap en natuurlijk de vele lekkere maaltijden.

Ook mijn ouders en zus ben ik veel dank verschuldigd. Zonder hen was het nooit zo ver gekomen.

Curriculum Vitae

Menno Ewout Verbeek werd op 13 maart 1972 in Vlaardingen geboren. In 1990 behaalde hij zijn VWO diploma aan de R.S.G. F.A. Minkema te Woerden. Aansluitend begon hij met de studie natuurkunde aan de Universiteit Utrecht. In 1991 haalde hij het propedeutisch examen en in 1996 het doctoraal examen met hoofdvak theoretische natuurkunde (beide met lof). Aansluitend begon hij als assistent in opleiding zijn promotieonderzoek in de numerieke wiskunde aan de Universiteit Utrecht. De resultaten van dit onderzoek zijn beschreven in dit proefschrift. In de zomer van 1999 onderbrak hij dit onderzoek voor drie maanden om als Graduate-student Research Assistant te werken bij het Los Alamos National Laboratory in de VS.