

Improvement of a multigrid solver for 3D EM diffusion

T.B. Jönsthövel
Balijelaan 54 Bis, 3521 GV Utrecht

June 16, 2006

Acknowledgements

I am deeply indebted to my supervisor, W.A. Mulder, of Shell International Exploration and Production for giving me the opportunity to perform this research at Shell in Rijswijk and his continual support throughout the period of research and in the preparation of this thesis.

I would like to thank my tutor, C.W. Oosterlee, of the Delft University of Technology, faculty EWI, for his comments, advice and being a positive driving force to that period of research.

Contents

1	Introduction	8
1.1	Application EM diffusion	8
1.1.1	Introduction EM methods	8
1.1.2	Basics EM diffusion	8
1.2	Maxwell equations	9
1.3	Discretization Maxwell equations	11
2	Multigrid	19
2.1	Basics Multigrid (two-grid)	19
2.1.1	Multigrid cycle	20
2.2	Multigrid Components Mulder	22
2.2.1	Coarse grid specification	22
2.2.2	Smoother	23
2.2.3	Restriction Operator	26
2.2.4	Prolongation	27
2.3	Numerical experiments	28
2.3.1	Variable conductivity	29
2.3.2	Problems encountered in test cases	30
2.4	Anisotropy	32
2.4.1	Definition	32
2.4.2	Anisotropy on stretched grid	33
2.4.3	Anisotropy in 3D	34
2.5	Multigrid solutions for anisotropy	35
2.5.1	Cell-block smoothing and semicoarsening	36
2.5.2	Line smoothing and semicoarsening	37
2.5.3	Standard and semicoarsening with plane relaxation	38
3	Improving solver	39
3.1	Modifications multigrid for Maxwell equations	39
3.1.1	Semicoarsening	39
3.1.2	Line smoother	40
3.1.3	Multigrid scheme	45
3.2	Local mode analysis	47
3.2.1	Basics LMA	48
3.2.2	LMA of cell-block smoother in 3D	50
3.2.3	LMA of line smoother in 3D	54
3.2.4	Results LMA	56
4	Numerical experiments	58
4.1	Variable conductivity	59
4.2	Variable permittivity	60
4.3	Marine example	62
5	Conclusion	64

A Cell-block smoother	66
B Line smoother	68
B.1 A_{int}	68
B.2 A_{Γ_1}	71
B.3 A_{Γ_2}	72
B.4 A_{Γ_3}	72
C Local mode analysis	73
C.1 Cell-Block Smoother	73
C.2 Line smoother	79
D BiCGStab2	85
E Power-law Grid stretching	86
F L2 norm	86
G Matlab	88
G.1 Implementation Matlab code	88
G.2 Recursive call Matlab multigrid solver: e3db_msoll	89

List of Figures

1	Schematic representation of EM method for marine applications. . . .	9
2	The figure in the left pane shows a grid cell with grid nodes and edge-averaged components of the electric field. The figure in the right pane shows the face-averaged magnetic field components that are obtained by taking the curl of the electric field	12
3	The first electric field component $E_{1,k,l,m}$ is located at the intersection of the four cells shown in the left figure. Four faces of its dual volume are sketched in the right figure. The first component of the curl of the magnetic field should coincide with the edge on which E_1 lives. In the left figure, the four vectors that contribute to this curl are shown. They are defined as normals to the four faces at the top. Before computing their curl, these vectors are projected onto the faces of the dual volume shown at the right figure, where they now become tangential vectors. The curl is evaluated by taking the path integral of the dual volume that is obtained for constant x and by averaging over the interval $[x_k, x_{k+1}]$	12
4	Stokes's theorem applied to the electric field components lying at the face of a grid cell. The outward normal of the cell represents the magnetic field component.	14
5	Placement of dual volumes in 3D.	16
6	Placement of dual volumes as seen from the side.	17
7	Structure of an F-Cycle (\bullet , smoothing; \circ , exact solution; \setminus , fine-to-coarse; $/$, coarse-to-fine transfer).	22
8	Defining coarser grids.	23
9	The six point molecule. Each grid node is surrounded by six electric field components at the middle of the cell edges.	25
10	All 24 unknowns needed for updating the residuals lying on the edges.	25
11	Schematic representation of the full weighting restriction operator based on dual volumes. 26	
12	2D prolongation	27
13	Linear interpolation	28
14	Example of grid stretching. The source is represented by the yellow ellips lying in the middle of the figure.	30
15	Example of stretched grid	34
16	Semicoarsening strategies	36
17	Restriction of fine grid components to coarser grids with semi coarsening.	40
18	Schematic representation of line smoother. All unknowns needed for one line relaxation are drawn.	41
19	Schematic representation of the line smoother. The 5-point molecules are coupled in order to obtain one line relaxation.	42

20	Sparsity structure of the line smoother. The blue dots represents non-zero entries of the line smoother matrix.	43
21	F-cycle with block post-smoothing and standard coarsening.	46
22	F-cycle with line pre- and post-smoothing and standard coarsening.	46
23	F-cycle with line pre- and post-smoothing and semi coarsening.	47
24	Values errors electric field components in fourier modes before applying GS cell-block smoothing step n	50
25	Values errors electric field components in fourier modes after applying GS cell-block smoothing step n	51
26	Values errors electric field components in fourier modes before applying GS line smoothing step n	55
27	Values errors electric field components in fourier modes after applying GS line smoothing step n	55
28	Model for a marine example with a resistive salt body ($h_{\min} = 50\text{m}$). The sea bottom has a depth of about 600m.	62
29	Example of recursive call of F-cycle for Matlab Multigrid solver, <code>e3db_mso11</code> . On level 0, $N_x = 17$ and level 3, $N_x = 3$	89

Abstract

W.A. Mulder of Shell International Exploration and Production used multigrid to solve the time-harmonic Maxwell equations. These equations result from 3D electromagnetic diffusion which is used in the exploration and investigation of possible oil/gas reservoirs in the sea surface. The Maxwell equations and Ohm's law were discretized on tensor-product Cartesian grids with the finite integration technique.

For small scale test problems the multigrid solver with a Symmetric cell-block Gauss-Seidel smoother showed no performance leak. But as the number of cells in each direction increased and grid stretching was applied, the discretization of the Maxwell equations became anisotropic in all three directions. The number of iterations increased dramatically and multigrid convergence rates deteriorated.

Two well-known remedies for solving anisotropy were introduced, semicoarsening and line smoothing resulting in two new alternative multigrid solvers. Three test problems were evaluated and the multigrid schemes were used both as a stand-alone solver and as a preconditioner for `bicgstab2`.

Preface

This thesis is broken up into four different parts.

- The first part, Introduction, will give a detailed description of the basics of electromagnetic diffusion, the governing Maxwell equations, the discretization of these equations, the multigrid solver and a first test problem.
- The second part, Multigrid. In this part the difficulties encountered by the multigrid solver of the first part are highlighted. Anisotropy due to grid stretching and the possible remedies semicoarsening and line smoothing are introduced. Also a Local Mode Analysis is carried out to get better insight in the smoothing abilities of the line smoother.
- The third part, Improving solver, is dedicated to the adaptation of the multigrid solver introduced in the first part. All the different multigrid components are treated separately and in detail. At the end of the third part, three test problems are evaluated comparing the old and new multigrid solvers.
- The fourth part, Conclusion. In this part results and research are summarized and a short list is given with starting points for further investigation in order to improve the multigrid schemes.

1 Introduction

1.1 Application EM diffusion

1.1.1 Introduction EM methods

Electromagnetic methods (EM) have played a minor role in hydrocarbon exploration (oil/gas) while highly effective in mineral and environmental geophysics [18]. Because EM methods use signals that diffuse in the ground, they cannot provide the same vertical resolution as modern seismic exploration. Due to improvements in magnetotelluric (MT) data collection, application of MT used in settings where other exploration methods fail and MT being a complementary information source to seismic exploration, EM methods become more widely used in hydrocarbon exploration.

Just as in seismic exploration, EM methods can contribute to effective hydrocarbon exploration in two distinct ways:

1. Imaging structures that could host potential reservoirs and/or source rocks
2. Providing evidence for direct indication of the presence of hydrocarbons (e.g. the Troll field, Amundsen, Johansen & Røsten (2004): A Sea Bed Logging (SBL) calibration survey over the Troll gas field)

Table 1 summarizes the most common EM methods used in oil and gas exploration.

Method	Source	Signal type (freq or time domain)	Measured fields (electric or magnetic)	Depth of investigation in a sedimentary basin	Land or Marine
MT	Natural	Frequency	E and H	1 - 10 km	Both
AMT (audio MT)	Natural	Frequency	E and H	100 - 1000 m	Land
CSAMT (controlled source audio MT)	Grounded Dipole	Frequency	E and H	100 - 2000 m	Both

Table 1: EM methods used in oil and gas exploration

Next a review of an arbitrary EM method is presented which can be used in seabottom exploration.

1.1.2 Basics EM diffusion

Consider the following situation. Suppose there is an indication of a reservoir containing an unknown liquid or gas in the sea bottom, as shown in figure 1.

As in figure 1 a large number of receivers is placed at that bottom of the sea. Next, a ship with a large cable, which acts as an EM source, sails above the receivers.

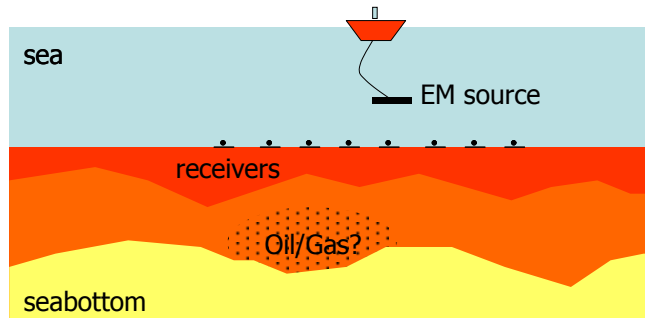


Figure 1: Schematic representation of EM method for marine applications.

The EM waves coming from the source, with frequencies in the $10^3 - 10^{-4}$ Hz band, travel through the water and diffuse into the earth and attenuate rapidly with depth. The penetration depth is called the skin depth. The surface measurement of electric and magnetic fields at the receivers gives the average resistivity from the surface to a depth equivalent of the skin depth. The skin depth increases as frequency decreases, and therefore a resistivity profile of the seabottom can be achieved by recording a range of frequencies. However, the seawater in deep oceans, a major conductor, screens out high frequency signals (above 10^{-2} Hz) needed to image structure in the upper few kilometers of the seabottom. But with modern recording equipment in low noise environments, higher frequency signals can be detected in moderate water depths.

1.2 Maxwell equations

Changes in the electromagnetic field described in the previous section are governed by the Maxwell equations. Later on it will become clear that these equations are the basis for the EM diffusion method.

The Maxwell equations (ME) are the set of four fundamental equations governing electromagnetism, i.e. the behavior of electric and magnetic fields. For time-varying fields, the differential form of these equations is [23],

$$\begin{aligned}
 \nabla \times \mathbf{E} + \partial_t \mathbf{B} &= 0, & (\text{Faraday's law}) \\
 \nabla \times \mathbf{H} - \partial_t \mathbf{D} &= \mathbf{J}, & (\text{Maxwell-Ampère law}) \\
 \nabla \cdot \mathbf{D} &= \rho_e, \\
 \nabla \cdot \mathbf{B} &= 0.
 \end{aligned} \tag{1}$$

The quantities above are functions of space \mathbf{x} and time t . The vector fields are \mathbf{E} (the electric field), \mathbf{H} (the magnetic field), \mathbf{D} (the electric displacement) and \mathbf{B} (the magnetic induction). The scalars are \mathbf{J} (the electric current density) and ρ_e (the electric charge density).

The equations from (1) form an underdetermined system [4]. A determinate system requires further assumptions. Hence, impose constitutive relations be-

tween the field quantities in order to make the system (1) definite. These take the form,

$$\begin{aligned}\mathbf{D} &= \epsilon \mathbf{E}, \\ \mathbf{B} &= \mu \mathbf{H}, \\ \mathbf{J} &= \sigma \mathbf{E} + \mathbf{J}_s.\end{aligned}$$

Here ϵ is the electric permittivity, μ is the magnetic permeability and σ is the conductivity of the media in which the electromagnetic field exists. Notice that $\sigma \mathbf{E}$ is in fact Ohms law and \mathbf{J}_s is the current density due to an externally applied electric source (e.g., the cable under the ship (1.1.2)).

In vacuum, ϵ and μ are isotropic and homogeneous. The corresponding permittivity of the vacuum is constant and denoted ϵ_0 while the permeability of the vacuum is also constant and denoted μ_0 . Vacuum is nonconductive so $\sigma = 0$. However, for more realistic problems, like geophysical data inversion, the material properties are usually not homogeneous and have discontinuities across the material boundaries (e.g., air and sea, sea and seabottom).

Choosing E and H as the unknown fields, Maxwell equations now become,

$$\begin{aligned}\nabla \times \mathbf{E} + \mu \partial_t \mathbf{H} &= 0, \\ \nabla \times \mathbf{H} - \sigma \mathbf{E} - \epsilon \partial_t \mathbf{E} &= \mathbf{J}_s, \\ \nabla \cdot (\epsilon \mathbf{E}) &= \rho_e, \\ \nabla \cdot (\mu \mathbf{H}) &= 0.\end{aligned}\tag{2}$$

In this particular case, ϵ and μ are assumed to be constant and can be written as: $\epsilon = \epsilon_r \epsilon_0$ and $\mu = \mu_r \mu_0$. Where, ϵ_r is the relative permittivity and ϵ_0 the vacuum value. Similarly, μ_r is the relative permeability and μ_0 the vacuum value.

The magnetic field can be eliminated from (2),

$$\begin{aligned}\nabla \times \mathbf{E} + \mu \partial_t \mathbf{H} = 0 &\iff -\mu^{-1} \nabla \times \mathbf{E} = \partial_t \mathbf{H} \\ \nabla \times \mathbf{H} - \sigma \mathbf{E} - \epsilon \partial_t \mathbf{E} = \mathbf{J}_s &\iff \nabla \times \partial_t \mathbf{H} - \sigma \partial_t \mathbf{E} - \epsilon \partial_{tt} \mathbf{E} = \partial_t \mathbf{J}_s \\ &\downarrow \\ \nabla \times \mu^{-1} \nabla \times \mathbf{E} + \sigma \partial_t \mathbf{E} + \epsilon \partial_{tt} \mathbf{E} &= -\partial_t \mathbf{J}_s\end{aligned}$$

As described in (1.1.2) most electromagnetic field computations are in the frequency domain. Also to avoid the use of an implicit time-stepping scheme, a transformation from the time to the frequency domain is introduced. Consider the following Fourier transformation [23]:

$$\mathbf{E}(\mathbf{x}, t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{\mathbf{E}}(\mathbf{x}, \omega) e^{-i\omega t} d\omega$$

Suppose $F_{x,t}(f(x, t))(\omega)$ is the Fourier transform of $f(x, t)$ then for the n^{th} derivative this yields:

$$\mathcal{F}_{\mathbf{x},t} \left[f^{(n)}(\mathbf{x}, t) \right] (\omega) = (i\omega)^n \mathcal{F}_{\mathbf{x},t} [f(\mathbf{x}, t)] (\omega)$$

Therefore the following equation is obtained

$$i\omega\mu_0\tilde{\sigma}\hat{\mathbf{E}} - \nabla \times \mu_r^{-1}\nabla \times \hat{\mathbf{E}} = -i\omega\mu_0\mathbf{J}_s, \quad (3)$$

where $\tilde{\sigma}(\mathbf{x}) = \sigma - i\omega\epsilon$ is the complex conductivity. Usually, $|\omega\epsilon| \ll \sigma$. From now, $\hat{\mathbf{E}}$ will be written as \mathbf{E} for convenience. On the boundaries of all domains introduced in the following sections, perfectly electrically conducting (PEC) boundary conditions will be used.

$$\mathbf{n} \times \mathbf{E} = 0 \text{ and } \mathbf{n} \cdot \mathbf{H} = 0$$

Here, \mathbf{n} is the outward normal on the boundary of the domain.

1.3 Discretization Maxwell equations

Equation (3) can be discretized by the finite integration technique (FIT) [5]. This scheme can be viewed as a finite-volume generalization of Yee's scheme [24] for tensor-product Cartesian grids with variable grid spacings.

Introduce a tensor-product grid with nodes at positions (x_k, y_l, z_m) for $k = 0 \dots N_x$, $l = 0 \dots N_y$ and $m = 0 \dots N_z$. Note that N_x , N_y and N_z are odd, integers and can be described as: $N_i = 2^{m_i} + 1$. The grid contains $N_x \times N_y \times N_z$ cells with these nodes as vertices. The cell centres are located at:

$$\begin{aligned} x_{k+\frac{1}{2}} &= \frac{1}{2}(x_k + x_{k+1}) \\ y_{l+\frac{1}{2}} &= \frac{1}{2}(y_l + y_{l+1}) \\ z_{m+\frac{1}{2}} &= \frac{1}{2}(z_m + z_{m+1}) \end{aligned}$$

Analogous to Yee's scheme, the electric field components are positioned at the edges of the cells and the magnetic field components are located at the middle of the faces of the cell (see figure 2).

Now, the component of the electric field $E_{1,k+\frac{1}{2},l,m}$ represents the average of $E_1(x, y_l, z_m)$ over the edge from x_k to x_{k+1} at given y_l and z_l . This can be written as:

$$E_1(x_{k+\frac{1}{2}}, y_l) = \frac{1}{x_{k+1} - x_k} \int_{x_k}^{x_{k+1}} E_1(x, y_l, z_m) dx$$

Other components of the electric field are defined in a similar way. The material properties, $\tilde{\sigma}$ and μ_r^{-1} are assumed to be given as cell-averaged values.

Next, each part of equation (3) will be discretized. First, dual volumes related to the edges are introduced. For a given edge, the dual volume is the

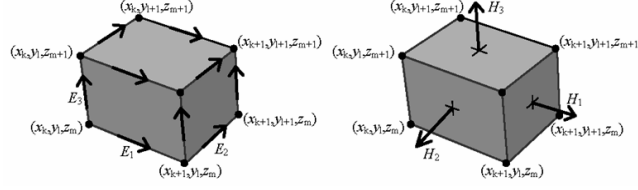


Figure 2: The figure in the left pane shows a grid cell with grid nodes and edge-averaged components of the electric field. The figure in the right pane shows the face-averaged magnetic field components that are obtained by taking the curl of the electric field

sum of the quarters of the total volume of the four neighboring cells (see figure 3).

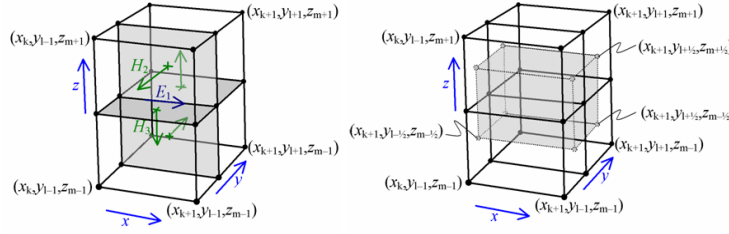


Figure 3: The first electric field component $E_{1,k,l,m}$ is located at the intersection of the four cells shown in the left figure. Four faces of its dual volume are sketched in the right figure. The first component of the curl of the magnetic field should coincide with the edge on which E_1 lives. In the left figure, the four vectors that contribute to this curl are shown. They are defined as normals to the four faces at the top. Before computing their curl, these vectors are projected onto the faces of the dual volume shown at the right figure, where they now become tangential vectors. The curl is evaluated by taking the path integral of the dual volume that is obtained for constant x and by averaging over the interval $[x_k, x_{k+1}]$.

The volume of a normal cell is defined by,

$$V_{k+\frac{1}{2}, l+\frac{1}{2}, m+\frac{1}{2}} = h_{k+\frac{1}{2}}^x h_{l+\frac{1}{2}}^y h_{m+\frac{1}{2}}^z,$$

with,

$$\begin{aligned} h_{k+\frac{1}{2}}^x &= x_{k+1} - x_k, \\ h_{l+\frac{1}{2}}^y &= y_{l+1} - y_l, \\ h_{m+\frac{1}{2}}^z &= z_{m+1} - z_m. \end{aligned}$$

The dual volume of the edge on which $E_{1,k+\frac{1}{2},l,m}$ lives, is given by,

$$V_{k+\frac{1}{2},l,m} = \frac{1}{4} h_{k+\frac{1}{2}}^x \sum_{m_2=0}^1 \sum_{m_3=0}^1 h_{l-\frac{1}{2}+m_2}^y h_{m-\frac{1}{2}+m_3}^z,$$

where,

$$\begin{aligned} d_k^x &= x_{k+\frac{1}{2}} - x_{k-\frac{1}{2}}, \\ d_l^y &= y_{l+\frac{1}{2}} - y_{l-\frac{1}{2}}, \\ d_m^z &= z_{m+\frac{1}{2}} - z_{m-\frac{1}{2}}. \end{aligned}$$

Therefore,

$$\begin{aligned} V_{k+\frac{1}{2},l,m} &= h_{k+\frac{1}{2}}^x d_l^y d_m^z, \\ V_{k,l+\frac{1}{2},m} &= h_{l+\frac{1}{2}}^y d_k^x d_m^z, \\ V_{k,l,m+\frac{1}{2}} &= h_{m+\frac{1}{2}}^z d_k^x d_l^y. \end{aligned}$$

Note that d_k^x , d_l^y and d_m^z are not defined at the boundaries yet. There are two options. First, take $d_0^x = h_{\frac{1}{2}}^x$ at $k = 0$ and $d_{N_x}^x = h_{N_x-\frac{1}{2}}^x$ at $k = N_x$, repeat for other directions. Or secondly, use Monk & Süli [10], $d_0^x = \frac{1}{2}h_{\frac{1}{2}}^x$ at $k = 0$ and $d_{N_x}^x = \frac{1}{2}h_{N_x-\frac{1}{2}}^x$ at $k = N_x$.

The discretization of (3):

1. $i\omega\mu_0\tilde{\sigma}\mathbf{E}$

The discrete form of this term multiplied by the corresponding dual volume (FIT) becomes $S_{k+\frac{1}{2},l,m}E_{1,k+\frac{1}{2},l,m}$, $S_{k,l+\frac{1}{2},m}E_{2,k+\frac{1}{2},l,m}$, $S_{k,l,m+\frac{1}{2}}E_{3,k,l,m+\frac{1}{2}}$ for the first, second and third component respectively. Here $S = i\omega\mu_0\tilde{\sigma}V$ is defined in terms of cell-averages.

e.g. The coefficient for $E_{1,k+\frac{1}{2},l,m}$ becomes:

$$\begin{aligned} S_{k+\frac{1}{2},l,m} &= \frac{1}{4} (S_{k+\frac{1}{2},l-\frac{1}{2},m-\frac{1}{2}} + S_{k+\frac{1}{2},l+\frac{1}{2},m-\frac{1}{2}} \\ &\quad + S_{k+\frac{1}{2},l-\frac{1}{2},m+\frac{1}{2}} + S_{k+\frac{1}{2},l+\frac{1}{2},m+\frac{1}{2}}) \end{aligned} \quad (4)$$

2. $\nabla \times \mu_r^{-1} \nabla \times \mathbf{E}$

The discretization of this term is not straightforward. First, the curl of the electric field components is discretized and placed at the middle of the faces of the grid cells. When divided by $i\omega\mu$, these are in fact the components of the magnetic field that are normal to the face of the cell. Next, the curl of the discretized components is again discretized in a similar way.

(a) $\mathbf{v} = \nabla \times \mathbf{E}$

The curl of \mathbf{E} can be discretized with Stokes's theorem (Adams, 2000):

$$\oint_C \mathbf{E} \cdot d\mathbf{r} = \int_V (\nabla \times \mathbf{E}) \cdot \mathbf{n} dV$$

Now consider the grid cell in figure 4. First, apply Stokes's Theorem to components E_1 and E_3 lying at the edge of the surface $[x_k, x_{k+1}] \times [z_m, z_{m+1}]$. In detail:

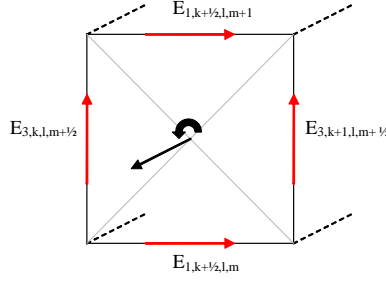


Figure 4: Stokes's theorem applied to the electric field components lying at the face of a grid cell. The outward normal of the cell represents the magnetic field component.

In continuous form with \mathbf{n} the outward normal of the grid cell:

$$\oint_C \mathbf{E} \cdot d\mathbf{r} = \oint_C E_1(x, y, z) dx + E_2(x, y, z) dy = \int_V (\nabla \times \mathbf{E}) \cdot \mathbf{n} dV.$$

In discrete form,

$$\begin{aligned} & \left(E_3(x_k, y_l, z_{m+\frac{1}{2}}) h_{m+\frac{1}{2}}^z + E_1(x_{k+\frac{1}{2}}, y_l, z_{m+1}) h_{k+\frac{1}{2}}^x \right) \\ & - \left(E_3(x_{k+1}, y_l, z_{m+\frac{1}{2}}) h_{m+\frac{1}{2}}^z + E_1(x_{k+\frac{1}{2}}, y_l, z_m) h_{k+\frac{1}{2}}^x \right) \\ & = V \cdot (\nabla \times \mathbf{E}) \begin{pmatrix} 0 & 1 & 0 \end{pmatrix}^T = V \cdot v_2(x_{k+\frac{1}{2}}, y_l, z_{m+\frac{1}{2}}) \\ & = h_{m+\frac{1}{2}}^z h_{k+\frac{1}{2}}^x \cdot v_2(x_{k+\frac{1}{2}}, y_l, z_{m+\frac{1}{2}}). \end{aligned}$$

Note that the path integral follows the direction of the curl operator and therefore two electric field components ($E_{1,k+\frac{1}{2},l,m}$ and $E_{3,k+1,l,m+\frac{1}{2}}$) point in the opposite direction.

Simplification yields,

$$v_{2,k+\frac{1}{2},l,m+\frac{1}{2}} = \frac{1}{h_{k+\frac{1}{2}}^x} \left(E_3(x_k, y_l, z_{m+\frac{1}{2}}) - E_3(x_{k+1}, y_l, z_{m+\frac{1}{2}}) \right) - \frac{1}{h_{m+\frac{1}{2}}^z} \left(E_1(x_{k+\frac{1}{2}}, y_l, z_{m+1}) - E_1(x_{k+\frac{1}{2}}, y_l, z_m) \right).$$

Repeat this procedure to obtain the two remaining discretized curl components v_1 and v_2 :

$$v_{1,k,l+\frac{1}{2},m+\frac{1}{2}} = \frac{1}{h_{l+\frac{1}{2}}^y} \left(E_3(x_k, y_{l+1}, z_{m+\frac{1}{2}}) - E_3(x_k, y_l, z_{m+\frac{1}{2}}) \right) - \frac{1}{h_{m+\frac{1}{2}}^z} \left(E_1(x_k, y_{l+\frac{1}{2}}, z_{m+1}) - E_1(x_k, y_{l+\frac{1}{2}}, z_m) \right)$$

$$v_{3,k+\frac{1}{2},l,m+\frac{1}{2}} = \frac{1}{h_{k+\frac{1}{2}}^x} \left(E_2(x_{k+1}, y_{l+\frac{1}{2}}, z_m) - E_2(x_k, y_{l+\frac{1}{2}}, z_m) \right) - \frac{1}{h_{l+\frac{1}{2}}^y} \left(E_1(x_{k+\frac{1}{2}}, y_{l+1}, z_m) - E_1(x_{k+\frac{1}{2}}, y_l, z_m) \right)$$

(b) $\mu_r^{-1} \nabla \times \mathbf{E}$

The scaling by μ_r^{-1} at the face requires another averaging procedure because the material properties are assumed to be given as cell-averaged values,

$$\iiint_V \mu_r^{-1} dV = V \mu_r^{-1}.$$

Hence, define $M = V \mu_r^{-1}$:

$$M_{k+\frac{1}{2},l+\frac{1}{2},m+\frac{1}{2}} = h_{k+\frac{1}{2}}^x h_{l+\frac{1}{2}}^y h_{m+\frac{1}{2}}^z \mu_{r,k+\frac{1}{2},l+\frac{1}{2},m+\frac{1}{2}}^{-1} \quad (5)$$

for a given cell $(k + \frac{1}{2}, l + \frac{1}{2}, m + \frac{1}{2})$. So, an averaging step in the z-direction provides:

$$M_{k+\frac{1}{2},l+\frac{1}{2},m} = \frac{1}{2} \left(M_{k+\frac{1}{2},l+\frac{1}{2},m-\frac{1}{2}} + M_{k+\frac{1}{2},l+\frac{1}{2},m+\frac{1}{2}} \right)$$

at the face $(k + \frac{1}{2}, l + \frac{1}{2}, m)$ between the cells $(k + \frac{1}{2}, l + \frac{1}{2}, m + \frac{1}{2})$ and $(k + \frac{1}{2}, l + \frac{1}{2}, m - \frac{1}{2})$.

In the previous step the curl of \mathbf{E} has been discretized. Now multiply with the discrete factor $\mu_r^{-1}V$:

$$\begin{aligned} u_{1,k,l+\frac{1}{2},m+\frac{1}{2}} &= M_{k,l+\frac{1}{2},m+\frac{1}{2}} v_{1,k,l+\frac{1}{2},m+\frac{1}{2}}, \\ u_{2,k+\frac{1}{2},l,m+\frac{1}{2}} &= M_{k+\frac{1}{2},l,m+\frac{1}{2}} v_{2,k+\frac{1}{2},l,m+\frac{1}{2}}, \\ u_{3,k+\frac{1}{2},l,m+\frac{1}{2}} &= M_{k+\frac{1}{2},l,m+\frac{1}{2}} v_{3,k+\frac{1}{2},l,m+\frac{1}{2}}. \end{aligned}$$

(c) $\nabla \times \mu_r^{-1} \nabla \times \mathbf{E}$

Finally, $\nabla \times \mathbf{u}$ needs to be discretized. Note that the components of \mathbf{u} are related to the magnetic field components by,

$$\begin{aligned} u_{1,k,l+\frac{1}{2},m+\frac{1}{2}} &= i\omega\mu_0 V_{k,l+\frac{1}{2},m+\frac{1}{2}} H_{1,k,l+\frac{1}{2},m+\frac{1}{2}}, \\ u_{2,k+\frac{1}{2},l,m+\frac{1}{2}} &= i\omega\mu_0 V_{k+\frac{1}{2},l,m+\frac{1}{2}} H_{2,k+\frac{1}{2},l,m+\frac{1}{2}}, \\ u_{3,k+\frac{1}{2},l,m+\frac{1}{2}} &= i\omega\mu_0 V_{k+\frac{1}{2},l,m+\frac{1}{2}} H_{3,k+\frac{1}{2},l,m+\frac{1}{2}}. \end{aligned}$$

where,

$$\begin{aligned} V_{k,l+\frac{1}{2},m+\frac{1}{2}} &= d_k^x h_{l+\frac{1}{2}}^y h_{m+\frac{1}{2}}^z, \\ V_{k+\frac{1}{2},l,m+\frac{1}{2}} &= h_{k+\frac{1}{2}}^x d_l^y h_{m+\frac{1}{2}}^z, \\ V_{k+\frac{1}{2},l,m+\frac{1}{2}} &= h_{k+\frac{1}{2}}^x h_{l+\frac{1}{2}}^y d_m^z. \end{aligned} \tag{6}$$

The components of \mathbf{u} lie on the edges of the dual volumes as introduced earlier as shown in figures 5 and 6.

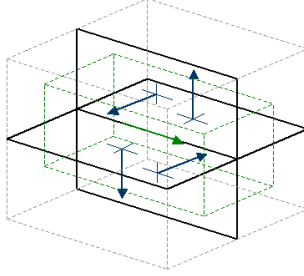


Figure 5: Placement of dual volumes in 3D.

The first component of the curl, w_1 , is evaluated by applying Stokes's Theorem again:

$$\oint_C \mathbf{E} \cdot d\mathbf{r} = \int_V (\nabla \times \mathbf{E}) \cdot \mathbf{n} dV$$

With C a rectangle of the dual volume that is obtained for constant x and V the dual volume.

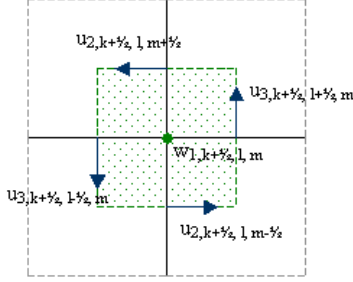


Figure 6: Placement of dual volumes as seen from the side.

2. $i\omega\mu_0\mathbf{J}_s$

(a) The discretization of the source term is straightforward,

$$\begin{aligned} s_{1,k+\frac{1}{2},l,m} &= i\omega\mu_0 V_{k+\frac{1}{2},l,m} J_{1,k+\frac{1}{2},l,m}, \\ s_{2,k,l,m+\frac{1}{2}} &= i\omega\mu_0 V_{k,l+\frac{1}{2},m} J_{2,k,l,m+\frac{1}{2}}, \\ s_{3,k,l,m+\frac{1}{2}} &= i\omega\mu_0 V_{k,l,m+\frac{1}{2}} J_{3,k,l,m+\frac{1}{2}}. \end{aligned}$$

Let the residual for an arbitrary electric field that is not necessarily a solution to the problem, be defined as

$$\mathbf{r} = V \left(i\omega\mu_0 \tilde{\sigma} \hat{\mathbf{E}} - \nabla \times \mu_r^{-1} \nabla \times \hat{\mathbf{E}} + i\omega\mu_0 \mathbf{J}_s \right). \quad (7)$$

In discrete form:

$$\begin{aligned} r_{1,k+\frac{1}{2},l,m} &= s_{1,k+\frac{1}{2},l,m} + S_{k+\frac{1}{2},l,m} E_{1,k+\frac{1}{2},l,m} \\ &\quad - \left[e_{l+\frac{1}{2}}^y u_{3,k+\frac{1}{2},l+\frac{1}{2},m} - e_{l-\frac{1}{2}}^y u_{3,k+\frac{1}{2},l-\frac{1}{2},m} \right] \\ &\quad + \left[e_{m+\frac{1}{2}}^z u_{2,k+\frac{1}{2},l,m+\frac{1}{2}} - e_{m-\frac{1}{2}}^z u_{2,k+\frac{1}{2},l,m-\frac{1}{2}} \right] \\ r_{2,k,l+\frac{1}{2},m} &= s_{2,k,l+\frac{1}{2},m} + S_{k,l+\frac{1}{2},m} E_{2,k,l+\frac{1}{2},m} \\ &\quad - \left[e_{m+\frac{1}{2}}^z u_{1,k,l+\frac{1}{2},m+\frac{1}{2}} - e_{m-\frac{1}{2}}^z u_{1,k,l+\frac{1}{2},m-\frac{1}{2}} \right] \\ &\quad + \left[e_{k+\frac{1}{2}}^x u_{3,k+\frac{1}{2},l+\frac{1}{2},m} - e_{k-\frac{1}{2}}^x u_{3,k-\frac{1}{2},l+\frac{1}{2},m} \right] \\ r_{3,k,l,m+\frac{1}{2}} &= s_{3,k,l,m+\frac{1}{2}} + S_{k,l,m+\frac{1}{2}} E_{3,k,l,m+\frac{1}{2}} \\ &\quad - \left[e_{k+\frac{1}{2}}^x u_{2,k+\frac{1}{2},l,m+\frac{1}{2}} - e_{k-\frac{1}{2}}^x u_{2,k-\frac{1}{2},l,m+\frac{1}{2}} \right] \\ &\quad + \left[e_{l+\frac{1}{2}}^y u_{1,k,l+\frac{1}{2},m+\frac{1}{2}} - e_{l-\frac{1}{2}}^y u_{1,k,l-\frac{1}{2},m+\frac{1}{2}} \right] \end{aligned} \quad (8)$$

where,

$$\begin{aligned} e_{k\pm\frac{1}{2}}^x &= \frac{1}{h_{k\pm\frac{1}{2}}^x}, \\ e_{l\pm\frac{1}{2}}^y &= \frac{1}{h_{l\pm\frac{1}{2}}^y}, \\ e_{m\pm\frac{1}{2}}^z &= \frac{1}{h_{m\pm\frac{1}{2}}^z}. \end{aligned} \tag{9}$$

It may appear that the weighting of the differences is in contradiction with Stokes's theorem as stated above in step 2c. However, the differences have been multiplied by the local dual volume.

The discretization has been completed. The next step is to find the solution \mathbf{E} of $\mathbf{r} = 0$ for a given domain, material parameters, source term and boundary conditions.

2 Multigrid

In this section a brief summary of the basic principles of multigrid is given. At the end of this section a description of the different multigrid components used by Mulder [12] can be found.

2.1 Basics Multigrid (two-grid)

Consider the following discretization of an arbitrary equation on a grid with spatial mesh size h , Ω_h :

$$L_h \mathbf{e}_h = \mathbf{s}_h \quad (10)$$

If the solution of this equation is approximated by e_h^m , the error and residual are as follows:

$$\begin{aligned} \delta \mathbf{e}_h^m & : = \mathbf{e}_h - \mathbf{e}_h^m \\ \mathbf{r}_h^m & : = \mathbf{s}_h - L_h \mathbf{e}_h^m \end{aligned}$$

This results in the defect equation which is equivalent with the original equation because $e_h = \delta \mathbf{e}_h^m + e_h^m$:

$$L_h \delta \mathbf{e}_h^m = \mathbf{r}_h^m$$

If a basic iterative method, like Jacobi or Gauss-Seidel, is used to solve the equation and the error is computed, then it appears that the error becomes smooth after several iteration steps. In that case, the iteration formula can be interpreted as an error averaging process. This error-smoothing is one of the two basic principles of the multigrid approach. The other principle is based on the fact that a quantity that is smooth on a certain grid can also be approximated on a coarser grid. So if the error of the approximation of the solution has become smooth after several relaxation sweeps, then this error can be approximated with a suitable procedure on a coarser grid.

Suppose that the matrix A_h can be approximated by a more easy to invert matrix \hat{L}_h then:

$$\hat{L}_h \delta \hat{\mathbf{e}}_h^m = \mathbf{r}_h^m \longrightarrow \mathbf{e}_h^{m+1} = \mathbf{e}_h^m + \delta \hat{\mathbf{e}}_h^m$$

The idea of multigrid is to approximately solve the defect equation on a coarser grid with spatial mesh size, e.g. $H := 2h$. Obviously, this will take less time and work than a conventional direct method on a grid with spatial mesh size h .

$$L_H \delta \hat{\mathbf{e}}_H^m = \mathbf{r}_H^m \quad (11)$$

Assume that L_H^{-1} exists. As \mathbf{r}_H^m and $\delta\hat{\mathbf{e}}_H^m$ are grid functions on the coarser grid, introduce two (linear) transfer operators:

$$I_h^H : \mathcal{G}(\Omega_h) \longrightarrow \mathcal{G}(\Omega_H), \quad I_H^h : \mathcal{G}(\Omega_H) \longrightarrow \mathcal{G}(\Omega_h)$$

These functions are necessary to restrict and prolongate the residuals and approximations of the error to different coarser and finer grids. This yields,

$$\begin{aligned} \mathbf{r}_H^m & : = I_h^H \mathbf{r}_h^m, \text{ restrict } \mathbf{r}_h^m \text{ to } \Omega_H \\ \delta\hat{\mathbf{e}}_h^m & : = I_H^h \delta\hat{\mathbf{e}}_H^m, \text{ prolongate } \delta\hat{\mathbf{e}}_H^m \text{ to } \Omega_h \end{aligned}$$

One choice for I_h^H can be the injection operator. For instance, the residual on a fine grid Ω_h will be mapped directly to the coarser grid Ω_H . No weighting has been applied. Other operators are based on (full) weighting (I_h^H) and linear or bilinear interpolation for I_H^h . In section (2.2) these operators will be described in detail.

Unfortunately coarse grid correction alone is not enough. In general, the interpolation of coarse grid corrections reintroduces high frequency error components on the fine grid [17]. One natural approach to reduce them is to introduce one or a few additional smoothing sweeps before and after the coarse grid correction. These sweeps are known as pre- and post-smoothing.

2.1.1 Multigrid cycle

The multigrid idea starts from the observation that in a well converged two-grid method (section 2.1) it is neither useful nor necessary to solve the coarse grid defect equation (11) exactly. Instead, without loss of convergence speed, one may replace $\delta\hat{\mathbf{e}}_H^m$ by a suitable approximation. A natural way to obtain such an approximation is to apply the two-grid idea to (11) again, now employing an ever coarser grid than Ω_H .

This is possible, as obviously the coarse grid equation (11) is of the same form as the original equation (10). If the convergence factor of the two-grid method is small enough, it is sufficient to perform only a few, say γ , two-grid iteration steps to obtain a good enough approximation to the solution of (11). This idea can, in a straightforward manner, be applied recursively, using coarser and coarser grids, down to some coarsest grid. On this coarsest grid any solution method may be used (e.g. a direct method or some relaxation-type method if it has sufficiently good convergence properties on that coarsest grid). In ideal cases, the coarsest grid consists of just one grid point.

For a formal description of multigrid methods use a sequence of coarser and coarser grids Ω_{h_k} , characterized by a sequence of mesh sizes h_k :

$$\Omega_{h_l}, \Omega_{h_{l-1}}, \dots, \Omega_{h_0}$$

The coarsest grid is characterized by the mesh size h_0 whereas the index l corresponds to the finest grid $\Omega_h : h = h_l$. For simplicity, replace the index h_k by k in the following. For each Ω_k , assume that linear operators

$$\begin{aligned}
L_k &: G(\Omega_k) \rightarrow G(\Omega_k), & S_k &: G(\Omega_k) \rightarrow G(\Omega_k), \\
I_k^{k-1} &: G(\Omega_k) \rightarrow G(\Omega_{k-1}), & I_{k-1}^k &: G(\Omega_{k-1}) \rightarrow G(\Omega_k)
\end{aligned} \tag{12}$$

are given, where the L_k are discretizations of L on Ω_k for $k = l, \dots, 0$, and where the original equation (10) reads

$$L_l \mathbf{e}_l = \mathbf{s}_l \quad (\Omega_l) \tag{13}$$

and is the discrete problem to solve. The operators S_k denote the linear iteration operator corresponding to given smoothing methods on Ω_k . Performing ν smoothing steps (applied to any discrete problem of the form $L_k \mathbf{e}_k = \mathbf{s}_k$ with initial approximation \mathbf{e}_k^m) resulting in the approximation $\bar{\mathbf{e}}_k^m$ will be denoted by

$$\bar{\mathbf{e}}_k^m = \text{SMOOTH}^\nu(\mathbf{e}_k^m, L_k, \mathbf{s}_k)$$

Now introduce multigrid cycle, more precisely an $(l+1)$ -grid cycle, to solve (13) for a fixed $l \geq 1$. Using the operators L_k ($k = l, l-1, \dots, 0$) as well as $S_k, I_k^{k-1}, I_{k-1}^k$ ($k = l, l-1, \dots, 1$), assuming parameters v_1, v_2 (the number of pre- and postsmoothing iterations) and γ to be fixed and starting on the finest grid $k = l$, the calculation of a new iterate \mathbf{e}_k^{m+1} from given approximation \mathbf{e}_k^m to the solution \mathbf{e}_k proceed as follows:

<p>Multigrid cycle $\mathbf{e}_k^{m+1} = \text{MGCYC}(k, \gamma, \mathbf{e}_k^m, L_k, \mathbf{s}_k, v_1, v_2)$</p> <p>(1) Presmoothing</p> <ul style="list-style-type: none"> - Compute $\bar{\mathbf{e}}_k^m$ by applying $v_1 (\geq 0)$ smoothing steps to \mathbf{e}_k^m $\bar{\mathbf{e}}_k^m = \text{SMOOTH}^{v_1}(\mathbf{e}_k^m, L_k, \mathbf{s}_k)$ <p>(2) Coarse grid correction</p> <ul style="list-style-type: none"> - Compute the defect $\bar{\mathbf{r}}_k^m := \mathbf{s}_k - L_k \bar{\mathbf{e}}_k^m$ - Restrict the defect $\bar{\mathbf{r}}_{k-1}^m := I_{k-1}^{k-1} \bar{\mathbf{r}}_k^m$ - Compute an approximate solution $\delta \mathbf{e}_{k-1}^m$ of the defect equation on Ω_{k-1} $L_{k-1} \delta \mathbf{e}_{k-1}^m = \bar{\mathbf{r}}_{k-1}^m \quad (*)$ <p>by</p> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p>If $k = 1$ use a direct or fast iterative solver for $(*)$</p> <p>If $k > 1$ solve $(*)$ approximately by performing $\gamma (\geq 1)$ k-grid cycles using the zero grid function as a first approximation</p> $\delta \mathbf{e}_{k-1}^m = \text{MGCYC}^\gamma(k-1, \gamma, 0, L_{k-1}, \bar{\mathbf{r}}_{k-1}^m, v_1, v_2)$ </div> <ul style="list-style-type: none"> - Interpolate the correction $\delta \mathbf{e}_k^m := I_{k-1}^k \delta \mathbf{e}_{k-1}^m$ - Compute the corrected approximation on Ω_k $\mathbf{e}_k^{m, \text{after CGC}} = \bar{\mathbf{e}}_k^m + \delta \mathbf{e}_k^m$ <p>(3) Postsmoothing</p> <ul style="list-style-type: none"> - Compute \mathbf{e}_k^{m+1} by applying $v_2 (\geq 0)$ smoothing steps to $\mathbf{e}_k^{m, \text{after CGC}}$ $\mathbf{e}_k^{m+1} = \text{SMOOTH}^{v_2}(\mathbf{e}_k^{m, \text{after CGC}}, L_k, \mathbf{s}_k)$

The different number of two-grid iterations steps determine the structure of a multigrid cycle. Possibilities are the V-cycle ($\gamma = 1$), W-cycle ($\gamma = 2$) or F-cycle ($\gamma = \gamma_k$). The main differences between these approaches are the number of pre- and post-smoothing steps and the different number of coarser grids used. However, through trial and error the F-cycle (see figure 7) has proven itself a relative low-cost and reliable multigrid cycle for this particular case (Maxwell equations). Therefore, the other cycles are not taken into account throughout the remainder of this thesis.

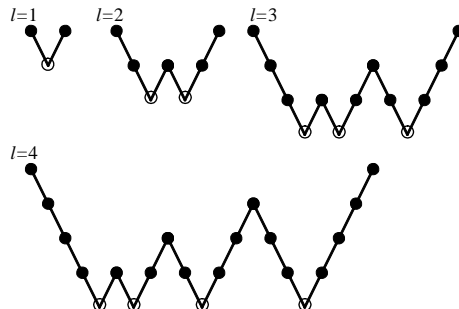


Figure 7: Structure of an F-Cycle (●, smoothing; ○, exact solution; \, fine-to-coarse; /, coarse-to-fine transfer).

2.2 Multigrid Components Mulder

As described above, the following multigrid components have to be defined:

- Coarse grid specification
- Smoother
- Restriction operator
- Prolongation operator

In [6] a multigrid method for a FIT discretization is presented. Mulder uses this approach as a starting point and modifies some components of the multigrid solver.

2.2.1 Coarse grid specification

The FIT discretization uses a tensor-product Cartesian grid. The coarse-grid cells are formed by combining $2 \times 2 \times 2$ fine-grid cells. This is a special case of the method in [6] where arbitrary coarser-grids are used with nodes that are not necessarily a subset of those on the fine grid. Obviously, in this case the coarse-grid nodes are a subset of the fine-grid nodes. For example, consider the highly simplified 2-D situation shown in figure 8.

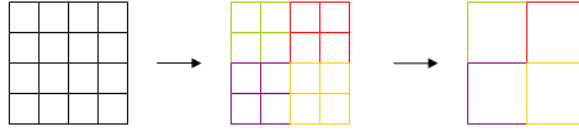


Figure 8: Defining coarser grids.

The fine grid with 4x4 cells maps to a coarser-grid with 2x2 cells. Four neighboring fine grid cells are put together in order to form one coarser-grid cell. This example can easily be extended to three dimensions and a more sophisticated (stretched) tensor-product Cartesian grid.

2.2.2 Smoother

The smoother has two vital tasks. First, it acts as a smoother. The high-frequency error components on the fine grids are smoothed down. Secondly, locally it computes a new approximation for the solution \mathbf{e}_H of $L_H \mathbf{e}_H = \mathbf{s}_H$, with H the coarsest spatial mesh size.

Multigrid methods are motivated by the fact that many iterative methods, especially if they are applied to elliptic problems, have a smoothing effect on the error between an exact solution and a numerical approximation. A smooth discrete error can be well represented on a coarser grid, where its approximation is much cheaper. The design of efficient relaxation methods for the multigrid solution of systems of partial differential equations often requires special attention. Due to the system of partial differential equations resulting from the discretization of the Maxwell equations a short discussion is given first about how the smoother should smooth the error for all unknowns in the equations (that are possibly of different type) of the system. Next, the smoother used here will be introduced and motivated.

Multigrid for systems of PDE's A good indication for the appropriate choice of relaxation method for a system of equations can be derived from the systems' determinant. If the main operators (or their principal parts) of the determinant lie on the main diagonal, smoothing is a straightforward matter. In that case, the differential operator that corresponds to the primary unknown in each equation is the leading operator. Therefore, a simple equation-wise decoupled relaxation method can efficiently be used. If, however, the main operators in a system are not in the desired position, the choice of an efficient smoother needs care. A first obvious choice in the case of strong off-diagonal operators in the differential system is coupled or collective smoothing: All unknowns in the system at a certain grid point are updated simultaneously.

Decoupled smoothing, however, is to be preferred for reasons of efficiency and simplicity. An elegant way to describe distributive relaxation is to introduce

a right preconditioner in the smoothing procedure [22]. This means that we introduce new variables \mathbf{w}_h , where $\bar{\mathbf{u}}_h = C_h \mathbf{w}_h$, and consider the transformed system $L_h C_h \mathbf{w}_h = \mathbf{f}_h$, with C_h chosen in such a way that the resulting operator $L_h C_h$ is suited for *decoupled* (non-collective) relaxation.

Coupled and decoupled smoothing approaches have their advantages and disadvantages. If a system of equations consists of elliptic and of other, non-elliptic, components, decoupled relaxation allows to choose different relaxation methods for the different operators appearing. However, for general systems of equations it may not be easy to find a suitable distributive relaxation scheme. Furthermore, the proper treatment of boundary conditions in distributive relaxation may not be trivial, as typically the systems' operator is transformed by the smoother but the boundary operator is sometimes not considered. In this respect the use of coupled smoother is straightforward and often robust. A significant difference in cost between coupled and distributive relaxation, however, lies in the line-wise treatment of the unknowns, which may be necessary in the case of stretched grids. The cost of a coupled line-wise relaxation step is substantially higher than of a decoupled line-wise relaxation. The latter can be set up as a tri- (or more) diagonal matrix, whereas in the coupled version all different unknowns at the line need to be updated simultaneously. For Maxwell's equations the two variants mentioned above are basically proposed by Arnold, Falk and Winther [3] (coupled relaxation) and Hiptmair [8] (decoupled smoothing). The smoothers are constructed such that the null-spaces from the curl-curl operator are handled adequately within smoothing.

Definition cell-block smoother As smoother, the coupled relaxation of Arnold, Falk and Winther [3] is used here. This smoother has the nice property that it automatically imposes the divergence-free character of $\tilde{\sigma} \mathbf{E}$ and does not require an explicit divergence correction as in [8].

The method selects one node and solves for the six degrees of freedom on the six edges attached to the node. The smoother is applied in a symmetric Gauss-Seidel fashion, following the lexicographical ordering of the nodes (x_k, y_l, z_m) . This implies that the index goes from (x_1, y_1, z_1) to (x_{Nx}, y_1, z_1) and then from (x_1, y_2, z_1) to (x_{Nx}, y_2, z_1) etc. Due to the PEC boundary conditions the system of equations of the nodes on the boundary do not have to be solved because the solution of the electric field components is already known there.

After each node is done, the electric field components are updated with the most recent solution. When the first smoothing step is finished the next one is carried out in opposite direction. So, the index starts with (x_{Nx}, y_{Ny}, z_{Nz}) and follows its way in a lexicographical order back to (x_1, y_1, z_1) .

When this smoother is applied to equation (3), a 6×6 system of equations has to be solved locally. Suppose a node at position (x_k, y_l, z_m) is selected as shown in figure 9.

This yields the following (local) system of equations

$$A \mathbf{E}_{k,l,m} = s - B \mathbf{E}_{sur} \quad (14)$$

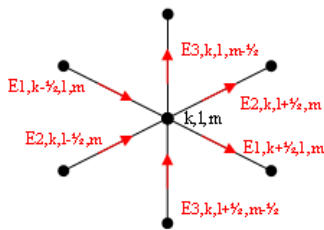


Figure 9: The six point molecule. Each grid node is surrounded by six electric field components at the middle of the cell edges.

Where, $E_{k,l,m}$ is a vector containing the six electric field components lying at the surrounding edges of the node. Matrix A contains all the coefficients belonging to six entries of the vector $E_{k,l,m}$. The vector s contains the six corresponding source terms (see equation 3) and the vector resulting from the matrix-vector multiplication BE_{sur} , contains all information from surrounding electric field components. The latter have to be taken into account because they are needed to compute the curl operator of equation (3). Consider figure 10.

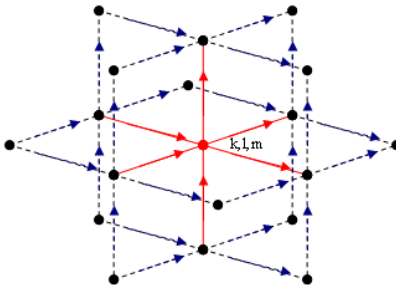


Figure 10: All 24 unknowns needed for updating the residuals lying on the edges.

The red arrows represent the six unknown electric field components at the edges and surrounding grid node (x_k, y_l, z_m) . The 24 blue arrows are the electric field components corresponding to neighboring grid nodes which are needed to compute the discretization of the curl operator at the same edges as the entries of $E_{k,l,m}$. All matrices and vectors of (14) are described in detail and can be found in appendix (A).

The question remains when this smoother should be applied. One full multigrid cycle contains two post-smoothing steps ($\nu_2 = 2$, see multigridcycle in section (2.1.1)) which is equivalent with one symmetric Gauss-Seidel iteration. Furthermore, pre-smoothing has not been applied ($\nu_1 = 0$) and other choices have not been taken into consideration here.

There is one remark to be made. In the case that $\tilde{\sigma} = 0$, which may occur

if the electric field in air is modelled and ϵ_r is set to zero, the local 6x6 systems become singular. This problem can be (artificially) avoided when solving the small local systems by replacing $\tilde{\sigma}$ with a small positive number.

2.2.3 Restriction Operator

In this section the restriction operator will be described.

Suppose, to keep matters simple, that $N_x = N_y = N_z = 2^m + 1$, with integer $m \geq 1$. The coarse grid is defined as above (2.2.1).

The discrete operator on the coarser grids is chosen to be the same as the one obtained by direct discretization. The cell-averaged material properties are obtained from the finer grids by summing the values of S and M of the fine-grid cells lying inside a coarser-grid cell. The coarsest grid has $2 \times 2 \times 2$ grid cells. Here, the smoother acts as a direct solver.

When usual coarsening is applied the situation occurs as drawn in figure 11.

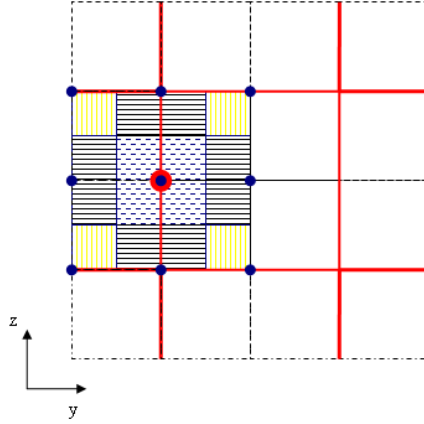


Figure 11: Schematic representation of the full weighting restriction operator based on dual volumes.

Figure 11 shows the grid from one side, the x -direction is perpendicular to the paper. The blue dots represent the fine grid electric field components, E_1 , lying at the edges of the grid cells. The red dot in the middle represents the coarse grid electric field component E_1 . The thick red lines define the edges of the coarse-grid cells, whereas the black lines represent the edges of the fine grid cells.

The idea behind the restriction of the fine grid components to coarse grid components is that the calculation of the coarse grid electric field components is in fact a weighted summation of surrounding fine grid electric field components. Each fine grid electric field component is multiplied by a weighting factor where upon the values of all 18 surrounding components are summed up to become

the value of one coarse grid electric field component. The question remains how to determine the weighting factors.

Keep in mind that in previous sections the dual volumes were introduced. These volumes determine to what extent a fine-grid electric field component should be taken into account when restricting to a coarser grid. In the figure above, the yellow and blue striped rectangles represent the parts of the dual volumes of the fine grid electric field components which lie in the dual volume of the coarse-grid electric field component. The weighting factor for each blue dot is the quotient of how much of the dual volume lies in the coarse-grid dual volume and the complete dual volume of this fine-grid electric field component. One can see and easily deduce that for the blue dots above this will be factors $\frac{1}{8}, \frac{1}{4}, \frac{1}{2}$ and 1.

It should be noted that this restriction operator is second-order accurate (exact for linear functions in a pointwise sense) on equidistant grids, but only first-order accurate on stretched grids. Another, simpler choice can be made for this restriction operator. However, numerical experiments on stretched grids show that this choice leads to divergence in some cases.

2.2.4 Prolongation

After computing the exact or approximate solution of the discrete equations on the coarse grids, the solutions need to be interpolated back to the fine grid and added to the fine-grid solutions. A natural prolongation operator is the scaled transpose of the restriction operator. This means that the same weights as above are applied for the electric field components. This operator is identical to constant interpolation in the coordinate direction of the component and bilinear interpolation in the other coordinates.

In schematic form, 2D prolongation for the electric field components in x -direction lying in the x, y -plane:

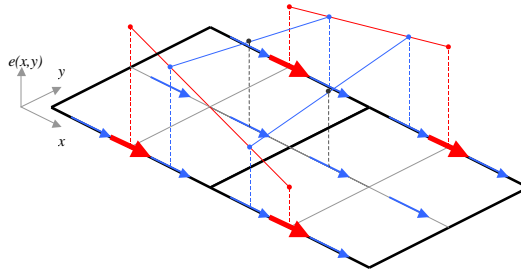


Figure 12: 2D prolongation

In the x, y -plane the red and blue arrows represent the electric field components on the coarse and fine grid respectively. Furthermore, the bold, black lines represent the coarse grid and the gray lines represent the fine grid. The third direction drawn in figure 12 is not the z -direction as defined in previous

sections but is the function value of the update of the error of the electric field components: $x, y \mapsto e(x, y)$.

Now suppose that on the coarse grid, the error of the electric field components in x -direction is updated with a smoother. The values of these updates are plotted as red dots above the red arrows. As mentioned above, in order to prolongate these errors to the fine grid, linear interpolation is used (see figure 13).

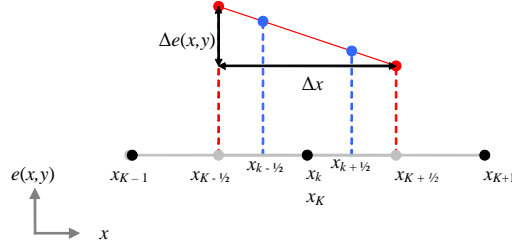


Figure 13: Linear interpolation

The error update of the electric field component at the middle of the edge of the coarse grid cell (x_{K-1}, x_K) lies above $x_{K-\frac{1}{2}}$. Note that this coarse grid point coincides with fine grid point x_{k-1} as described in section (1.3). The function of the error update $\delta e(x, y)$ is linearized around $x_{K-\frac{1}{2}}$,

$$\delta e(x, \cdot) = \delta e(x_{K-\frac{1}{2}}, \cdot) + \delta e'(x_{K-\frac{1}{2}}, \cdot)(x - x_{K-\frac{1}{2}})$$

One can easily see that this function transforms to,

$$\delta e(x, y) = \delta e(x_{K-\frac{1}{2}}, y) + \frac{\delta e(x_{K+\frac{1}{2}}, y) - \delta e(x_{K-\frac{1}{2}}, y)}{x_{K+\frac{1}{2}} - x_{K-\frac{1}{2}}}(x - x_{K-\frac{1}{2}})$$

The error updates of the electric fields on the fine grid between $x_{K-\frac{1}{2}}$ and $x_{K+\frac{1}{2}}$ can now be calculated with this local linearization around $x_{K-\frac{1}{2}}$. When these updates are obtained the (blue) fine grid components which do not coincide with the coarse grid can be computed with the same procedure, linearizing the error update around e.g. $(x_{k-\frac{1}{2}}, y_l)$ and $(x_{k-\frac{1}{2}}, y_{l+1})$.

In practice, this can be done by a standard Matlab procedure called `interp` which can perform linear interpolations. The coarse grid, fine grid and error correction of the electric field components on the coarse grid are fed to the Matlab function. The output of the function is the linear interpolation of the coarse grid electric field components to the fine grid.

2.3 Numerical experiments

In this section a short summary is given of the problems encountered in the performance of the multigrid solver. First a test problem will be introduced.

This problem is defined by Mulder and is a variant of a problem that can be found in [4]. The multigrid solver has been tested with this case. Problems that did arise from other test cases are listed at the end of this section.

2.3.1 Variable conductivity

This test problem is based on eigenfunctions and was first introduced in [4]. It has been modified by Mulder in [12] to allow for the use of perfect conduction (PEC) boundary conditions. The domain is $\Omega = [0, 2\pi]^3 m^3$. Define $\psi = \sin kx \sin ly \sin mz$ with k, l and m positive integers. Let the exact solution be,

$$E_1 = \alpha_1 \partial_x \psi, \quad E_2 = \alpha_2 \partial_y \psi, \quad E_3 = \alpha_3 \partial_z \psi$$

The domain Ω is split into two parts, Ω_1 with $z < \pi$ and Ω_2 for $z > \pi$, so that Ω is the union of their closure. The conductivity $\sigma = \sigma_0 + \sigma_1 (x + 1)(y + 2)(z - \pi)^2$ in Ω_1 , and $\sigma = \sigma_0$ in Ω_2 . Set $\epsilon_r = 0, \mu_r = 1$ and $\omega = 10^6 \text{Hz}$. The other parameters are chosen as $\alpha_1 = \alpha_2 = -2\text{V}, \alpha_3 = 1\text{V}, k = l = m = 1, \sigma_0 = 10\text{S/m}$ and $\sigma_1 = 1\text{S/m}$. Note that ψ is based on sine functions, causing the tangential electric field components to vanish at the boundaries in agreement with the PEC boundary conditions used here. The current source is defined by $\mathbf{J}_s = -\tilde{\sigma} \hat{\mathbf{E}} + \nabla \times (i\omega\mu)^{-1} \nabla \times \mathbf{E}$. Using the exact solution this results in:

$$\begin{aligned} \mathbf{J}_s = & -\tilde{\sigma} \begin{pmatrix} \alpha_1 \partial_x \psi \\ \alpha_2 \partial_y \psi \\ \alpha_3 \partial_z \psi \end{pmatrix} + \\ & (i\omega\mu)^{-1} \begin{pmatrix} [l^2(\alpha_1 - \alpha_2) + m^2(\alpha_1 - \alpha_3)] \partial_x \psi \\ [k^2(\alpha_2 - \alpha_1) + m^2(\alpha_2 - \alpha_3)] \partial_y \psi \\ [k^2(\alpha_3 - \alpha_1) + l^2(\alpha_3 - \alpha_2)] \partial_z \psi \end{pmatrix} \end{aligned}$$

In the next section a short description of grid-stretching is given. In the last section of this section a list can be found of the several tests and problems encountered by using the multigrid solver to solve this test problem.

Grid stretching The grid stretching used in [12] is called power law grid stretching. For instance, assume there are $N_x = N_y = N_z = 2^m + 1$ gridpoints in each direction. One option can be an equidistant grid with spatial mesh size $h = \frac{1}{N_x}$. However, suppose there is a small, compact area like a sphere were many discontinuities occur in the parameters of the equation. Furthermore, viewed from a particular distance from this area there are no problems with the parameters anymore. On an equidistant grid the spatial mesh size is the same. Because it would be very expensive to refine the whole grid, it may be useful to have more fine-grid cells only at the sphere and to keep the number of grid points the same. This can be reached by means of grid stretching.

The idea behind stretching is to make the cells around the object (origin) small and let the spatial mesh size increase gradually for each cell that sits further from this origin. The spatial mesh ratio between two arbitrary neighboring cells is the same, $1 + \alpha$. Obviously, this is called power law stretching as the spatial mesh size satisfies the following formula:

$$h_i(j) = h_i(0)(1 + \alpha)^j$$

Where $h_i(j)$ is the spatial mesh size of cell j in direction $i = x, y, z$. And $h_i(0)$ is the spatial mesh size of the first grid cell lying at the origin.

An example of a grid stretched from the source is presented in figure 14.

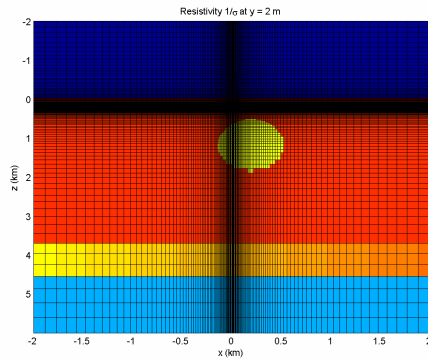


Figure 14: Example of grid stretching. The source is represented by the yellow ellipsis lying in the middle of the figure.

2.3.2 Problems encountered in test cases

This section contains a list of problems encountered when solving with the multigrid solver described in section (2.2).

1. Consider the test problem based on eigenvalues (2.3.1). Let $\sigma_0 = 10$ S/m and $\sigma_1 = 1$ S/m which avoids zero values for the conductivity.

N	h_{\max}	MG	bi
16	0.39	7	6
32	0.20	8	7
64	0.098	8	7
128	0.049	8	6

Table 2: Number of iterations for the first test problem with $\sigma_0=10$ S/m and $\sigma_1=1$ S/m

Table 2 lists the number of iterations and errors. The number of grid cells in each direction is given by $N_x = N_y = N_z = N$. For each grid, the number of iterations with pure multigrid (MG) and with multigrid as a preconditioner for **bicgstab2** (bi) are given. Note that **bicgstab2** costs a bit more per iteration because it requires an additional evaluation of the residual. Also, each **bicgstab2** step is counted as two iterations because it involves two multigrid cycles. Because convergence checks are carried out halfway and at the end of a full iteration step, the method may stop after an odd number of iterations. The iterations were stopped when the l_2 norm (see appendix (F)) of the residual had dropped a factor 10^{-8} from its original value for a zero solution. The results in the table above show grid independent convergence for the multigrid method. The number of iterations of **bicgstab2** is one less but not worth the extra cost in terms of cpu-time.

2. Again, consider the test problem based on eigenvalues. Let $\sigma_0 = 10$ S/m and $\sigma_1 = 1$ S/m and use power law grid stretching with a spatial mesh size ratio $(1 + \alpha)$ between two neighboring cells. Let $\alpha = 0.04$.

N	h_{\max}	MG	bi
16	0.45	8	6
32	0.26	11	8
64	0.17	12	14
128	0.13	81	32

Table 3: Number of iterations for the first test problem with $\sigma_0=10$ S/m, $\sigma_1=1$ S/m and power-law grid stretching with $\alpha = 0.04$

The grid independent convergence rates of multigrid are lost. However, **bicgstab2** is able to deal with the slow-converging components of the solution and needs significantly fewer iterations to converge.

In [12] several more test have been carried out. With different values for σ_0 and σ_1 , a minimum-norm solution is also constructed when a vacuum region appears. As can be expected, grid stretching causes even more problems in that case. For some parameter settings the multigrid method did not converge at all. The iterations were stopped when the norm of the residual failed to decrease. Even **bicgstab2** did not converge in fewer than 100 iterations in that particular situation.

Also more realistic test cases can be found in [12]. All cases show the same convergence pattern. Without grid stretching the method usually converges, but when the power law ratio α becomes significant large (> 0.02), multigrid breaks down or needs a very large number of iterations to converge. Stretching the grid has an effect similar to the use of variable coefficients, in this case $\mu_r^{-1}(\mathbf{x})$, inside the difference operators. When these coefficients show large variations or the grid is stretched the problem becomes anisotropic. Multigrid methods

based on pointwise smoothing a standard grid coarsening may fail to converge if there is strong anisotropy in the coefficients of the governing equation. In that case more sophisticated smoothers or coarsening strategies may be required. If slow convergence is caused by just a few components of the solution, a Krylov subspace method can be used to effectively remove them. In this particular case the matrix A is non-symmetric and complex. Therefore `bicgstab2` [19] is a suitable alternative, multigrid will be accelerated by the Krylov method. The multigrid can be seen as a preconditioner for `bicgstab2` in this case. The iterative scheme of `bicgstab2` can be found in appendix (D).

2.4 Anisotropy

In section (2.3) it has been mentioned that grid stretching causes anisotropy in the discretization of the Maxwell equations. Hence, the convergence rates of the multigrid solver deteriorate for fine grids. In this section a definition of anisotropy is given and afterwards it is shown how grid stretching can cause this effect.

2.4.1 Definition

The Maxwell equations were stated in 3D. But to get a clear view of the definition of anisotropy a 2D example for the Poisson equations is given first. In section (2.4.3) the extension from 2D to 3D will be carried out and possible cures for solving anisotropy will be introduced.

The discussion here is based on a 2D anisotropic elliptic model problem:

$$\begin{aligned} -\varepsilon\partial_{xx}e - \partial_{yy}e &= s^\Omega(x, y), & \left(\Omega = (0, 1)^2, \quad e = e(x, y) \right) \\ e &= s^\Gamma(x, y), & (\partial\Omega) \end{aligned} \quad (15)$$

Here $0 < \varepsilon \ll 1$, the case were $\varepsilon \gg 1$ is not any different, only the role of the directions x and y interchanges.

When discretizing equation (15) with a standard 5-point difference operator the following discrete problem is obtained:

$$\begin{aligned} L_h(\varepsilon)e_h &= s_h^\Omega, & (\Omega_h) \\ e_h &= \bar{s}, & (\Gamma_h) \end{aligned}$$

where Ω_h is a square grid with spatial mesh size $h = h_x = h_y$ (similar as in figure 8) and Γ_h is the set of gridpoints lying on the boundary of Ω . In stencil notation, the discrete operator $L_h(\varepsilon)$ reads

$$L_h(\varepsilon) = \frac{1}{h^2} \begin{bmatrix} & -1 & & \\ -\varepsilon & 2(1 + \varepsilon) & -\varepsilon & \\ & -1 & & \end{bmatrix}_h$$

In this case, the discrete anisotropy is aligned with the grid. In 2D such problems are characterized by the coefficients in front of the e_{xx} and e_{yy} terms, which may differ by orders of magnitude. In the next section the role of anisotropy introduced by discretization will be discussed, as it is the case with stretched grids.

The problem with the discrete operator $L_h(\varepsilon)$ is that if ε goes to 0, the h -ellipticity measure [17] of the anisotropic operator tends to 0. In that case the smoothing properties of a standard pointwise smoothing scheme will deteriorate for $\varepsilon \rightarrow 0$. Suppose that a standard pointwise relaxation such as Gauss-Seidel in lexicographical order (GS-LEX) is applied to the system above. Then it will appear that the smoothing effect of the error is poor with respect to the x -direction. Pointwise relaxation has a smoothing effect only with respect to the "strong coupling" in the operator, i.e. the y -direction. If the error is plotted after several iteration steps, the error will be smooth in y -direction and irregular in the x -direction.

For example, consider GS-LEX, then the error relation becomes:

$$\begin{aligned} \delta e_h^{m+1}(x_k, y_l) &= \frac{1}{2(\varepsilon + 1)} [\varepsilon \delta e_h^{m+1}(x_{k-1}, y_l) + \varepsilon \delta e_h^m(x_{k+1}, y_l) \\ &\quad + \delta e_h^{m+1}(x_k, y_{l-1}) + \delta e_h^m(x_k, y_{l+1})] \end{aligned}$$

Now let $\varepsilon \rightarrow 0$,

$$\delta e_h^{m+1}(x_k, y_l) = \frac{1}{2} [\delta e_h^{m+1}(x_k, y_{l-1}) + \delta e_h^m(x_k, y_{l+1})]$$

Obviously, there is no averaging effect with respect to the x -direction and therefore no smoothing with respect to this direction is achieved. Such non smooth errors can no longer be efficiently reduced by means of a coarser grid which is obtained by standard coarsening, i.e. by doubling the mesh size in both directions.

This failure can also be explained and verified by applying local mode analysis [17],[15] to the GS-LEX smoother for the problem. The multigrid convergence factor will increase towards 1 for $\varepsilon \rightarrow 0$ or $\varepsilon \rightarrow \infty$. In general, pointwise relaxation and standard coarsening is not a feasible combination for highly anisotropic problems.

The next section will list solutions for this complication.

2.4.2 Anisotropy on stretched grid

As mentioned above, when stretched grids are used, the discretization may introduce anisotropies. Consider a standard 2D Poisson-like elliptic equation:

$$\begin{aligned} -\partial_{xx}e - \partial_{yy}e &= s^\Omega(x, y), \quad \left(\Omega = (0, 1)^2, \quad e = e(x, y) \right) \\ e &= s^\Gamma(x, y), \quad (\partial\Omega) \end{aligned} \quad (16)$$

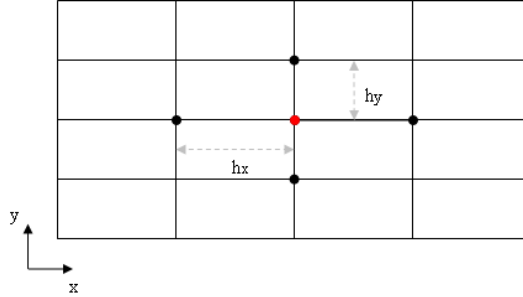


Figure 15: Example of stretched grid

Introduce the following stretched grid, presented in figure 15

with for example, $h_x = \frac{h_y}{\sqrt{\varepsilon}}$, $0 < \varepsilon < 1$. The use of standard central differences for the finite difference discretization with $h_y = h$ leads to the following stencils

$$\begin{aligned} \partial_{xx}e &\simeq \varepsilon \frac{e(x_{k-1}, y_l) - 2e(x_k, y_l) + e(x_{k+1}, y_l)}{h^2} + O(h^2), \\ \partial_{yy}e &\simeq \frac{e(x_k, y_{l-1}) - 2e(x_k, y_l) + e(x_k, y_{l+1}))}{h^2} + O(h^2). \end{aligned}$$

Obviously, the same discrete operator as for equation (15) is obtained.

With more complex variants of the stretched grid, e.g. the one used in the test problem above (power law grid stretching), one will encounter long thin mesh cells as well as wide cells. This indicates that discrete problems that have varying anisotropies in the x - as well as in the y - direction will be encountered.

2.4.3 Anisotropy in 3D

In this section the extension will be made from 2D to 3D elliptic anisotropic problems. The principal phenomena in the 3D anisotropic case are similar to those in 2D. However, with respect to optimal algorithms, the situation is somewhat more involved than in 2D. The possibilities for choosing different multigrid components will be introduced in the next section.

The discussion here is based on the 3D anisotropic elliptic model problem

$$\begin{aligned} -ae_{xx} - be_{yy} - ce_{zz} &= s^\Omega(x, y, z) \quad (\Omega = (0, 1)^3) \\ e &= s^\Gamma(x, y, z) \quad (\Gamma = \partial\Omega) \end{aligned} \quad (17)$$

Assume that the operator is elliptic, that all coefficients a, b and c have the same sign and that (17) is discretized on $G_h := \{(x, y, z) : x = ih_x, y = jh_y, z = kh_z; i, j, k \in \mathbb{Z}\}$

by the well-known seven-point discretization. In 3D stencil notation, the discrete operator is

$$\frac{1}{h^2} \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & -c & 0 \\ 0 & 0 & 0 \end{array} \right]_h \left[\begin{array}{ccc} & -b & \\ -a & 2(a+b+c) & -a \\ & -b & \end{array} \right]_h \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & -c & 0 \\ 0 & 0 & 0 \end{array} \right]_h$$

For constant coefficients, four representative parameter sets can be distinguished (without loss of generality):

$$\begin{aligned} \text{Case 1} & : a \approx b \approx c \\ \text{Case 2} & : a \gg b \approx c \\ \text{Case 3} & : a \approx b \gg c \\ \text{Case 4} & : a \gg b \gg c \end{aligned} \tag{18}$$

For Case 1 of (18) standard coarsening combined with an Gauss-Seidel red-black smoother results in an efficient multigrid solver [17] as there is no anisotropy in equations (17) as h_x, h_y and h_z are of the same size.

In Cases 2,3 and 4 there is a strong coupling of unknowns in multiple directions. A general rule in the case of standard coarsening, is that smooth errors in all coordinate directions are obtained if all strongly coupled unknowns are relaxed collectively.

In Case 2 there is strong coupling of unknowns in only the x -direction. Hence, relaxing all the unknowns of each single line in that direction simultaneously can be used safely and will result in excellent smoothing [17]. This is line smoothing in x -direction and will be introduced section (2.5.2).

For Cases 3 and 4 the situation is more involved. In Case 3, all unknowns lying in the same (x, y) -plane are strongly coupled. Hence, all these unknowns should be solved collectively. This is plane relaxation and will be introduced in section (2.5.3).

For Case 4 the situation is similar as in Case 3. Proper smoothing is guaranteed by (x, y) -plane relaxation. If this relaxation is performed by 2D multigrid, one has to take into account that the corresponding discrete 2D problems are now anisotropic themselves (in each plane). Thus all x -line smoothing should be used within the 2D multigrid plane solver.

However, these solutions apply only to the 3D anisotropic elliptic model problem. The Maxwell equations on stretched grids are more involved. In next section (2.5) multigrid solutions for anisotropy are introduced.

2.5 Multigrid solutions for anisotropy

There are various solutions which can reduce the effects of anisotropy [17]. In this section three types of solutions will be highlighted. First, cell-block smoothing in combination with multiple semi-coarsening. Secondly, a combination of line smoothing and semi-coarsening. Thirdly, standard and semicoarsening in

combination with plane relaxation. In section (3.1.3) adapted multigrid schemes based on the improvements introduced are given.

In case of the Maxwell equations cell-block smoothing means pointwise smoothing for e.g. the Poisson equation in section (2.5).

2.5.1 Cell-block smoothing and semicoarsening

There is the option of using the cell-block smoothing in combination with semicoarsening strategies instead of line or plane relaxations and standard coarsening. First semicoarsening is introduced, next the semicoarsening strategies will be highlighted.

Definition semicoarsening Again, consider the example of the 2D elliptic equation (16) with anisotropy due to grid stretching. The idea is to keep pointwise relaxation for smoothing but to change the grid-coarsening according to the one-dimensional smoothness of errors. This means that a coarser grid is defined by doubling the mesh size only in that direction in which the errors are smooth:

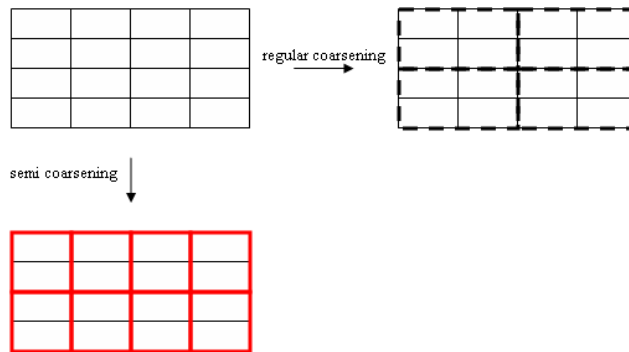


Figure 16: Semicoarsening strategies

In the case of regular coarsening, after one grid coarsening, the grid goes from 4x4 cells to 2x2 cells. However, the cells remain stretched and this yields an anisotropic discrete operator for equation (16). Now by applying x -semicoarsening, after one grid coarsening, the grid of 4x4 cells coarsens to 4x2 grid cells. It is easy to see that the grid cells are, in the particular case of figure 16, squares. Hence, $H_x = H_y$ and the anisotropy at this coarser grid will not be present in the coarse grid discretization of equation (16). Local mode analysis of the error indicates that the quality of a smoother depends on the range of high frequencies and thus on the choice of the coarse grid.

There are some practical complications with the use of semicoarsening. The restriction and prolongation operators have to be modified to appropriate transfer quantities. It is obvious that the transfer operators use different 1D electric

field components to compute coarse-grid electric field components when the structure of the grid changes.

Semicoarsening strategies Again, consider the four cases resulting from the 3D anisotropic elliptic model introduced in section (2.4.3). For Case 2, the strong coupling was in one direction and x -semicoarsening, i.e. coarsening only in the x -direction, will result in an efficient solver if combined with a point smoother. Similarly, y -semicoarsening and z -semicoarsening will result in satisfactory solution methods if the strong coupling is in only one of these directions. For Case 3, with strong coupling in two directions, coarsening should take place in two directions simultaneously.

For the Maxwell equations on stretched grids, it remains to be studied which semicoarsening strategy improves the multigrid solver. Because as mentioned before, with power-law grid stretching applied to the discretization of the Maxwell equations the strong coupling in the unknowns differs for each grid cell and can be in multiple directions. Hence, alternating semicoarsening in multiple directions can be a useful choice. For each multigrid F-cycle a different coarsening strategy may be applied. First coarsening in x, y -direction and applying the cell-block smoother, in the next F-cycle coarsening in y, z -direction and in the third coarsening in x, z -direction should take place. When this sequence is done, the pattern is repeated.

2.5.2 Line smoothing and semicoarsening

Another possible solution for solving 3D anisotropies in the discretization is the use of a 2D multiple semicoarsening strategy and a line relaxation in the third dimension. In order to obtain a robust solver for general anisotropies, this third dimension should not be coarsened [17].

Definition line smoother The line smoother updates all unknowns lying one a line simultaneously. The line smoother in x -direction will update all the unknowns lying on a line $[x_1, x_{N_x}]$ keeping y_l and z_m constant. For y (and in 3D z -line smoothing) the modification is straightforward.

Gauss-Seidel-type line relaxations are particularly efficient smoothers for anisotropic problems if the anisotropy is aligned with the grid. This is due to the general observation that errors become smooth in both directions if strongly connected unknowns are updated collectively. With grid stretching only in x -direction as in section (2.4.2) appears, obviously, there is a strong coupling between the unknowns in x -direction.

In [17] LFA analysis is applied to a lexicographic line Gauss-Seidel smoother confirming excellent smoothing factors. Alternating line smoothers can deal with problems with varying anisotropy within one problem as it happens in the case of stretched grids. There are, of course, other line smoothers one can use, e.g. ω -Jacobi or zebra line Gauss-Seidel smoothing (line smoother x -direction, first all even rows in y -direction and next the odd rows). However, as in [12] a cell-block symmetric GS-LEX is preferred it is expected that a lexicographical

Gauss-Seidel line smoother will give an improvement in the performance of the multigrid solver with stretched grids.

2.5.3 Standard and semicoarsening with plane relaxation

Due to the grid stretching the discretization of the Maxwell equations became anisotropic. However, as stretching is applied in three directions it is not straightforward to see in which direction there is a strong coupling of unknowns (see section (2.4.3)). Compared to the 3D anisotropic elliptic model problem of section (2.4.3), the parameters a , b and c will be of different value for each cell in the grid. For problem (17) with (smoothly varying) variable coefficients $a(x, y, z)$, $b(x, y, z)$ and $c(x, y, z)$ robust multigrid methods exist which have good convergence properties [17] independent of the size of the parameters a , b and c . One robust 3D method is based on the combination of standard coarsening and alternating plane relaxation for smoothing. Each smoothing step consists of applying three plane relaxations (an (x, y) -, (y, z) - and an (x, z) -plane relaxation). If 2D multigrid is used as the plane solver, alternating line relaxation within each plane solver is required in order to guarantee good smoothing properties for all choices of a, b and c .

In the case of the Maxwell equations there is no information about the relative size of the coefficients and hence about the direction of strong coupling of the unknowns. A robust multigrid variant, based on semicoarsening, is to coarsen only along one coordinate, say z , while employing plane relaxation for x and y [17].

However, the costs of plane relaxation will be very high. The Maxwell equations are evaluated on grids up to $2^8 + 1 = 256$ grid points in each direction. Plane relaxation for a system of equations may result in large computation times. An alternative approach for this robust variant is to "mimic" the effect of plane relaxation by using line smoothers in two directions. The main idea is to keep the number of grid points in one direction, say z , constant and apply line smoothers in the x and y direction in which the coarsening takes place.

3 Improving solver

In this section the multigrid solver will be adapted in order to improve its performance on stretched grids. The modifications of the multigrid solver are introduced step by step. First, the inclusion of semicoarsening will be discussed, hence the restriction and prolongation transfer operators have to be adapted. Secondly, the line smoother is discussed. At the end of the section two possible combinations of the line smoother and semicoarsening are given and motivated. These two adapted multigrid solvers will be used for the test cases in the second part.

The first test problem in section (2.3.1) will also be the first test case here and two new cases are included. In the last section the results of all the cases are collected.

3.1 Modifications multigrid for Maxwell equations

In section (2.5) it has been shown that multigrid with standard coarsening and cell-block smoothing revealed a performance leak for stretched grids. The anisotropy in the discretization of the Maxwell equations is due to grid stretching. In other words, because of the big differences in mesh sizes of the grid cells due to grid stretching there will be strong coupling in the operator if there are coefficients in e.g. x -direction which are multiplied by a large weight and coefficients in y -direction which are multiplied by a small weight. In the direction of the strong coupling the cell-block smoother *can* smooth the error but in the directions of weak coupling the cell-block smoother will perform poorly. According to section (2.5) it is to be expected that semicoarsening and the use of a line smoother will successfully decrease the error.

3.1.1 Semicoarsening

The implementation of semicoarsening is straightforward compared to the line smoother. In the grid introduced in section (1.3), there are three directions x, y and z in which the number of grid points is equal, $N = 2^m + 1$, $m = 1, 2, \dots, 8$. The idea behind a 2D semicoarsening is to fix the number of grid points in one direction and to apply standard coarsening in the remaining two directions. With power-law grid stretching in three directions (as will be used in all three test cases) semicoarsening produces rectangular grid cells of different size.

In order to make a semicoarsening strategy two multigrid components have to be adapted, the restriction and prolongation transfer operators. The discrete operator on the coarse grids is chosen to be the one obtained by direct discretization. The cell-averaged material properties are obtained from the finer grids by summing the values of S and M (parameters discretization, section (1.3)) of the fine-grid cells lying inside a coarser-grid cell.

Restriction When semicoarsening is applied in x -direction one would get the restriction operator as sketched in figure 17.

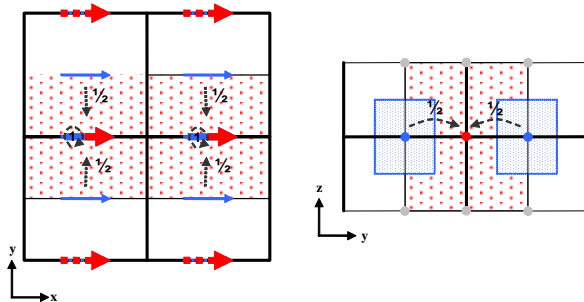


Figure 17: Restriction of fine grid components to coarser grids with semi coarsening.

Because the grid cells are stretched in x -direction, the number of grid points is fixed there. In figure 17 the same definition for the arrows as for standard coarsening (section (2.2.3)) is used. The coarse grid in figure 17 consists of cubic cells.

Eventually, two types of restriction will have to be implemented for semi-coarsening. In the two directions with standard coarsening the weights as defined in figure 11 are used. In the direction where no coarsening takes place, the weights as defined in figure 17 have to be used. The major difference between figure 11 and 17 is the number of fine grid electric field components contributing to a coarse grid electric field component. Instead of eighteen (standard coarsening) just nine (3D) fine grid components will be used. The calculation of the weights is the same. The ratio between the quantity of the fine grid dual volume lying in the coarse grid dual volume determines the weight.

Prolongation In section (2) the prolongation operator is worked out in detail. There is a special Matlab function `interp` which can perform linear interpolations of the error between fine and coarse grids. Despite the fact that when semicoarsening is applied and thus the structure of coarse grids changes with respect to the coarse grids resulting from standard coarsening, the prolongation function adapts easily to these other grids. Only grid indices have to be adapted when calling on the Matlab function. These modifications and the standard coarsening Matlab procedures can be found in appendix (G). Hence, no further explanation is necessary for the interpolation of the coarse grid electric field components in case of semicoarsening to the fine grid.

3.1.2 Line smoother

The construction and implementation of the line smoother is less trivial compared to the adaptations for semicoarsening. The idea behind the line smoother is to update all unknowns lying on a straight line simultaneously. If the equations for one direction are known, the two remaining directions can be written

out in a similar manner. Therefore, in this section only the x -direction is discussed in detail.

When simultaneously relaxing all electric field components on a straight line, one will get a structure of the line smoother as shown in figure 18.

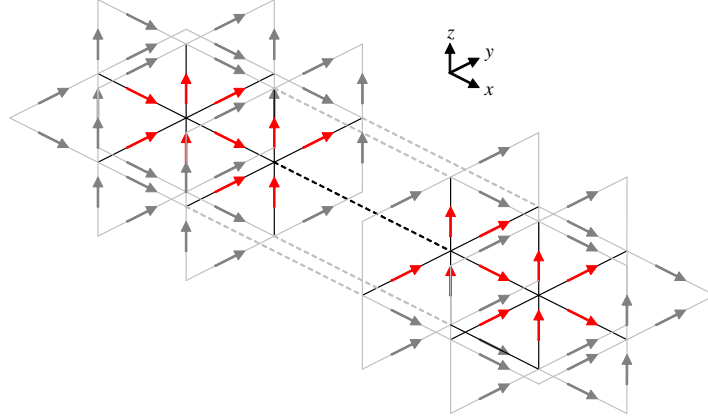


Figure 18: Schematic representation of line smoother. All unknowns needed for one line relaxation are drawn.

The red arrows represent the electric field components lying at the middle of the cell edges. The gray arrows represent the electric field components which are necessary in order to update the (red) unknown electric field components lying around the grid nodes. The major difference between this line smoother and the cell-block smoother, which is drawn in figure (10), is that the number of unknowns to be solved simultaneously is increased.

Suppose the number of grid nodes in x -direction is N_x , than.

$$\hat{A}\hat{\mathbf{E}}_{l,m} = s - \hat{B}\hat{\mathbf{E}}_{sur} \quad (19)$$

represents the system resulting from the line smoother of figure 18.

First, the structure of vector $\hat{\mathbf{E}}_{l,m}$ will be discussed. The cell-block smoother had six unknown electric field components collected in vector $\mathbf{E}_{k,l,m}$ of system (14)

$$\mathbf{E}_{k,l,m} = \left[E_{1,k-\frac{1}{2},l,m}, E_{1,k+\frac{1}{2},l,m}, E_{2,k,l-\frac{1}{2},m}, E_{2,k,l+\frac{1}{2},m}, E_{3,k,l,m-\frac{1}{2}}, E_{3,k,l,m+\frac{1}{2}} \right]^T$$

where $E_{1,k-\frac{1}{2},l,m}, \dots, E_{3,k,l,m+\frac{1}{2}}$ are defined and placed as in figure (9). The line smoother has $5 \cdot N_x + 1$ unknowns

$$\hat{\mathbf{E}}_{l,m} = \begin{bmatrix} E_{1,\frac{1}{2},l,m} & E_{2,\frac{1}{2},l-\frac{1}{2},m} & E_{2,l+\frac{1}{2},m} & E_{3,\frac{1}{2},l,m-\frac{1}{2}} & E_{3,\frac{1}{2},l,m+\frac{1}{2}} & \cdots \\ E_{1,k,l,m} & E_{2,k,l-\frac{1}{2},m} & E_{2,k,l+\frac{1}{2},m} & E_{3,k,l,m-\frac{1}{2}} & E_{3,k,l,m+\frac{1}{2}} & \cdots \\ E_{1,N_x-1,l,m} & E_{2,N_x-1,l-\frac{1}{2},m} & E_{2,N_x-1,l+\frac{1}{2},m} & E_{3,N_x-1,l,m-\frac{1}{2}} & E_{3,N_x-1,l,m+\frac{1}{2}} \\ E_{1,N_x,l,m} \end{bmatrix}^T$$

A powerful processor (e.g., 64-bit) is needed to solve system (19) for large N_x, N_y and N_z as this may be a costly process.

Matrix \hat{A} has a special structure. It is complex, symmetric and sparse. Note that the line to be solved represents in fact, a chain of 5-point molecules. One molecule consists of five electric field components lying around one grid node. This is an important observation as it makes the derivation of the equations concerning the line smoother easier.

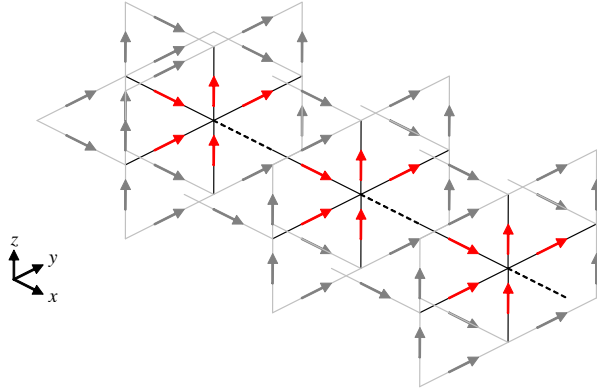


Figure 19: Schematic representation of the line smoother. The 5-point molecules are coupled in order to obtain one line relaxation.

In figure 19 each 5-point molecule is surrounded by numerous (gray) electric field components. As only the (red) unknown electric field components are updated, the surrounding components have to be temporarily fixed values. As the line solver processes line-by-line moves, the other (gray) known electric field components will be updated and the roles of the components interchange.

Now that the ordering of the unknowns is determined, the derivation of the equations of the line smoother is straightforward. Because of the coupling between the 5-points molecules matrix \hat{A} will have a special form shown in figure 20.

In figure 20 the structure of the line smoother and the five point molecules can be seen. The matrix blocks at the main diagonal represent the coefficients of the unknowns scattered around the grid nodes. The off-diagonal blocks represent the coupling between the 5-point molecules. The extra, small two blocks

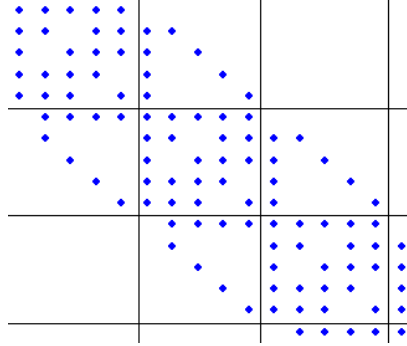


Figure 20: Sparsity structure of the line smoother. The blue dots represents non-zero entries of the line smoother matrix.

at the last two rows of blocks of the matrix in figure 20 represent the single 1-point molecule at the end of the line smoother (see also figure 18). Hence, the boundaries will be treated separately in the next derivation.

With figure 20 and the structure of the line smoother of figure 18 in mind, matrix \hat{A} is splitted up in four types of submatrices

$$\hat{A} = \begin{bmatrix} \hat{A}_{\Gamma_1} & \hat{A}_{int} & & \emptyset \\ & \ddots & & \\ & & \hat{A}_{int} & \\ \emptyset & & & \hat{A}_{\Gamma_2} \\ & & & & \hat{A}_{\Gamma_3} \end{bmatrix}$$

1. \hat{A}_{int} , This submatrix represents all interior 5-point molecules that do not need the electric field components lying at the edges of the domain. Due to the structure of the line, submatrix \hat{A}_{int} can also be splitted up,

$$\hat{A}_{int} = (\hat{A}_{int,k-1,l,m} \quad \hat{A}_{int,k,l,m} \quad \hat{A}_{int,k+1,l,m})$$

where 5x5 matrices $\hat{A}_{int,k-1,l,m}$ and $\hat{A}_{int,k+1,l,m}$ represent the coupling of the 5-point molecule at grid point k, l, m with surrounding electric field components belonging to the 5-point molecules at grid point $k - 1, l, m$ and $k + 1, l, m$ respectively. The 5x5 small matrix $\hat{A}_{int,k,l,m}$ contains the coefficients of the electric field components lying at the edges surrounding the node k, l, m . For convenience, the entries of the matrices can be found in appendix (B).

2. \hat{A}_{Γ_1} , This submatrix represents the 5 point molecule lying at nodes $1, l, m$, i.e. in the upper-left corner of figure 18. Figure 18 indicates that there will be no coupling with a node lying left to node $1, l, m$. However, due to

the PEC boundary conditions the values of the electric field components on the boundary are set to zero and therefore the contribution of the boundary to matrix \hat{A}_{Γ_1} is also zero. Hence, \hat{A}_{Γ_1} can be splitted up as,

$$\hat{A}_{\Gamma_1} = \begin{pmatrix} \hat{A}_{\Gamma_1,1,l,m} & \hat{A}_{\Gamma_1,2,l,m} \end{pmatrix}$$

where 5x5 small matrix $\hat{A}_{\Gamma_1,2,l,m}$ represents the coupling of the 5-point molecule at grid point 1, l, m with the electric field components belonging to the 5-point molecule at grid point 2, l, m . The 5x5 small matrix $\hat{A}_{\Gamma_1,1,l,m}$ contains the coefficients of the electric field components lying at the edges surrounding the node k, l, m . The entries of both matrices can be found in appendix (B).

3. \hat{A}_{Γ_2} , This submatrix represents the 5 point molecule lying at node $N_x - 1, l, m$. It lies in the lower-right corner of figure 18. The coupling of the 5-point molecule with the molecule surrounding node N_x, l, m differs from previous nodes. The last molecule consists of just one electric field component at the edge between $N_x - 1, l, m$ and N_x, l, m . Due to the PEC boundary conditions the values of the electric field components lying at the edges of the boundary are set to zero, therefore no contribution to matrix \hat{A}_{Γ_2} will be made. Hence, \hat{A}_{Γ_2} can be splitted up as,

$$\hat{A}_{\Gamma_2} = \begin{pmatrix} \hat{A}_{\Gamma_2,N_x-2,l,m} & \hat{A}_{\Gamma_2,N_x-1,l,m} & \hat{A}_{\Gamma_2,N_x,l,m} \end{pmatrix}$$

where 5x5 matrix $\hat{A}_{\Gamma_2,N_x-2,l,m}$ represents the coupling of the 5-point molecule at grid point N_x-1, l, m with surrounding electric field components belonging to the 5-point molecule at grid point $N_x - 2, l, m$. The 5x1 small matrix $\hat{A}_{\Gamma_2,N_x,l,m}$ contains the coefficients of electric field component $E_{1,N_x-\frac{1}{2},l,m}$. The 5x5 small matrix $\hat{A}_{\Gamma_2,N_x-1,l,m}$ contains the coefficients of the electric field components lying at the edges surrounding the node $N_x - 1, l, m$. The entries of the matrices can be found in appendix (B).

4. \hat{A}_{Γ_3} , This submatrix represents the 1-point molecule lying around node N_x, l, m . The latter lies in the lower-right corner of figure 18. Due to the PEC boundary conditions the values of the electric field components lying at the boundary is set to zero. Hence, there is no contribution of these elements to matrix \hat{A}_{Γ_3} . Matrix \hat{A}_{Γ_3} can be splitted up as,

$$\hat{A}_{\Gamma_3} = \begin{pmatrix} \hat{A}_{\Gamma_3,N_x-1,l,m} & \hat{A}_{\Gamma_3,N_x,l,m} \end{pmatrix}$$

where 1x5 small matrix $\hat{A}_{\Gamma_3,N_x-1,l,m}$ contains the coefficients of the electric field components lying around node $N_x - 1, l, m$. The 1x1 sub-sub matrix $\hat{A}_{\Gamma_3,N_x,l,m}$ contains the coefficient of electric field component $E_{1,N_x-\frac{1}{2},l,m}$. All entries of the matrices can be found in appendix (B).

The construction of the line smoother is now complete. The question remains how to manage the line smoother and semicoarsening in the proper way to obtain an improved multigrid solver. In the next section a choice will be made and motivated.

3.1.3 Multigrid scheme

Two improved strategies are introduced here and implemented so that multigrid convergence rates will improve on stretched grids. As mentioned earlier the F-cycle appears to be an effective multigrid strategy (2). Therefore this scheme will be the cycle-of-choice.

First, the following notation is introduced for convenience,

1. PS : cell-block smoother
 - (a) PS-pre : pre smoothing
 - (b) PS-post : post smoothing
2. xyz-LS : line smoother in x -direction, followed by a line smoothing sweep in y -direction and z -direction
 - (a) xyz-LS-pre : pre smoothing
 - (b) xyz-LS-post : post smoothing
3. std-r : standard restriction in all directions
4. xy-r : restriction of fine grid to coarse grid only in directions x and y . The number of grid points in z -direction remains the same
5. std-p : standard prolongation in all directions
6. xy-p : prolongation of coarse grid to fine grid only in directions x and y . The number of grid points in z -direction remains the same

It is difficult to visualize the F-cycle strategy for the general case, therefore suppose that the finest grid contains $N = 2^4 + 1 = 17$ grid points in each direction. Obviously, the following strategies can easily be extrapolated to the general case.

Standard coarsening and point smoother

The F-cycle scheme of the original multigrid solver, cell-block smoother with standard coarsening,

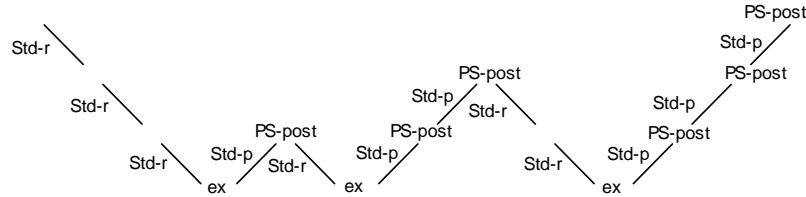


Figure 21: F-cycle with block post-smoothing and standard coarsening.

In this strategy no pre-smoothing steps are applied and standard coarsening is carried out.

Standard coarsening and line smoother

The first alternative for the original multigrid solver is the use of the line smoother in combination with standard coarsening,

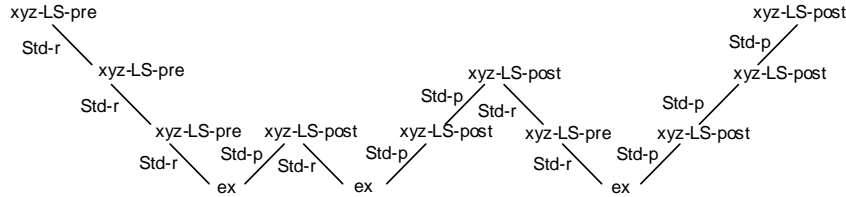


Figure 22: F-cycle with line pre- and post-smoothing and standard coarsening.

A robust version of the line smoother has been chosen. Each smoothing step consists of six sweeps through the grid. First, one sweep with the x -direction line smoother and due to the *symmetric* version, a second sweep in reversed order. Then one sweep with the y -direction line smoother and a sweep in reversed order. It is followed by one sweep with the z -direction line smoother and a sweep in reversed order.

In this strategy one pre-smoothing steps is applied. Compared to the original version without pre-smoothing more smoothing sweeps have to be carried out. It is expected that the error becomes significantly smoother at least in the direction of the line.

In this method, standard coarsening is carried out. The line smoother is supposed to deal with the anisotropy in the discrete equations due to the grid

stretching. If linewise anisotropies occur, semicoarsening would be a feasible approach. However, due to the power-law grid stretching strong coupling appears in varying directions and therefore semicoarsening may fail [17].

Semi coarsening and line smoother

The second alternative is to combine line smoothing and semicoarsening in an appropriate fashion. An example is given in figure 23.

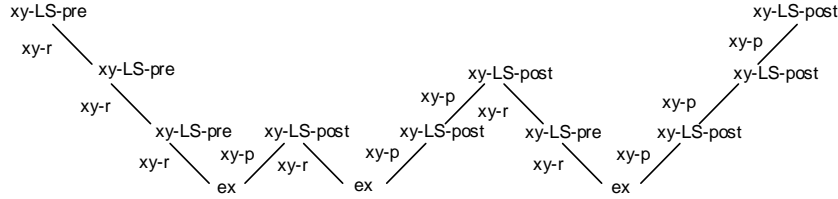


Figure 23: F-cycle with line pre- and post-smoothing and semi coarsening.

This strategy has two differences compared to the first alternative multigrid solver. First, the line smoother is not used in three but in two directions. Hence, one pre-smoothing step is cheaper in terms of memory space and cpu-time. Secondly, semicoarsening is used instead of standard coarsening. In fact a plane solver is simulated (see section (3.1))

Suppose applying one x, y -line pre-smoothing step. When restricting the electric field components to the coarser grid, the error in the x, y -direction may be smooth. During one F-cycle the direction of the semicoarsening and line smoother remains the same. When the F-cycle is finished the direction changes and a suitable new direction is chosen. For instance the following scheme is chosen,

F-cycle #1: x, y -line smoother, restriction and prolongation only in x, y -direction, number of points in z -direction remains the same

F-cycle #2: y, z -line smoother, restriction and prolongation only in y, z -direction, number of points in x -direction remains the same

F-cycle #3: x, z -line smoother, restriction and prolongation only in x, z -direction, number of points in y -direction remains the same

Obviously, when F-cycle #3 is performed, the process continues with F-cycle #1 and so on. In this way a plane solver is simulated.

3.2 Local mode analysis

In this section a first attempt of a local mode analysis (LMA) of the cell-block smoother and 3D line smoother is presented. LMA is a powerful tool for quantitative analysis and the design of efficient multigrid methods for general problems

according to [17]. Furthermore, because of the non-trivial discretization due to the curl-curl operator of the Maxwell equations, a special form of LMA will be used here. The local mode analysis which has been performed by Shah [15] who carried out an analysis of the multigrid components solving the incompressible Navier-Stokes equations in 2D, will be used as a guide line. Before introducing the governing equations and carrying out the analysis the basics are presented to get a better insight in the basic principles of local mode analysis.

LMA can be a useful tool to obtain information about smoothing factors, two-grid convergence factors as well as error reduction factors. In this section the focus will be at the smoothing factors as the analysis on the remaining two factors is beyond the scope of this thesis.

3.2.1 Basics LMA

In LMA linear discrete operators with constant coefficients are considered. Any general discrete operator can be linearized locally and can be replaced locally by an operator with constant coefficients. However, the discretization of the Maxwell equations described above is linear already. So no further assumptions or adaptations to the system have to be made. All considerations in the context of LMA are based on grid functions of the form [17]:

$$\varphi(\boldsymbol{\theta}, \mathbf{x}) = e^{i\boldsymbol{\theta} \cdot \mathbf{x}/h}$$

where \mathbf{x} varies on the given infinite grid G_h and $\boldsymbol{\theta}$ is a continuous parameter that characterizes the frequency of the grid function under consideration.

In general, all multigrid operators (discretization L_h , smoothing and inter-grid transfer operators S_h, I_h^H, I_h^h , the coarse grid operator L_H and thus coarse grid correction and two-grid operators) will operate on these grid functions $\varphi(\boldsymbol{\theta}, \mathbf{x})$ and are defined on the infinite grids G_h and G_H .

Since the grid functions $\varphi(\boldsymbol{\theta}, \mathbf{x})$ are defined on the infinite grid G_h the influence of boundaries and boundary conditions is not taken into account. This is the main objective of LMA: to determine the quantitative convergence behavior and efficiency of an multigrid algorithm if a proper boundary treatment is included. In most cases this is a fair statement as additional suitable relaxations near the boundaries require an amount of work which is negligible for $h \rightarrow 0$.

Now, introduce a grid with fixed mesh size $\mathbf{h} = (h_1, h_2, h_3)$ and define $G_h = \{\mathbf{x} = \mathbf{k}\mathbf{h} := (k_1h_1, k_2h_2, k_3h_3), k \in \mathbb{Z}^3\}$. On G_h consider a discrete operator L_h corresponding to a difference stencil

$$L_{\mathbf{h}} \hat{=} [s_{\mathbf{k}}]_{\mathbf{h}}, \quad \mathbf{k} = (k_1, k_2, k_3) \in \mathbb{Z}^3$$

So,

$$L_{\mathbf{h}} w_{\mathbf{h}}(\mathbf{x}) = \sum_{\mathbf{k} \in V} s_{\mathbf{k}} w_{\mathbf{h}}(\mathbf{x} + \mathbf{k}\mathbf{h})$$

with constant coefficients $s_{\mathbf{k}} \in \mathbb{C}$, which will usually depend on \mathbf{h} . Here, V is a finite index set.

In 3D, the fundamental quantities in the LMA are the grid functions

$$\varphi(\boldsymbol{\theta}, \mathbf{x}) = e^{i\theta_1 x_1/h_1} e^{i\theta_2 x_2/h_2} e^{i\theta_3 x_3/h_3} \quad \text{for } \mathbf{x} \in G_h$$

Because $\varphi(\boldsymbol{\theta}, \mathbf{x}) = \varphi(\boldsymbol{\theta}', \mathbf{x})$ for $\mathbf{x} \in G_h$ if $\boldsymbol{\theta} = \boldsymbol{\theta}' \bmod(2\pi)$ it is sufficient to consider $\varphi(\boldsymbol{\theta}, \mathbf{x})$ with $\boldsymbol{\theta} \in [-\pi, \pi]^3$.

For $\boldsymbol{\theta} \in [-\pi, \pi]^3$ all grid functions $\varphi(\boldsymbol{\theta}, \mathbf{x})$ are (formal) eigenfunctions of any discrete operator with constant coefficients which can be described by a difference stencil. The relation

$$L_h \varphi(\boldsymbol{\theta}, \mathbf{x}) = \tilde{L}_h \varphi(\boldsymbol{\theta}, \mathbf{x}), \quad \mathbf{x} \in G_h$$

holds with,

$$\tilde{L}_h(\boldsymbol{\theta}) = \sum_{\mathbf{k} \in V} s_{\mathbf{k}} e^{i\boldsymbol{\theta} \mathbf{k}}$$

Where $\tilde{L}_h(\boldsymbol{\theta})$ is the symbol of L_h .

For the smoothing analysis the high and low frequency components on G_h with respect to G_H have to be distinguished. The definition is based on the fact that only those frequency components

$$\varphi(\boldsymbol{\theta}, \mathbf{x}) \quad \text{with } \boldsymbol{\theta} \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]^3$$

are distinguishable on G_H that is constructed with standard $\mathbf{H} = 2\mathbf{h}$ coarsening. For each $\boldsymbol{\theta}' \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]^3$, seven other frequency components $\varphi(\boldsymbol{\theta}, \mathbf{x})$ with $\boldsymbol{\theta} \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]^3$ coincide on G_H with $\varphi(\boldsymbol{\theta}', \mathbf{x})$ and are not distinguishable on G_H . This yields

$$\varphi \text{ low frequency component} \iff \boldsymbol{\theta} \in T^{low} := \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]^3$$

$$\varphi \text{ high frequency component} \iff \boldsymbol{\theta} \in T^{high} := [\pi, \pi]^3 \setminus \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]^3$$

Remember the Gauss-Seidel smoother described in section (2). Locally there are three unknowns, E_1 , E_2 and E_3 each lying on a different edge of a six point molecule (10). For simplicity the three unknowns are renamed here as u, v, w respectively. Assume that at the start of the smoothing process the errors in u, v and w are given in Fourier modes $(\theta_1, \theta_2, \theta_3)$ as follows

$$\begin{bmatrix} \delta u \\ \delta v \\ \delta w \end{bmatrix} = \begin{bmatrix} \delta u_0 \\ \delta v_0 \\ \delta w_0 \end{bmatrix} e^{i\boldsymbol{\theta} \mathbf{x}/\mathbf{h}} \quad (20)$$

where $\boldsymbol{\theta} \mathbf{x}/\mathbf{h} = \left(\frac{\theta_1 x}{h_1} + \frac{\theta_2 y}{h_2} + \frac{\theta_3 z}{h_3}\right)$ and $\delta u_0, \delta v_0, \delta w_0$ are the amplitudes. The Gauss-Seidel smoother updates each electric field component twice at a given mesh point. Because the unknowns are updated sequentially some of the unknowns are updated twice (δu_2), some once (δu_1) and others have not been

updated (δu_0). The aim is to determine the symbol \mathcal{M} which arises from a relation between the fully corrected values ($\delta u_2, \delta v_2, \delta w_2$) and the uncorrected values ($\delta u_0, \delta v_0, \delta w_0$). Hence,

$$\begin{bmatrix} \delta u_2 \\ \delta v_2 \\ \delta w_2 \end{bmatrix} = \mathcal{M} \begin{bmatrix} \delta u_0 \\ \delta v_0 \\ \delta w_0 \end{bmatrix}$$

The smoothing factor is then given by

$$\mu_{loc} = \sup_{\theta \in T^{high}} [\rho(\mathcal{M})]$$

where

$$\rho(\mathcal{M}) = \max_{i \in \dim(\mathcal{M})} |\lambda_i|, \quad \lambda_i = i\text{-th eigenvalue } \mathcal{M}$$

3.2.2 LMA of cell-block smoother in 3D

The description of the LMA of the cell-block and line smoother is quite similar. Therefore, in this section the analysis of the cell-block smoother will be carried out. In the next section the same procedure is repeated concerning the line smoother but then most details will be left out.

To carry out the local mode analysis for cell-block Gauss-Seidel, consider an arbitrary local section of the mesh with the electric field components lying on the edges of the grid.

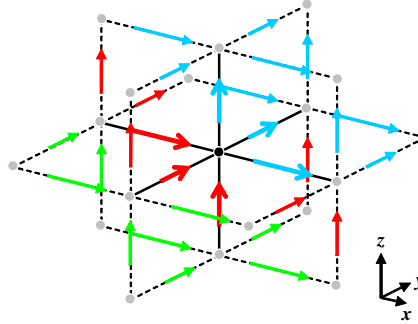


Figure 24: Values errors electric field components in fourier modes before applying GS cell-block smoothing step n .

In figures (24) and (25) a schematic representation is given of the values of the unknowns of the electric field components in Fourier modes (as in equation (20)) before and after applying one Gauss-Seidel cell-block smoothing step. The three different colors represent the states of the unknowns during the smoothing process. The green arrows represent the unknowns which are updated twice.

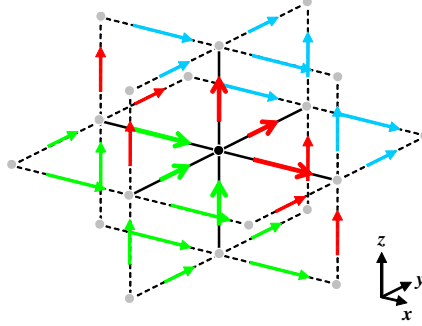


Figure 25: Values errors electric field components in fourier modes after applying GS cell-block smoothing step n

The red arrows represent the unknowns which are updated once and the blue arrows represent the unknowns which have not been updated yet.

As described in section (2.2.2), six errors surrounding one mesh point are updated simultaneously. In figures (24) and (25) these are the six arrows surrounding the black mesh point in the middle of the figures. The remaining 24 errors, which are needed to update the six errors lying around the black mesh point, are not updated and therefore will not change of value (color).

The discretization of the Maxwell equations as in equation (8) can be written as,

$$L_h \mathbf{e}_h = \mathbf{s}_h \quad (21)$$

where \mathbf{e} is the exact solution of equation (21), \mathbf{s} represents the source term of equation (8) and h is the spatial mesh size of the grid.

When applying a relaxation scheme, the exact solution will be approximated by $\mathbf{e}_h^{\tilde{m}}$ for the \tilde{m} -th iteration,

$$\mathbf{r}_h^{\tilde{m}} = \mathbf{s}_h - L_h \mathbf{e}_h^{\tilde{m}} \quad (22)$$

where $\mathbf{r}_h^{\tilde{m}}$ represents the residual.

The Gauss-Seidel cell-block smoother updates six electric field components lying around one node. For the remaining part of this section, the system of equations around grid node k, l, m will be examined. Hence, locally system (22) can be rewritten as

$$\hat{\mathbf{r}}_h^{\tilde{m}} = \hat{\mathbf{s}}_h - B \hat{\mathbf{e}}_{sur,h}^{\tilde{m}} - A \hat{\mathbf{e}}_h^{\tilde{m}} \quad (23)$$

where $\hat{\mathbf{r}}_h^{\tilde{m}}$ is the 6x1 residual vector, $\hat{\mathbf{s}}_h$ is the 6x1 vector and contains the source terms. $B \hat{\mathbf{e}}_{sur,h}^{\tilde{m}}$ is a matrix-vector multiplication where B contains the coefficients of the surrounding electric field components drawn as blue arrows in figure (10). The matrix A contains all the coefficients of the six unknown

electric field components drawn as red arrows in figure (10) and gathered in 6x1 vector

$$\hat{\mathbf{e}}_h^{\tilde{m}} = \left[E_{1,k-\frac{1}{2},l,m}, E_{1,k+\frac{1}{2},l,m}, E_{2,k,l-\frac{1}{2},m}, E_{2,k,l+\frac{1}{2},m}, E_{3,k,l,m-\frac{1}{2}}, E_{3,k,l,m+\frac{1}{2}} \right]^T$$

The Gauss-Seidel cell-block smoothing process consists of several steps.

In smoothing step \tilde{m} the approximation $\hat{\mathbf{e}}_h^{\tilde{m}}$ for $\hat{\mathbf{e}}_h$ around mesh node k, l, m was obtained. Applying the smoother again yields,

$$\hat{\mathbf{e}}_h^{\tilde{m}+1} = (1 - \omega_r)\hat{\mathbf{e}}_h^{\tilde{m}} + \omega_r\hat{\mathbf{e}}_h^* \quad (24)$$

where $\hat{\mathbf{e}}_h^{\tilde{m}+1}$ is the weighted sum of the previous approximation $\hat{\mathbf{e}}_h^{\tilde{m}}$ and the approximation for $\hat{\mathbf{e}}_h^{\tilde{m}+1}$ resulting from this smoothing step. Here $\hat{\mathbf{e}}_h^*$ is the new approximation for \mathbf{e}_h in smoothing step \tilde{m} ,

$$A\hat{\mathbf{e}}_h^* = \hat{\mathbf{s}}_h - B\hat{\mathbf{e}}_{sur,h}^{\tilde{m}},$$

hence,

$$\hat{\mathbf{e}}_h^* = A^{-1}\hat{\mathbf{d}}_h^{\tilde{m}}, \quad (25)$$

where

$$\hat{\mathbf{d}}_h^{\tilde{m}} = \hat{\mathbf{s}}_h - B\hat{\mathbf{e}}_{sur,h}^{\tilde{m}} = \hat{\mathbf{r}}_h^{\tilde{m}} + A\hat{\mathbf{e}}_h^{\tilde{m}}.$$

Now substituting the expression for $\hat{\mathbf{e}}_h^*$ (25) into (24) yields,

$$\hat{\mathbf{e}}_h^{\tilde{m}+1} = (1 - \omega_r)\hat{\mathbf{e}}_h^{\tilde{m}} + \omega_r A^{-1}\hat{\mathbf{d}}_h^{\tilde{m}}$$

Rewrite,

$$\begin{aligned} \hat{\mathbf{e}}_h^{\tilde{m}+1} &= (1 - \omega_r)\hat{\mathbf{e}}_h^{\tilde{m}} + \omega_r A^{-1}\hat{\mathbf{d}}_h^{\tilde{m}} \\ &= (1 - \omega_r)\hat{\mathbf{e}}_h^{\tilde{m}} + \omega_r A^{-1}(\hat{\mathbf{r}}_h^{\tilde{m}} + A\hat{\mathbf{e}}_h^{\tilde{m}}) \\ &= (1 - \omega_r)\hat{\mathbf{e}}_h^{\tilde{m}} + \omega_r A^{-1}\hat{\mathbf{r}}_h^{\tilde{m}} + \omega_r\hat{\mathbf{e}}_h^{\tilde{m}} \\ &= \hat{\mathbf{e}}_h^{\tilde{m}} + \omega_r A^{-1}\hat{\mathbf{r}}_h^{\tilde{m}} \end{aligned}$$

Thus,

$$A \left(\frac{\hat{\mathbf{e}}_h^{\tilde{m}+1} - \hat{\mathbf{e}}_h^{\tilde{m}}}{\omega_r} \right) = \hat{\mathbf{r}}_h^{\tilde{m}} \quad (26)$$

Rewrite system (26) in terms of errors. Subtract the local exact solution $\hat{\mathbf{e}}_h$ from (23),

$$\begin{aligned} \hat{\mathbf{r}}_h^{\bar{m}} &= \hat{\mathbf{s}}_h - B\hat{\mathbf{e}}_{sur,h}^{\bar{m}} - A\hat{\mathbf{e}}_h^{\bar{m}} \\ 0 &= \hat{\mathbf{s}}_h - B\hat{\mathbf{e}}_{sur,h} - A\hat{\mathbf{e}}_h \\ \hat{\mathbf{i}}_h^{\bar{m}} &= -B\delta\hat{\mathbf{e}}_{sur,h}^{\bar{m}} - A\delta\hat{\mathbf{e}}_h^{\bar{m}} \end{aligned} \quad (27)$$

where,

$$\begin{aligned} \hat{\mathbf{e}}_h^{\bar{m}} - \hat{\mathbf{e}}_h &= \delta\hat{\mathbf{e}}_h^{\bar{m}} \\ \hat{\mathbf{e}}_h^{\bar{m}+1} - \hat{\mathbf{e}}_h &= \delta\hat{\mathbf{e}}_h^{\bar{m}+1} \end{aligned} \quad (28)$$

Substitute (27) and (28) into (26),

$$A \left(\frac{\delta\hat{\mathbf{e}}_h^{\bar{m}+1} - \delta\hat{\mathbf{e}}_h^{\bar{m}}}{\omega_r} \right) = -B\delta\hat{\mathbf{e}}_{sur,h}^{\bar{m}} - A\delta\hat{\mathbf{e}}_h^{\bar{m}} \quad (29)$$

When the errors are expressed as Fourier modes according to section (3.2.1), system (29) can be rewritten. Only the first equation of (29) in Fourier modes is given here, the remaining five equations can be found in appendix (C.1).

$$\begin{aligned} & F e^{i(-\frac{\theta_1}{2})} \begin{bmatrix} +d_{1,k-\frac{1}{2},l,m} \frac{(\delta u_2 - \delta u_1)}{\omega_u} \\ -h^{-2} M_{k-\frac{1}{2},l+\frac{1}{2},m} e^{i(\frac{\theta_1}{2} + \frac{\theta_2}{2})} \frac{(\delta v_1 - \delta v_0)}{\omega_v} \\ +h^{-2} M_{k-\frac{1}{2},l-\frac{1}{2},m} e^{i(\frac{\theta_1}{2} - \frac{\theta_2}{2})} \frac{(\delta v_2 - \delta v_1)}{\omega_v} \\ -h^{-2} M_{k-\frac{1}{2},l,m+\frac{1}{2}} e^{i(\frac{\theta_1}{2} + \frac{\theta_3}{2})} \frac{(\delta w_1 - \delta w_0)}{\omega_w} \\ +h^{-2} M_{k-\frac{1}{2},l,m-\frac{1}{2}} e^{i(\frac{\theta_1}{2} - \frac{\theta_3}{2})} \frac{(\delta w_2 - \delta w_1)}{\omega_w} \end{bmatrix} \\ = & F e^{i(-\frac{\theta_1}{2})} \begin{bmatrix} h^{-2} \left(M_{k-\frac{1}{2},l+\frac{1}{2},m} e^{i(\theta_2)} + M_{k-\frac{1}{2},l,m+\frac{1}{2}} e^{i(\theta_3)} \right) \delta u_0 \\ +d_{1,k-\frac{1}{2},l,m} \delta u_1 \\ +h^{-2} \left(M_{k-\frac{1}{2},l-\frac{1}{2},m} e^{i(-\theta_2)} + M_{k-\frac{1}{2},l,m-\frac{1}{2}} e^{i(-\theta_3)} \right) \delta u_2 \\ -h^{-2} M_{k-\frac{1}{2},l+\frac{1}{2},m} e^{i(\frac{\theta_1}{2} + \frac{\theta_2}{2})} \delta v_0 \\ +h^{-2} \left(M_{k-\frac{1}{2},l-\frac{1}{2},m} e^{i(\frac{\theta_1}{2} - \frac{\theta_2}{2})} + M_{k-\frac{1}{2},l+\frac{1}{2},m} e^{i(-\frac{\theta_1}{2} + \frac{\theta_2}{2})} \right) \delta v_1 \\ -h^{-2} M_{k-\frac{1}{2},l-\frac{1}{2},m} e^{i(-\frac{\theta_1}{2} - \frac{\theta_2}{2})} \delta v_2 \\ -h^{-2} M_{k-\frac{1}{2},l,m+\frac{1}{2}} e^{i(\frac{\theta_1}{2} + \frac{\theta_3}{2})} \delta w_0 \\ +h^{-2} \left(M_{k-\frac{1}{2},l,m-\frac{1}{2}} e^{i(\frac{\theta_1}{2} - \frac{\theta_3}{2})} + M_{k-\frac{1}{2},l,m+\frac{1}{2}} e^{i(-\frac{\theta_1}{2} + \frac{\theta_3}{2})} \right) \delta w_1 \\ -h^{-2} M_{k-\frac{1}{2},l,m-\frac{1}{2}} e^{i(-\frac{\theta_1}{2} - \frac{\theta_3}{2})} \delta w_2 \end{bmatrix} \end{aligned}$$

where,

$$\begin{aligned} F &= e^{i(k\theta_1 + l\theta_2 + m\theta_3)} \\ d_{1,k-\frac{1}{2},l,m} &= S_{k-\frac{1}{2},l,m} - h^{-2} \left(M_{k-\frac{1}{2},l+\frac{1}{2},m} + M_{k-\frac{1}{2},l-\frac{1}{2},m} + M_{k-\frac{1}{2},l,m+\frac{1}{2}} + M_{k-\frac{1}{2},l,m-\frac{1}{2}} \right) \\ h^{-2} &= \left(e_{k\pm\frac{1}{2}}^x \right)^2 = \left(e_{l\pm\frac{1}{2}}^y \right)^2 = \left(e_{m\pm\frac{1}{2}}^z \right)^2, \quad \text{equidistant grid} \end{aligned}$$

with $S_{k\pm\frac{1}{2},l\pm\frac{1}{2},m\pm\frac{1}{2}}, M_{k\pm\frac{1}{2},l\pm\frac{1}{2},m\pm\frac{1}{2}}$ and $e_{k\pm\frac{1}{2}}^x, e_{l\pm\frac{1}{2}}^y, e_{m\pm\frac{1}{2}}^z$ from equations (5), (4) and (9) respectively.

After some manipulation of system (29) in Fourier modes, this yields a relation between the corrected and uncorrected values as

$$P \begin{bmatrix} \delta u_1 \\ \delta v_1 \\ \delta w_1 \\ \delta u_2 \\ \delta v_2 \\ \delta w_2 \end{bmatrix} = Q \begin{bmatrix} \delta u_0 \\ \delta v_0 \\ \delta w_0 \end{bmatrix} \quad (30)$$

where P, Q are complex matrices of order 6x6 and 6x3, respectively. The coefficients of the matrices cannot be found explicitly in this thesis but can be found in the implementation of the local mode analysis in Matlab in appendix (G).

Remember that the aim of the local mode analysis was to find the relation between the final corrected values ($\delta u_2, \delta v_2, \delta w_2$) and the uncorrected values ($\delta u_0, \delta v_0, \delta w_0$). The system of equations (30) can be written as

$$P^{-1}Q = \begin{bmatrix} \mathcal{N} \\ -.- \\ \mathcal{M} \end{bmatrix}, \quad (31)$$

where \mathcal{N}, \mathcal{M} are 3x3 complex matrices. The symbol \mathcal{M} is now easily obtained and de smoothing factor can be calculated.

3.2.3 LMA of line smoother in 3D

In this section the governing equations of the local mode analysis for a line smoother in x -direction are given. The local mode analysis of the line smoother follows the same procedure as in the previous section. The resulting system of equations yields a relation between the fully corrected and uncorrected values of the electric field components.

In figures (26) and (27) a schematic representation is given of the values of the unknowns of the electric field components in Fourier modes (as in equation (20)) before and after applying one Gauss-Seidel line smoothing step. The three different colors represent the states of the unknowns during the smoothing process. The green arrows represent the unknowns that are updated twice. The red arrows represent the unknowns that are updated once and the blue arrows represent the unknowns that have not been updated yet.

The main difference between figures (26) and (27) and figures (24) and (25) representing the cell-block smoother introduced in previous section, is the number of unknowns that are updated after one smoothing step. The cell-block smoother updates six unknowns lying around the black mesh point simultaneously. The line smoother updates fourteen unknowns surrounding the nodes that lie on the line in x -direction. Note that before applying the line smoother

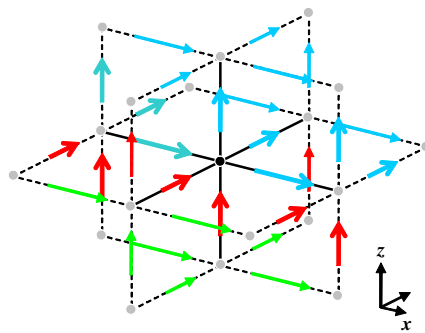


Figure 26: Values errors electric field components in fourier modes before applying GS line smoothing step n .

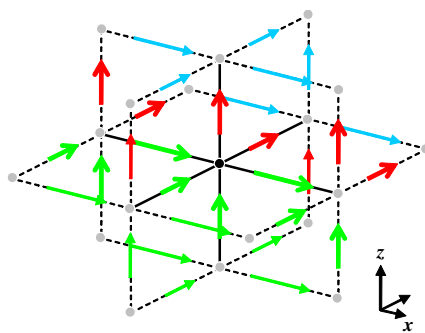


Figure 27: Values errors electric field components in fourier modes after applying GS line smoothing step n .

the unknowns that lie on the x -line, which intersects the black mesh point k, l, m , were not yet updated. However, when the line smoother is applied the unknowns are directly updated. This is a major difference between the cell-block smoother and the line smoother.

Apply the same procedure as in previous section. System (29) can be written out in Fourier modes,

$$\begin{aligned}
& F e^{i(-\frac{\theta_1}{2})} \left[\begin{array}{l} -d_{1,k-\frac{1}{2},l,m} \frac{(\delta u_2 - \delta u_0)}{\omega_u} \\ +h^{-2} \left(M_{k-\frac{1}{2},l+\frac{1}{2},m} e^{i(\frac{\theta_1}{2} + \frac{\theta_2}{2})} - M_{k-\frac{1}{2},l+\frac{1}{2},m} e^{i(-\frac{\theta_1}{2} + \frac{\theta_2}{2})} \right) \frac{(\delta v_1 - \delta v_0)}{\omega_v} \\ +h^{-2} \left(M_{k-\frac{1}{2},l-\frac{1}{2},m} e^{i(-\frac{\theta_1}{2} - \frac{\theta_2}{2})} - M_{k-\frac{1}{2},l-\frac{1}{2},m} e^{i(\frac{\theta_1}{2} - \frac{\theta_2}{2})} \right) \frac{(\delta v_2 - \delta v_1)}{\omega_v} \\ +h^{-2} \left(M_{k-\frac{1}{2},l,m+\frac{1}{2}} e^{i(\frac{\theta_1}{2} + \frac{\theta_3}{2})} - M_{k-\frac{1}{2},l,m+\frac{1}{2}} e^{i(-\frac{\theta_1}{2} + \frac{\theta_3}{2})} \right) \frac{(\delta w_1 - \delta w_0)}{\omega_w} \\ +h^{-2} \left(M_{k-\frac{1}{2},l,m-\frac{1}{2}} e^{i(-\frac{\theta_1}{2} - \frac{\theta_3}{2})} - M_{k-\frac{1}{2},l,m-\frac{1}{2}} e^{i(\frac{\theta_1}{2} - \frac{\theta_3}{2})} \right) \frac{(\delta w_2 - \delta w_1)}{\omega_w} \end{array} \right] \\
= & F e^{i(-\frac{\theta_1}{2})} \left[\begin{array}{l} h^{-2} \left(M_{k-\frac{1}{2},l+\frac{1}{2},m} e^{i(\theta_2)} + M_{k-\frac{1}{2},l,m+\frac{1}{2}} e^{i(\theta_3)} + h^2 d_{1,k-\frac{1}{2},l,m} \right) \delta u_0 \\ +h^{-2} \left(M_{k-\frac{1}{2},l-\frac{1}{2},m} e^{i(-\theta_2)} + M_{k-\frac{1}{2},l,m-\frac{1}{2}} e^{i(-\theta_3)} \right) \delta u_2 \\ +h^{-2} \left(M_{k-\frac{1}{2},l+\frac{1}{2},m} e^{i(-\frac{\theta_1}{2} + \frac{\theta_2}{2})} - M_{k-\frac{1}{2},l+\frac{1}{2},m} e^{i(\frac{\theta_1}{2} + \frac{\theta_2}{2})} \right) \delta v_0 \\ +h^{-2} \left(M_{k-\frac{1}{2},l-\frac{1}{2},m} e^{i(\frac{\theta_1}{2} - \frac{\theta_2}{2})} - M_{k-\frac{1}{2},l-\frac{1}{2},m} e^{i(-\frac{\theta_1}{2} - \frac{\theta_2}{2})} \right) \delta v_1 \\ +h^{-2} \left(M_{k-\frac{1}{2},l,m+\frac{1}{2}} e^{i(-\frac{\theta_1}{2} + \frac{\theta_3}{2})} - M_{k-\frac{1}{2},l,m+\frac{1}{2}} e^{i(\frac{\theta_1}{2} + \frac{\theta_3}{2})} \right) \delta w_0 \\ +h^{-2} \left(M_{k-\frac{1}{2},l,m-\frac{1}{2}} e^{i(\frac{\theta_1}{2} - \frac{\theta_3}{2})} - M_{k-\frac{1}{2},l,m-\frac{1}{2}} e^{i(-\frac{\theta_1}{2} - \frac{\theta_3}{2})} \right) \delta w_1 \end{array} \right]
\end{aligned}$$

The five remaining equations can be found in appendix (C.2).

In next section the line and block smoother are compared for the eigenvalue test problem of section (2.3.1).

3.2.4 Results LMA

This section contains a brief overview of the local mode analysis of the cell-block and line smoother. The eigenvalue test problem of section (2.3.1) is used to carry out the analysis. The Matlab code of this test problem can be found in appendix (G). The equations resulting from the local mode analysis are captured by system (30) and represented by matrices P and Q . The code in schematic form:

```

input matrix A and B
loop over all theta out of [-pi,-pi/2] and [pi/2,pi]

    define coefficients P,Q
    obtain M from P^-1*Q

    add max(abs(eig(M))) to vector mu
end loop
output max(mu)

```

For the eigenvalue test problem without stretching, for different values of the relaxation parameter ω and grid sizes (and thus h), the following results were obtained

ω	$N = 64$		32		16		8		4	
	Line	CB smthr								
1	0.20	0.25	0.20	0.25	0.21	0.28	0.21	0.77	0.17	0.35
0.9	0.19	0.26	0.19	0.26	0.19	0.27	0.20	0.55	0.17	0.36
0.8	0.19	0.27	0.19	0.27	0.19	0.27	0.19	0.47	0.16	0.34
0.7	0.24	0.27	0.24	0.27	0.24	0.37	0.22	0.49	0.17	0.31

Table 4: Smoothing factor of Gauss-Seidel line and cell-block smoother, where $\omega = \omega_u = \omega_v = \omega_w$ and N is the number of grid points in each direction.

Note that $N = 128$ could not be analyzed due to limited computer memory capacity. From table 4 it is shown that the smoothing factor of the line smoother on a non-stretched, uniform grid is somewhat better than the smoothing factor of the cell-block smoother. Changing the relaxation parameter is not useful for the line smoother. For coarse grids ($N < 8$) the smoothing constant increases, therefore the smoothing ability decreases. The main conclusion from this analysis is that the line smoother is overall a better smoother, at some grids the smoothing constant is a factor 2 smaller.

Due to the complex discretization of the curl-curl operator resulting from the Maxwell equations the analysis of the two-grid convergence factors and error reduction factors is beyond the scope of this thesis.

4 Numerical experiments

In this section three different test problems are considered originating from [12]. The first problem has already been presented in section (2.3.1), a short summary will be given here. The second problem resembles the first problem but the permittivity is chosen variable in the domain. Both test problems are artificial and are based on exact solutions with eigenfunctions. The third example is a realistic marine example with a variation on the power-law grid stretching (hyperbolic cosine grid stretching, see appendix (E)).

The iterations were terminated once the norm of the residual decreased to 10^{-8} times the value for a zero electric field. The computations were carried out on a 64-bit AMD Opteron 238 machine running Linux. The algorithm was coded in Matlab, with the smoother and residual evaluation written in C¹. The listed computational times are therefore not optimal but nevertheless, they provide some indication of the performance.

In section (3.1.3) two modifications of the multigrid solver were introduced: semi-coarsening and line relaxation. Together with the original multigrid method with the Symmetric cell-block Gauss-Seidel smoother and standard coarsening, four methods will be compared here:

1. bi,SGS: The original scheme of section (2) employs post-smoothing with Symmetric cell-block Gauss-Seidel, F-cycles and `bicgstab2`.
2. bi,SLGS3: Pre- and post-smoothing with Symmetric Line Gauss-Seidel in three directions, F-cycles and `bicgstab2` (section (3.1)).
3. SC,SLGS2: Pre- and post-smoothing with Symmetric Line Gauss-Seidel in two directions, semi-coarsening in these two directions and F-cycles. The direction that is not coarsened changes after each full cycle and remains the same within the cycle (section (3.1)).
4. bi,SLGS2: same as method 3 but with `bicgstab2`

Note that the original Symmetric cell-block Gauss-Seidel and Symmetric Line Gauss-Seidel multigrid methods without `bicgstab2` are not taken into account during testing. Previous test results (section (2.3) and [12]) show that a combination with `bicgstab2` convergences faster and at worst in an equal number of steps compared to the multigrid solvers without `bicgstab2`. Therefore, the first two methods '1:bi,SGS ' and '2:bi,SLGS3 ' use multigrid in combination with the `bicgstab2`. The third method '3:SC,SLGS2 ' does not use `bicgstab2`. It is add to method 3, resulting in method 4. Because `bicgstab2` requires a fixed preconditioner for every iteration 3 alternating semicoarsening multigrid cycles are carried out.

¹MEX files and runs provided by W.A. Mulder, Shell

4.1 Variable conductivity

This section contains a summarizes the test problem of section (2.3.1). The first artificial test problem is based on eigenfunctions on the domain $[0, 2\pi]^3 m^3$. The conductivity is $\sigma = 10 + c(x, y, z)$, with $c(x, y, z) = (x + 1)(y + 2)(z - \pi)^2$ for $z < \pi$ and zero elsewhere, $\epsilon_r = 0$ and $\mu_r = 1$. The electric field components are $E_1 = -2 \cos(x) \sin(y) \sin(z)$, $E_2 = -2 \sin(x) \cos(y) \sin(z)$ and $E_3 = \sin(x) \sin(y) \cos(z)$ V/m. The angular frequency is $\omega = 10^6 \text{Hz}$. The current source term follows from these expressions and equation (3).

Table 5 lists convergence results for grids with $N \times N \times N$ cells and different amounts of power-law grid stretching (section (2.3.1)).

N	α	1:bi,SGS		2:bi,SLGS3		3:SC,SLGS2		4:bi,SC,SLGS2	
		iter	cpu (s)	iter	cpu (s)	iter	cpu (s)	iter	cpu (s)
32	0	7	$6.4 \cdot 10^0$	3	$3.7 \cdot 10^1$	4	$4.2 \cdot 10^1$	6	$5.9 \cdot 10^1$
64		7	$3.6 \cdot 10^1$	4	$4.3 \cdot 10^2$	4	$3.5 \cdot 10^2$	6	$5.3 \cdot 10^2$
128		6	$2.6 \cdot 10^2$	4	$5.3 \cdot 10^3$	8	$8.1 \cdot 10^3$	9	$9.2 \cdot 10^3$
32	0.04	8	$7.6 \cdot 10^0$	4	$5.1 \cdot 10^1$	4	$4.3 \cdot 10^1$	6	$6.3 \cdot 10^1$
64		14	$7.8 \cdot 10^1$	4	$4.3 \cdot 10^2$	5	$4.5 \cdot 10^2$	6	$5.4 \cdot 10^2$
128		35	$1.5 \cdot 10^3$	8	$1.1 \cdot 10^4$	7	$7.2 \cdot 10^3$	9	$9.4 \cdot 10^3$
32	0.06	10	$9.6 \cdot 10^0$	4	$5.1 \cdot 10^1$	4	$4.1 \cdot 10^1$	6	$6.3 \cdot 10^1$
64		21	$1.2 \cdot 10^2$	5	$5.5 \cdot 10^2$	5	$4.5 \cdot 10^2$	6	$5.4 \cdot 10^2$
128		69	$2.9 \cdot 10^3$	11	$1.5 \cdot 10^4$	7	$7.3 \cdot 10^3$	9	$9.5 \cdot 10^3$
32	0.1	16	$9.6 \cdot 10^0$	—	—	4	$3.5 \cdot 10^1$	6	$6.4 \cdot 10^1$
64		41	$1.2 \cdot 10^2$	—	—	6	$5.4 \cdot 10^2$	6	$5.4 \cdot 10^2$
128		233	$9.9 \cdot 10^3$	—	—	6	$6.5 \cdot 10^3$	6	$6.5 \cdot 10^3$

Table 5: Iteration counts and cpu-times in seconds for the first test problem with variable conductivity

The grid-independent convergence rates on equidistant grids are only observed for the methods 1 and 2 and perhaps 4. When grid stretching is applied, the multigrid solver with semi-coarsening and line smoothing, methods 3 and 4, shows the best performance in terms of iteration count. They appear to be insensitive to the amount of stretching when considering a fixed number of cells. The original cell-block smoother and the line smoother without semi-coarsening but with `bicgstab2`, method 1 and 2, show a performance leak when grid stretching is applied. Especially on the finest grids with $N = 128$ the number of multigrid iterations increases heavily.

Though two new methods (3 and 4) were introduced, the amount of cpu-time needed to evaluate the multigrid schemes still remains high. As the number of iterations remains low for methods 3 and 4 when grid stretching is applied, the multigrid solver using semi-coarsening and line smoothing in two directions nevertheless needs three times more computation time than the original multigrid solver in combination with `bicgstab2` for $\alpha = 0.06$. Even when the number of iterations between methods 1 and 4 differ with a factor 10 to 20. This is a good

example of the amount of work needed for semi-coarsening and line smoothing. Only for strong stretching with $\alpha = 0.1$ method 3 and 4 beat the original method on the finest grid with $N = 128$ in terms of iteration count and cpu-time.

The two adapted multigrid solvers, methods 3 and 4, perform remarkably better in terms of iteration counts compared to the original multigrid solver. Furthermore, the multigrid solver using semi-coarsening and line smoothing performs significantly better than the multigrid solver using standard coarsening and line smoothing, because the latter is cpu-time consuming and does not show good convergence rates when stretching is applied.

4.2 Variable permittivity

The same domain is considered as before, with conductivity $\sigma = 1$ S/m and $\epsilon_r = 0$. The permittivity is piecewise constant with $\mu_r(z) = \mu_r^L$ for $z < \pi$ and $\mu_r(z) = \mu_r^R$ for $z > \pi$. The source terms are set to

$$\begin{aligned} J_1 &= \cos(x) \sin(y) \sin(z) \left[1 - \frac{3}{i\omega\mu_0\mu_r(z)} \right] \\ J_2 &= \sin(x) \cos(y) \sin(z) \left[1 - \frac{3}{i\omega\mu_0\mu_r(z)} \right] \\ J_3 &= -2 \sin(x) \sin(y) \cos(z) \left[1 - \frac{3}{i\omega\mu_0\mu_r(z)} \right] \end{aligned}$$

Hence, the solution is

$$\begin{aligned} E_1 &= -\cos(x) \sin(y) [\sin(z) + \gamma g_1(z)] \\ E_2 &= -\sin(x) \cos(y) [\sin(z) + \gamma g_1(z)] \\ E_3 &= 2 \sin(x) \sin(y) [\cos(z) + \gamma g_3(z)] \end{aligned}$$

Let $q^{L,R} = \sqrt{2 - i\omega\mu_0\mu_r^{L,R}}$ then

$$\begin{aligned} g_1^L(z) &= 2 \sinh(q^L z) \\ g_3^L(z) &= -\frac{2}{q^L} \cosh(q^L z) \end{aligned}$$

for $z < \pi$, whereas for $z > \pi$,

$$\begin{aligned} g_1^R(z) &= 2a \sinh(q^R(2\pi - z)) \\ g_3^R(z) &= \frac{2a}{q^R} \cosh(q^R(2\pi - z)) \end{aligned}$$

Here

$$a = \frac{\sinh(q^L \pi)}{\sinh(q^R \pi)}$$

$$\gamma = \frac{3}{2i\omega\mu_0} \left(\frac{1}{\mu_r^R} - \frac{1}{\mu_r^L} \right) / \left(\frac{\cosh(q^L \pi)}{q^L} + a \frac{\cosh(q^R \pi)}{q^R} \right)$$

Note that $g_1(z)$ is continuous across $z = \pi$, with a discontinuous derivative and $g_3(z)$ is discontinuous across $z = \pi$.

In table 6 the test results are presented with $\mu_r^L = 1$ and $\mu_r^R = 2$.

N	α	1:bi,SGS		2:bi,SLGS3		3:SC,SLGS2		4:bi,SC,SLGS2	
		iter	cpu (s)	iter	cpu (s)	iter	cpu (s)	iter	cpu (s)
32	0	7	$6.5 \cdot 10^0$	4	$5.1 \cdot 10^1$	5	$5.2 \cdot 10^1$	6	$6.3 \cdot 10^1$
64		7	$3.6 \cdot 10^1$	4	$4.3 \cdot 10^2$	9	$8.1 \cdot 10^2$	9	$8.1 \cdot 10^2$
128		6	$2.5 \cdot 10^2$	4	$5.4 \cdot 10^3$	19	$2.0 \cdot 10^4$	15	$1.6 \cdot 10^4$
32	0.04	10	$9.6 \cdot 10^0$	4	$5.1 \cdot 10^1$	5	$5.4 \cdot 10^1$	6	$6.2 \cdot 10^1$
64		16	$9.1 \cdot 10^1$	4	$4.3 \cdot 10^2$	9	$8.2 \cdot 10^2$	9	$8.1 \cdot 10^2$
128		38	$1.6 \cdot 10^3$	9	$1.2 \cdot 10^4$	15	$1.6 \cdot 10^4$	15	$1.6 \cdot 10^4$
32	0.06	12	$1.1 \cdot 10^0$	4	$5.1 \cdot 10^1$	6	$6.3 \cdot 10^1$	6	$6.4 \cdot 10^1$
64		25	$1.4 \cdot 10^2$	6	$6.6 \cdot 10^2$	9	$8.1 \cdot 10^2$	9	$8.1 \cdot 10^2$
128		86	$3.7 \cdot 10^3$	16	$2.2 \cdot 10^4$	14	$1.5 \cdot 10^4$	15	$1.6 \cdot 10^4$
32	0.1	19	$1.1 \cdot 10^1$	—	—	6	$6.3 \cdot 10^1$	6	$6.4 \cdot 10^1$
64		55	$1.4 \cdot 10^2$	—	—	9	$8.1 \cdot 10^2$	9	$8.3 \cdot 10^2$
128		325	$1.4 \cdot 10^4$	—	—	13	$1.5 \cdot 10^4$	12	$1.3 \cdot 10^4$

Table 6: Iteration counts and cpu-times in seconds for the second test problem with variable permittivity and a discontinuous solution

As in test problem (4.1) multigrid convergence rates seem to deteriorate when grid stretching is applied. Unfortunately, the multigrid solver using semi-coarsening and line smoothing is not performing so well either. Where convergence rates were preserved earlier, in this test problem the number of iterations increases for every method. Only method 4 seems to perform well in terms of iteration counts and again seems to be insensitive to the amount of stretching for a fixed number of cells. However, compared to the original method computation times for the adapted multigrid solvers remain large.

As the multigrid iteration process stops once the norm of the residual decreases to 10^{-8} times its value for a zero electric field the expected number of iterations will be 8 [17]. Therefore, neither of these methods satisfies the criterion of a robust multigrid solver except method 4. Also, note that the number of iterations increases heavily on finer grids as observed before in test problem (4.1). Maybe the use of semicoarsening can introduce artificial anisotropy into the discrete equations on coarser grids [17]. However, test results on the coarse grids with $N < 128$ show that this is probably not the case. Further investigation has to be done here.

4.3 Marine example

A more realistic subsurface model was constructed by modifying the SEG/EAGE salt model [2]. This model was designed for simulating seismic wave propagation and contains a complex salt body surrounded by sediments. The sea water has depths around 120m. Its sediments are 13500 by 13480 by 4680m. Because CSEM requires a deeper sea than provided by this model, the water depth was increased with 500m by moving down the model. Next, velocities were replaced by conductivities. The water velocity of 1500 m/s was replaced by $\frac{1}{0.3}$ S/m. Velocities above 4000 m/s, indicative of salt, were replaced by $\frac{1}{30}$ S/m. Below 3956 m, basement was set to 500 S/m. The conductivity of the sediments was determined by $\sigma_{sed} = (v/1700)^{-3.88}$ S/m, with the velocity v in m/s. This choice was motivated by [9]. For air, σ was set to 10^{-10} S/m. The conductivity was mapped to a new grid with hyperbolic cosine stretching (appendix (E)). The horizontal dimensions were the same as for the original model, whereas z ranged from -5000 m to 5000 m. A total of 31 receivers were positioned at the sea-bottom between $x = 4000$ m and 10000 m at 200 m spacing with $y = 0$ m. Depths ranged between 605 and 625 m. The model is displayed in figure 28.

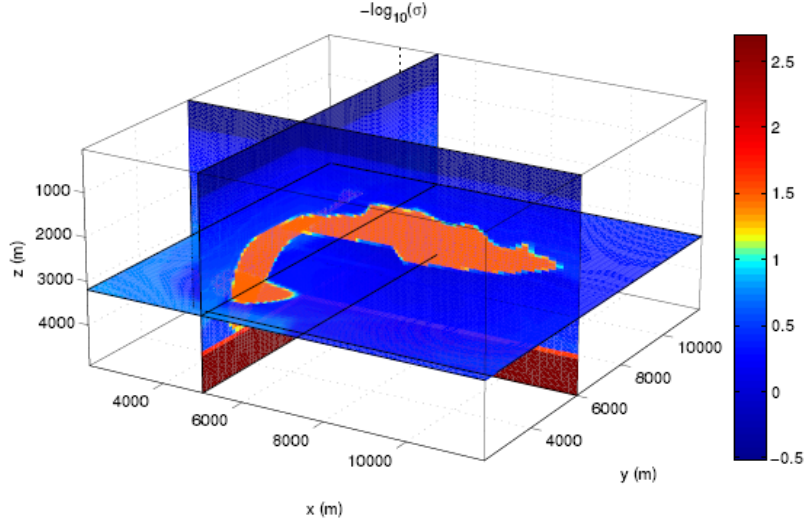


Figure 28: Model for a marine example with a resistive salt body ($h_{\min} = 50$ m). The sea bottom has a depth of about 600m.

For its discrete representation, the exact cell averages are computed. The frequency is $\omega/2\pi = 5$ Hz. A finite-length current source with normalized strength was placed in the interval between $(x, y, z) = (6400, 6500, 500)$ and $(x, y, z) = (6600, 6500, 500)$, all in meters.

The following table 7 lists the results for a grid with 128^3 cells. Note that hy-

perbolic cosine stretching is used instead of power-law grid stretching (appendix (E)), using a smallest cell width h_{\min} . The largest ratio between neighboring cells is given by $1 + \alpha$.

$N = 128$		1:bi,SGS		2:bi,SLGS3		3:SC,SLGS2		4:bi,SC,SLGS2	
h_{\min}	α	iter	cpu (s)	iter	cpu (s)	iter	cpu (s)	iter	cpu (s)
50	0.035	15	$6.4 \cdot 10^2$	5	$7.1 \cdot 10^3$	14	$1.5 \cdot 10^4$	12	$1.3 \cdot 10^4$
20	0.059	75	$3.3 \cdot 10^3$	14	$2.0 \cdot 10^4$	14	$1.5 \cdot 10^4$	12	$1.3 \cdot 10^4$
10	0.074	163	$7.1 \cdot 10^3$	26	$3.7 \cdot 10^4$	15	$1.6 \cdot 10^4$	15	$1.7 \cdot 10^4$
5	0.089	409	$1.8 \cdot 10^4$	44	$6.3 \cdot 10^4$	21	$2.3 \cdot 10^4$	18	$2.0 \cdot 10^4$

Table 7: Iteration counts and cpu-times in seconds for the realistic marine example

In this last test problem the same pattern can be observed as for the previous two examples. On coarse grids, where h_{\min} is of order 10^1 and α between 0.035 and 0.074 the convergence rates are only stable for the multigrid solver with semicoarsening and line smoothing. The original solver and the second adapted multigrid variant, standard coarsening and line smoothing, perform not well. Only method 3 and 4, the multigrid solver with semicoarsening and line smoother shows relative consistency in terms of iteration count. However, when h_{\min} becomes smaller multigrid convergence rates deteriorate for all methods and the original method still beats new methods 3 and 4 in terms of cpu-time. Apparently fine grids seem to affect the anisotropy in the discretization of the Maxwell equations. Furthermore, the two appliances for solving anisotropy due to grid stretching are not able to solve the problem for fine grids. In spite of the improvements in decreasing the number of iterations, further investigation has to be done in order to regain grid-independent convergence rates.

5 Conclusion

A multigrid method for electromagnetic diffusion was developed by W.A. Mulder of Shell International Exploration and Production. The aim of EM diffusion is to give some complementary information on the location of a source. The governing equations of EM diffusion are the Maxwell equations. They were discretized on tensor-product Cartesian grids with the finite integration technique (FIT) introduced by Weiland [5]. The scheme can be viewed as a finite-volume generalization of Yee's scheme [24]. The system of equations resulting from the discretization were solved by a multigrid method. In Feigh *et al* [6] a multigrid method for the FIT discretization is presented and Mulder used this approach for the components of the multigrid solver. Four multigrid components have to be defined which directly affect the rate of convergence of the multigrid solver: the coarse grid operator, the smoother, the restriction and prolongation operators. The original solver uses standard 3D coarsening, combining 2x2x2 fine-grid cells into a single coarse-grid cell. The restriction operator was based on volume averaging, the prolongation operator on piecewise constant interpolation and the Symmetric cell-block Gauss-Seidel (SGS) smoother.

In [12] several test problems were evaluated using this multigrid solver. The method showed textbook convergence rates on grids with constant spacing. When stretching is applied the multigrid convergence rates deteriorates for grids with a large number of cells in each direction. This is a well known problem when multigrid is applied to stretched grids. Because of the stretching anisotropy is introduced into the discretizations. The multigrid convergence remedies include line relaxation and semi-coarsening [17].

The smoothing factors of the cell-block Gauss-Seidel smoother and a new Line Gauss-Seidel smoother were analyzed with a Local Mode Analysis (LMA) [17]. The analysis of a cell-block and line smoother used in a multigrid solver for solving the Navier-Stokes equations [15] was used as a point of departure. LMA was only carried out on equidistant grids.

Both line smoothing and semicoarsening were used to adapt the original solver resulting in two new multigrid schemes. The first scheme uses standard coarsening and pre- and post-smoothing with Symmetric Line Gauss-Seidel in three directions. The second scheme uses semicoarsening in two directions and pre- and post-smoothing by Symmetric Line Gauss-Seidel in these two directions. The direction that is not coarsened changes after each full cycle and remains the same within the cycle. All three methods, the original method and the two adapted versions, are also combined with `bicgstab2` [19] where multigrid acts as a preconditioner.

Three test problems from [12] were evaluated comparing the original and the adapted multigrid solvers. The first two problems are unphysical problems based on eigenvalues but with variable conductivity and permittivity. The third test problem is a more realistic subsurface model constructed by modifying the SEG/EAGE salt model [2].

The goal of this investigation was to improve the original multigrid solver developed by W.A. Mulder in [12] in order to regain grid-independent convergence

rates, small numbers of iterations and reduce cpu-times when grid stretching is applied. All three test problems showed the same pattern. The number of iterations was reduced significantly in some cases. The multigrid method with semi-coarsening and line smoothing in combination with `bicgstab2` performed best in terms of iteration counts. However, the cpu-times became significantly higher than the original method except for one of two cases. The method with only line smoothing in three directions and no semicoarsening did not perform well for any test problem and is therefore not a reasonable alternative for the original multigrid solver.

The multigrid solver with semi-coarsening and line smoothing in combination with `bicgstab2` is the most robust solver and therefore the main objective of this thesis has been reached. Moreover, further investigation in reducing the computation times is needed which can make this solver significantly faster. The special structure, complex and symmetric, of the matrix of the line smoother can be exploited here.

Finally, none of the adapted multigrid solvers satisfies the requirement that it is insensitive to grid stretching *and* shows grid-independent convergence rates. Therefore other suggestions to improve multigrid solvers are listed below which have been made earlier in [12]:

- Other choices of restriction and prolongation operators as only a few have been investigated here. The adverse effects of strongly varying coefficients can be reduced by using operator-weighted restriction and prolongation operators.
- Different ways of defining coarser grids. For instance by not coarsening cells if they have large widths relative to the smallest cell [14], or by using coarser grids with nodes that do not coincide with the finer grid [6].
- The use of local grid refinement on equidistant grids. This should give very good convergence rates while still allowing for finer grids where necessary, at the expense of increased code complexity.

$$\begin{aligned}
& e_{l+\frac{1}{2}}^y M_{k,l+\frac{1}{2},m-\frac{1}{2}} \left[e_{l+\frac{1}{2}}^y \left[E_{3,k,l+1,m-\frac{1}{2}} - E_{3,k,l,m-\frac{1}{2}} \right] - e_{m-\frac{1}{2}}^z \left[E_{2,k,l+\frac{1}{2},m} - E_{2,k,l+\frac{1}{2},m-1} \right] \right] + \\
& \dots \\
& e_{l-\frac{1}{2}}^y M_{k,l-\frac{1}{2},m-\frac{1}{2}} \left[e_{l-\frac{1}{2}}^y \left[E_{3,k,l,m-\frac{1}{2}} - E_{3,k,l-1,m-\frac{1}{2}} \right] - e_{m-\frac{1}{2}}^z \left[E_{2,k,l-\frac{1}{2},m} - E_{2,k,l-\frac{1}{2},m-1} \right] \right]. \\
\\
& r_{3,k,l,m+\frac{1}{2}} = s_{3,k,l,m+\frac{1}{2}} + S_{k,l,m+\frac{1}{2}} E_{3,k,l,m+\frac{1}{2}} - \dots \\
& e_{k+\frac{1}{2}}^x M_{k+\frac{1}{2},l,m+\frac{1}{2}} \left[e_{m+\frac{1}{2}}^z \left[E_{1,k+\frac{1}{2},l,m+1} - E_{1,k+\frac{1}{2},l,m} \right] - e_{k+\frac{1}{2}}^x \left[E_{3,k+1,l,m+\frac{1}{2}} - E_{3,k,l,m+\frac{1}{2}} \right] \right] + \\
& \dots \\
& e_{k-\frac{1}{2}}^x M_{k-\frac{1}{2},l,m+\frac{1}{2}} \left[e_{m+\frac{1}{2}}^z \left[E_{1,k-\frac{1}{2},l,m+1} - E_{1,k-\frac{1}{2},l,m} \right] - e_{k-\frac{1}{2}}^x \left[E_{3,k,l,m+\frac{1}{2}} - E_{3,k-1,l,m+\frac{1}{2}} \right] \right] - \\
& \dots \\
& e_{l+\frac{1}{2}}^y M_{k,l+\frac{1}{2},m+\frac{1}{2}} \left[e_{l+\frac{1}{2}}^y \left[E_{3,k,l+1,m+\frac{1}{2}} - E_{3,k,l,m+\frac{1}{2}} \right] - e_{m+\frac{1}{2}}^z \left[E_{2,k,l+\frac{1}{2},m+1} - E_{2,k,l+\frac{1}{2},m} \right] \right] + \\
& \dots \\
& e_{l-\frac{1}{2}}^y M_{k,l-\frac{1}{2},m+\frac{1}{2}} \left[e_{l-\frac{1}{2}}^y \left[E_{3,k,l,m+\frac{1}{2}} - E_{3,k,l-1,m+\frac{1}{2}} \right] - e_{m+\frac{1}{2}}^z \left[E_{2,k,l-\frac{1}{2},m+1} - E_{2,k,l-\frac{1}{2},m} \right] \right].
\end{aligned}$$

B Line smoother

System of equations of the x -direction line smoother in matrix notation,

$$\hat{A}\hat{\mathbf{E}}_{l,m} = s - \hat{B}\hat{\mathbf{E}}_{sur} \quad (32)$$

where,

$$\hat{\mathbf{E}}_{l,m} = \begin{bmatrix} E_{1,\frac{1}{2},l,m} & E_{2,\frac{1}{2},l-\frac{1}{2},m} & E_{2,,l+\frac{1}{2},m} & E_{3,\frac{1}{2},l,m-\frac{1}{2}} & E_{3,\frac{1}{2},l,m+\frac{1}{2}} & \cdots \\ E_{1,k,l,m} & E_{2,k,l-\frac{1}{2},m} & E_{2,k,l+\frac{1}{2},m} & E_{3,k,l,m-\frac{1}{2}} & E_{3,k,l,m+\frac{1}{2}} & \cdots \\ E_{1,N_x-1,l,m} & E_{2,N_x-1,l-\frac{1}{2},m} & E_{2,N_x-1,l+\frac{1}{2},m} & E_{3,N_x-1,l,m-\frac{1}{2}} & E_{3,N_x-1,l,m+\frac{1}{2}} \\ E_{1,N_x,l,m} \end{bmatrix}^T$$

Matrix \hat{A} is splitted up in submatrices,

$$\hat{A} = \begin{bmatrix} \hat{A}_{\Gamma_1} & & & & & \\ & \hat{A}_{int} & & & \emptyset & \\ & & \ddots & & & \\ & & & \hat{A}_{int} & & \\ & \emptyset & & & \hat{A}_{\Gamma_2} & \\ & & & & & \hat{A}_{\Gamma_3} \end{bmatrix}$$

The matrices will be given in the following order: \hat{A}_{int} , \hat{A}_{Γ_1} , \hat{A}_{Γ_3} and \hat{A}_{Γ_2} . First the entries of the \hat{A} matrix will be given. Next the entries of the right hand side of (32) will be given.

B.1 \hat{A}_{int}

The structure of \hat{A}_{int} is given as follows,

$$\begin{pmatrix} \mathbf{0} & l_{12} & l_{13} & l_{14} & l_{15} & m_{11} & m_{12} & m_{13} & m_{14} & m_{15} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & l_{22} & \mathbf{0} & \mathbf{0} & \mathbf{0} & m_{21} & m_{22} & \mathbf{0} & m_{24} & m_{25} & r_{21} & r_{22} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & l_{33} & \mathbf{0} & \mathbf{0} & m_{31} & \mathbf{0} & m_{33} & m_{34} & m_{35} & r_{31} & \mathbf{0} & r_{33} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & l_{44} & \mathbf{0} & m_{41} & m_{42} & m_{43} & m_{44} & \mathbf{0} & r_{41} & \mathbf{0} & \mathbf{0} & r_{44} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & l_{55} & m_{51} & m_{52} & m_{53} & \mathbf{0} & m_{55} & r_{51} & \mathbf{0} & \mathbf{0} & \mathbf{0} & r_{55} \end{pmatrix}$$

where,

$l_{11} = 0$	$m_{11} = S_{k-\frac{1}{2},l,m} - \left(e^y_{l+\frac{1}{2}}\right)^2 M_{k-\frac{1}{2},l+\frac{1}{2},m} - \dots$ $\left(e^y_{l-\frac{1}{2}}\right)^2 M_{k-\frac{1}{2},l-\frac{1}{2},m} - \left(e^z_{m+\frac{1}{2}}\right)^2 M_{k-\frac{1}{2},l,m+\frac{1}{2}} - \dots$ $\left(e^z_{m-\frac{1}{2}}\right)^2 M_{k-\frac{1}{2},l,m-\frac{1}{2}}$
$l_{12} = -e^y_{l-\frac{1}{2}} e^x_{k-\frac{1}{2}} M_{k-\frac{1}{2},l-\frac{1}{2},m}$	$m_{12} = -l_{12}$
$l_{13} = e^y_{l+\frac{1}{2}} e^x_{k-\frac{1}{2}} M_{k-\frac{1}{2},l+\frac{1}{2},m}$	$m_{13} = -l_{13}$
$l_{14} = -e^z_{m-\frac{1}{2}} e^x_{k-\frac{1}{2}} M_{k-\frac{1}{2},l,m-\frac{1}{2}}$	$m_{14} = -l_{14}$
$l_{15} = e^z_{m+\frac{1}{2}} e^x_{k-\frac{1}{2}} M_{k-\frac{1}{2},l,m+\frac{1}{2}}$	$m_{15} = -l_{15}$

$l_{21} = 0$	$m_{21} = e^x_{k-\frac{1}{2}} e^y_{l-\frac{1}{2}} M_{k-\frac{1}{2},l-\frac{1}{2},m}$	$r_{21} = -e^x_{k+\frac{1}{2}} e^y_{l-\frac{1}{2}} M_{k+\frac{1}{2},l-\frac{1}{2},m}$
$l_{22} = \left(e^x_{k-\frac{1}{2}}\right)^2 M_{k-\frac{1}{2},l-\frac{1}{2},m}$	$m_{22} = S_{k,l-\frac{1}{2},m} - \dots$ $\left(e^z_{m+\frac{1}{2}}\right)^2 M_{k,l-\frac{1}{2},m+\frac{1}{2}} - \dots$ $\left(e^z_{m-\frac{1}{2}}\right)^2 M_{k,l-\frac{1}{2},m-\frac{1}{2}} - \dots$ $\left(e^x_{k+\frac{1}{2}}\right)^2 M_{k+\frac{1}{2},l-\frac{1}{2},m} - \dots$ $\left(e^x_{k-\frac{1}{2}}\right)^2 M_{k-\frac{1}{2},l-\frac{1}{2},m}$	$r_{22} = \left(e^x_{k+\frac{1}{2}}\right)^2 M_{k+\frac{1}{2},l-\frac{1}{2},m}$
$l_{23} = 0$	$m_{23} = 0$	$r_{23} = 0$
$l_{24} = 0$	$m_{24} = e^z_{m-\frac{1}{2}} e^y_{l-\frac{1}{2}} M_{k,l-\frac{1}{2},m-\frac{1}{2}}$	$r_{24} = 0$
$l_{25} = 0$	$m_{25} = -e^z_{m+\frac{1}{2}} e^y_{l-\frac{1}{2}} M_{k,l-\frac{1}{2},m+\frac{1}{2}}$	$r_{25} = 0$

$l_{31} = 0$	$m_{31} = -e^x_{k-\frac{1}{2}} e^y_{l+\frac{1}{2}} M_{k-\frac{1}{2},l+\frac{1}{2},m}$	$r_{31} = e^x_{k+\frac{1}{2}} e^y_{l+\frac{1}{2}} M_{k+\frac{1}{2},l+\frac{1}{2},m}$
$l_{32} = 0$	$m_{32} = 0$	$r_{32} = 0$
$l_{33} = \left(e^x_{k-\frac{1}{2}}\right)^2 M_{k-\frac{1}{2},l+\frac{1}{2},m}$	$m_{33} = S_{k,l+\frac{1}{2},m} - \dots$ $\left(e^z_{m+\frac{1}{2}}\right)^2 M_{k,l+\frac{1}{2},m+\frac{1}{2}} - \dots$ $\left(e^z_{m-\frac{1}{2}}\right)^2 M_{k,l+\frac{1}{2},m-\frac{1}{2}} - \dots$ $\left(e^x_{k+\frac{1}{2}}\right)^2 M_{k+\frac{1}{2},l+\frac{1}{2},m} - \dots$ $\left(e^x_{k-\frac{1}{2}}\right)^2 M_{k-\frac{1}{2},l+\frac{1}{2},m}$	$r_{33} = \left(e^x_{k+\frac{1}{2}}\right)^2 M_{k+\frac{1}{2},l+\frac{1}{2},m}$
$l_{34} = 0$	$m_{34} = -e^z_{m-\frac{1}{2}} e^y_{l+\frac{1}{2}} M_{k,l+\frac{1}{2},m-\frac{1}{2}}$	$r_{34} = 0$
$l_{35} = 0$	$m_{35} = e^z_{m+\frac{1}{2}} e^y_{l+\frac{1}{2}} M_{k,l+\frac{1}{2},m+\frac{1}{2}}$	$r_{35} = 0$

$l_{41} = 0$	$m_{41} = e^{x_{k-\frac{1}{2}}} e^{z_{m-\frac{1}{2}}} M_{k-\frac{1}{2}, l, m-\frac{1}{2}}$	$r_{41} = -e^{x_{k+\frac{1}{2}}} e^{z_{m-\frac{1}{2}}} M_{k+\frac{1}{2}, l, m-\frac{1}{2}}$
$l_{42} = 0$	$m_{42} = e^{y_{l-\frac{1}{2}}} e^{z_{m-\frac{1}{2}}} M_{k, l-\frac{1}{2}, m-\frac{1}{2}}$	$r_{42} = 0$
$l_{43} = 0$	$m_{43} = -e^{y_{l+\frac{1}{2}}} e^{z_{m-\frac{1}{2}}} M_{k, l+\frac{1}{2}, m-\frac{1}{2}}$	$r_{43} = 0$
$l_{44} = \left(e^{x_{k-\frac{1}{2}}}\right)^2 M_{k-\frac{1}{2}, l, m-\frac{1}{2}}$	$m_{44} = S_{k, l, m-\frac{1}{2}} - \dots$ $\left(e^{x_{k+\frac{1}{2}}}\right)^2 M_{k+\frac{1}{2}, l, m-\frac{1}{2}} - \dots$ $\left(e^{x_{k-\frac{1}{2}}}\right)^2 M_{k-\frac{1}{2}, l, m-\frac{1}{2}} - \dots$ $\left(e^{y_{l+\frac{1}{2}}}\right)^2 M_{k, l+\frac{1}{2}, m-\frac{1}{2}} - \dots$ $\left(e^{y_{l-\frac{1}{2}}}\right)^2 M_{k, l-\frac{1}{2}, m-\frac{1}{2}}$	$r_{44} = \left(e^{x_{k+\frac{1}{2}}}\right)^2 M_{k+\frac{1}{2}, l, m-\frac{1}{2}}$
$l_{45} = 0$	$m_{45} = 0$	$r_{45} = 0$

$l_{31} = 0$	$m_{31} = -e^{x_{k-\frac{1}{2}}} e^{z_{m+\frac{1}{2}}} M_{k-\frac{1}{2}, l, m+\frac{1}{2}}$	$r_{31} = e^{x_{k+\frac{1}{2}}} e^{z_{m+\frac{1}{2}}} M_{k+\frac{1}{2}, l, m+\frac{1}{2}}$
$l_{32} = 0$	$m_{32} = -e^{y_{l-\frac{1}{2}}} e^{z_{m+\frac{1}{2}}} M_{k, l-\frac{1}{2}, m+\frac{1}{2}}$	$r_{32} = 0$
$l_{33} = 0$	$m_{33} = e^{y_{l+\frac{1}{2}}} e^{z_{m+\frac{1}{2}}} M_{k, l+\frac{1}{2}, m+\frac{1}{2}}$	$r_{33} = 0$
$l_{34} = 0$	$m_{34} = 0$	$r_{34} = 0$
$l_{35} = \left(e^{x_{k-\frac{1}{2}}}\right)^2 M_{k-\frac{1}{2}, l, m+\frac{1}{2}}$	$m_{35} = S_{k, l, m+\frac{1}{2}} - \dots$ $\left(e^{x_{k+\frac{1}{2}}}\right)^2 M_{k+\frac{1}{2}, l, m+\frac{1}{2}} - \dots$ $\left(e^{x_{k-\frac{1}{2}}}\right)^2 M_{k-\frac{1}{2}, l, m+\frac{1}{2}} - \dots$ $\left(e^{y_{l+\frac{1}{2}}}\right)^2 M_{k, l+\frac{1}{2}, m+\frac{1}{2}} - \dots$ $\left(e^{y_{l-\frac{1}{2}}}\right)^2 M_{k, l-\frac{1}{2}, m+\frac{1}{2}}$	$r_{35} = \left(e^{x_{k+\frac{1}{2}}}\right)^2 M_{k+\frac{1}{2}, l, m+\frac{1}{2}}$

The *local* right hand side of (32) belonging to matrix \hat{A}_{int} is a 5x1 vector,

$$\begin{pmatrix}
-s_{1,k-\frac{1}{2},l,m} - \left(e_{l+\frac{1}{2}}^y\right)^2 M_{k-\frac{1}{2},l+\frac{1}{2},m} E_{1,k-\frac{1}{2},l+1,m} - \left(e_{l-\frac{1}{2}}^y\right)^2 M_{k-\frac{1}{2},l-\frac{1}{2},m} E_{1,k-\frac{1}{2},l-1,m} - \dots \\
\left(e_{m+\frac{1}{2}}^z\right)^2 M_{k-\frac{1}{2},l,m+\frac{1}{2}} E_{1,k-\frac{1}{2},l,m+1} - \left(e_{m-\frac{1}{2}}^z\right)^2 M_{k-\frac{1}{2},l,m-\frac{1}{2}} E_{1,k-\frac{1}{2},l,m-1} \\
-s_{2,k,l-\frac{1}{2},m} - \left[e_{l-\frac{1}{2}}^y E_{3,k,l-1,m+\frac{1}{2}} + e_{m+\frac{1}{2}}^z E_{2,k,l-\frac{1}{2},m+1}\right] e_{m+\frac{1}{2}}^z M_{k,l-\frac{1}{2},m+\frac{1}{2}} + \dots \\
\left[e_{l-\frac{1}{2}}^y E_{3,k,l-1,m-\frac{1}{2}} - e_{m-\frac{1}{2}}^z E_{2,k,l-\frac{1}{2},m-1}\right] e_{m-\frac{1}{2}}^z M_{k,l-\frac{1}{2},m-\frac{1}{2}} - \dots \\
\left[e_{k+\frac{1}{2}}^x M_{k+\frac{1}{2},l-\frac{1}{2},m} E_{1,k+\frac{1}{2},l-1,m} - e_{k-\frac{1}{2}}^x M_{k-\frac{1}{2},l-\frac{1}{2},m} E_{1,k-\frac{1}{2},l-1,m}\right] e_{l-\frac{1}{2}}^y \\
-s_{2,k,l+\frac{1}{2},m} + \left[e_{l+\frac{1}{2}}^y E_{3,k,l+1,m+\frac{1}{2}} - e_{m+\frac{1}{2}}^z E_{2,k,l+\frac{1}{2},m+1}\right] e_{m+\frac{1}{2}}^z M_{k,l+\frac{1}{2},m+\frac{1}{2}} - \dots \\
\left[e_{l+\frac{1}{2}}^y E_{3,k,l+1,m-\frac{1}{2}} + e_{m-\frac{1}{2}}^z E_{2,k,l+\frac{1}{2},m-1}\right] e_{m-\frac{1}{2}}^z M_{k,l+\frac{1}{2},m-\frac{1}{2}} + \dots \\
\left[e_{k+\frac{1}{2}}^x M_{k+\frac{1}{2},l+\frac{1}{2},m} E_{1,k+\frac{1}{2},l+1,m} - e_{k-\frac{1}{2}}^x M_{k-\frac{1}{2},l+\frac{1}{2},m} E_{1,k-\frac{1}{2},l+1,m}\right] e_{l+\frac{1}{2}}^y \\
-s_{3,k,l,m-\frac{1}{2}} + \left[e_{k-\frac{1}{2}}^x M_{k-\frac{1}{2},l,m-\frac{1}{2}} E_{1,k-\frac{1}{2},l,m-1} - e_{k+\frac{1}{2}}^x M_{k+\frac{1}{2},l,m-\frac{1}{2}} E_{1,k+\frac{1}{2},l,m-1}\right] e_{m-\frac{1}{2}}^z - \dots \\
\left[e_{l+\frac{1}{2}}^y E_{3,k,l+1,m-\frac{1}{2}} + e_{m-\frac{1}{2}}^z E_{2,k,l+\frac{1}{2},m-1}\right] e_{l+\frac{1}{2}}^y M_{k,l+\frac{1}{2},m-\frac{1}{2}} - \dots \\
\left[e_{l-\frac{1}{2}}^y E_{3,k,l-1,m-\frac{1}{2}} - e_{m-\frac{1}{2}}^z E_{2,k,l-\frac{1}{2},m-1}\right] e_{l-\frac{1}{2}}^y M_{k,l-\frac{1}{2},m-\frac{1}{2}} \\
-s_{3,k,l,m+\frac{1}{2}} + \left[e_{k-\frac{1}{2}}^x M_{k-\frac{1}{2},l,m+\frac{1}{2}} E_{1,k-\frac{1}{2},l,m+1} - e_{k+\frac{1}{2}}^x M_{k+\frac{1}{2},l,m+\frac{1}{2}} E_{1,k+\frac{1}{2},l,m+1}\right] e_{m+\frac{1}{2}}^z - \dots \\
\left[e_{l+\frac{1}{2}}^y E_{3,k,l+1,m+\frac{1}{2}} + e_{m+\frac{1}{2}}^z E_{2,k,l+\frac{1}{2},m+1}\right] e_{l+\frac{1}{2}}^y M_{k,l+\frac{1}{2},m+\frac{1}{2}} - \dots \\
\left[e_{l-\frac{1}{2}}^y E_{3,k,l-1,m+\frac{1}{2}} - e_{m-\frac{1}{2}}^z E_{2,k,l-\frac{1}{2},m+1}\right] e_{l-\frac{1}{2}}^y M_{k,l-\frac{1}{2},m+\frac{1}{2}}
\end{pmatrix} \quad (33)$$

B.2 \mathbf{A}_{Γ_1}

The structure of \hat{A}_{Γ_1} is given as follows,

$$\begin{pmatrix}
m_{11} & m_{12} & m_{13} & m_{14} & m_{15} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
m_{21} & m_{22} & \mathbf{0} & m_{24} & m_{25} & r_{21} & r_{22} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
m_{31} & \mathbf{0} & m_{33} & m_{34} & m_{35} & r_{31} & \mathbf{0} & r_{33} & \mathbf{0} & \mathbf{0} \\
m_{41} & m_{42} & m_{43} & m_{44} & \mathbf{0} & r_{41} & \mathbf{0} & \mathbf{0} & r_{44} & \mathbf{0} \\
m_{51} & m_{52} & m_{53} & \mathbf{0} & m_{55} & r_{51} & \mathbf{0} & \mathbf{0} & \mathbf{0} & r_{55}
\end{pmatrix}$$

The entries of this matrix are similar to the m_{ij} - and r_{ij} - entries of matrix \hat{A}_{int} . Due to the PEC boundary conditions there is no contribution of the boundary to matrix \hat{A}_{Γ_1} . The *local* right hand side of (32) belonging to matrix \hat{A}_{Γ_1} is a 5x1 vector. The entries are similar to vector (33).

B.3 \mathbf{A}_{Γ_2}

The structure of \hat{A}_{Γ_2} is given as follows,

$$\begin{pmatrix} \mathbf{0} & l_{12} & l_{13} & l_{14} & l_{15} & m_{11} & m_{12} & m_{13} & m_{14} & m_{15} & \mathbf{0} \\ \mathbf{0} & l_{22} & \mathbf{0} & \mathbf{0} & \mathbf{0} & m_{21} & m_{22} & \mathbf{0} & m_{24} & m_{25} & r_{21} \\ \mathbf{0} & \mathbf{0} & l_{33} & \mathbf{0} & \mathbf{0} & m_{31} & \mathbf{0} & m_{33} & m_{34} & m_{35} & r_{31} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & l_{44} & \mathbf{0} & m_{41} & m_{42} & m_{43} & m_{44} & \mathbf{0} & r_{41} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & l_{55} & m_{51} & m_{52} & m_{53} & \mathbf{0} & m_{55} & r_{51} \end{pmatrix}$$

The entries of this matrix are similar to the l_{ij} -, m_{ij} - and r_{ij} - entries of matrix \hat{A}_{int} . Due to the PEC boundary conditions there is no contribution of the boundary to matrix \hat{A}_{Γ_2} . The *local* right hand side of (32) belonging to matrix \hat{A}_{Γ_2} is a 5x1 vector. The entries are similar to vector (33).

B.4 \mathbf{A}_{Γ_3}

The structure of \hat{A}_{Γ_3} is given as follows,

$$(\mathbf{0} \quad l_{12} \quad l_{13} \quad l_{14} \quad l_{15} \quad m_{11})$$

The entries of this matrix are similar to the l_{ij} -, m_{ij} - and r_{ij} - entries of matrix \hat{A}_{int} . Due to the PEC boundary conditions there is no contribution of the boundary to matrix \hat{A}_{Γ_3} . The *local* right hand side of (32) belonging to matrix \hat{A}_{Γ_3} is a 1x1 vector.

$$\begin{pmatrix} -s_{1,k-\frac{1}{2},l,m} - \left(e_{l+\frac{1}{2}}^y\right)^2 M_{k-\frac{1}{2},l+\frac{1}{2},m} E_{1,k-\frac{1}{2},l+1,m} - \left(e_{l-\frac{1}{2}}^y\right)^2 M_{k-\frac{1}{2},l-\frac{1}{2},m} E_{1,k-\frac{1}{2},l-1,m} - \dots \\ \left(e_{m+\frac{1}{2}}^z\right)^2 M_{k-\frac{1}{2},l,m+\frac{1}{2}} E_{1,k-\frac{1}{2},l,m+1} - \left(e_{m-\frac{1}{2}}^z\right)^2 M_{k-\frac{1}{2},l,m-\frac{1}{2}} E_{1,k-\frac{1}{2},l,m-1} \end{pmatrix}$$

C Local mode analysis

C.1 Cell-Block Smoother

Introduce, $F = e^{i(k\theta_1 + l\theta_2 + m\theta_3)}$ and write $h^{-2} = \left(e^x_{k \pm \frac{1}{2}}\right)^2 = \left(e^y_{l \pm \frac{1}{2}}\right)^2 = \left(e^z_{m \pm \frac{1}{2}}\right)^2$.

First equation,

$$\begin{aligned}
 & F e^{i(-\frac{\theta_1}{2})} \left[\begin{array}{l} -d_{1,k-\frac{1}{2},l,m} \frac{(\delta u_2 - \delta u_1)}{\omega_u} \\ +h^{-2} M_{k-\frac{1}{2},l+\frac{1}{2},m} e^{i(\frac{\theta_1}{2} + \frac{\theta_2}{2})} \frac{(\delta v_1 - \delta v_0)}{\omega_v} \\ -h^{-2} M_{k-\frac{1}{2},l-\frac{1}{2},m} e^{i(\frac{\theta_1}{2} - \frac{\theta_2}{2})} \frac{(\delta v_2 - \delta v_1)}{\omega_v} \\ +h^{-2} M_{k-\frac{1}{2},l,m+\frac{1}{2}} e^{i(\frac{\theta_1}{2} + \frac{\theta_3}{2})} \frac{(\delta w_1 - \delta w_0)}{\omega_w} \\ -h^{-2} M_{k-\frac{1}{2},l,m-\frac{1}{2}} e^{i(\frac{\theta_1}{2} - \frac{\theta_3}{2})} \frac{(\delta w_2 - \delta w_1)}{\omega_w} \end{array} \right] \\
 = & F e^{i(-\frac{\theta_1}{2})} \left[\begin{array}{l} h^{-2} \left(M_{k-\frac{1}{2},l+\frac{1}{2},m} e^{i(\theta_2)} + M_{k-\frac{1}{2},l,m+\frac{1}{2}} e^{i(\theta_3)} \right) \delta u_0 \\ +d_{1,k-\frac{1}{2},l,m} \delta u_1 \\ +h^{-2} \left(M_{k-\frac{1}{2},l-\frac{1}{2},m} e^{i(-\theta_2)} + M_{k-\frac{1}{2},l,m-\frac{1}{2}} e^{i(-\theta_3)} \right) \delta u_2 \\ -h^{-2} M_{k-\frac{1}{2},l+\frac{1}{2},m} e^{i(\frac{\theta_1}{2} + \frac{\theta_2}{2})} \delta v_0 \\ +h^{-2} \left(M_{k-\frac{1}{2},l-\frac{1}{2},m} e^{i(\frac{\theta_1}{2} - \frac{\theta_2}{2})} + M_{k-\frac{1}{2},l+\frac{1}{2},m} e^{i(-\frac{\theta_1}{2} + \frac{\theta_2}{2})} \right) \delta v_1 \\ -h^{-2} M_{k-\frac{1}{2},l-\frac{1}{2},m} e^{i(-\frac{\theta_1}{2} - \frac{\theta_2}{2})} \delta v_2 \\ -h^{-2} M_{k-\frac{1}{2},l,m+\frac{1}{2}} e^{i(\frac{\theta_1}{2} + \frac{\theta_3}{2})} \delta w_0 \\ +h^{-2} \left(M_{k-\frac{1}{2},l,m-\frac{1}{2}} e^{i(\frac{\theta_1}{2} - \frac{\theta_3}{2})} + M_{k-\frac{1}{2},l,m+\frac{1}{2}} e^{i(-\frac{\theta_1}{2} + \frac{\theta_3}{2})} \right) \delta w_1 \\ -h^{-2} M_{k-\frac{1}{2},l,m-\frac{1}{2}} e^{i(-\frac{\theta_1}{2} - \frac{\theta_3}{2})} \delta w_2 \end{array} \right]
 \end{aligned}$$

with,

$$d_{1,k-\frac{1}{2},l,m} = S_{k-\frac{1}{2},l,m} - h^{-2} \left(M_{k-\frac{1}{2},l+\frac{1}{2},m} + M_{k-\frac{1}{2},l-\frac{1}{2},m} + M_{k-\frac{1}{2},l,m+\frac{1}{2}} + M_{k-\frac{1}{2},l,m-\frac{1}{2}} \right)$$

Second equation,

$$\begin{aligned}
& F e^{i(+\frac{\theta_1}{2})} \left[\begin{array}{l} -d_{1,k+\frac{1}{2},l,m} \frac{(\delta u_1 - \delta u_0)}{\omega_u} \\ -h^{-2} M_{k+\frac{1}{2},l+\frac{1}{2},m} e^{i(-\frac{\theta_1}{2} + \frac{\theta_2}{2})} \frac{(\delta v_1 - \delta v_0)}{\omega_v} \\ +h^{-2} M_{k+\frac{1}{2},l-\frac{1}{2},m} e^{i(-\frac{\theta_1}{2} - \frac{\theta_2}{2})} \frac{(\delta v_2 - \delta v_1)}{\omega_v} \\ -h^{-2} M_{k+\frac{1}{2},l,m+\frac{1}{2}} e^{i(-\frac{\theta_1}{2} + \frac{\theta_3}{2})} \frac{(\delta w_1 - \delta w_0)}{\omega_w} \\ +h^{-2} M_{k+\frac{1}{2},l,m-\frac{1}{2}} e^{i(-\frac{\theta_1}{2} - \frac{\theta_3}{2})} \frac{(\delta w_2 - \delta w_1)}{\omega_w} \end{array} \right] \\
= & F e^{i(+\frac{\theta_1}{2})} \left[\begin{array}{l} h^{-2} \left(M_{k+\frac{1}{2},l+\frac{1}{2},m} e^{i(\theta_2)} + M_{k+\frac{1}{2},l,m+\frac{1}{2}} e^{i(\theta_3)} + h^2 d_{1,k+\frac{1}{2},l,m} \right) \partial u_0 \\ +h^{-2} \left(M_{k+\frac{1}{2},l-\frac{1}{2},m} e^{i(-\theta_2)} + M_{k+\frac{1}{2},l,m-\frac{1}{2}} e^{i(-\theta_3)} \right) \partial u_2 \\ +h^{-2} M_{k+\frac{1}{2},l+\frac{1}{2},m} \left(e^{i(-\frac{\theta_1}{2} + \frac{\theta_2}{2})} - e^{i(\frac{\theta_1}{2} + \frac{\theta_2}{2})} \right) \partial v_0 \\ +h^{-2} M_{k+\frac{1}{2},l-\frac{1}{2},m} \left(e^{i(\frac{\theta_1}{2} - \frac{\theta_2}{2})} - e^{i(-\frac{\theta_1}{2} - \frac{\theta_2}{2})} \right) \partial v_1 \\ +h^{-2} M_{k+\frac{1}{2},l,m+\frac{1}{2}} \left(e^{i(-\frac{\theta_1}{2} + \frac{\theta_3}{2})} - e^{i(\frac{\theta_1}{2} + \frac{\theta_3}{2})} \right) \partial w_0 \\ +h^{-2} M_{k+\frac{1}{2},l,m-\frac{1}{2}} \left(e^{i(\frac{\theta_1}{2} - \frac{\theta_3}{2})} - e^{i(-\frac{\theta_1}{2} - \frac{\theta_3}{2})} \right) \partial w_1 \end{array} \right]
\end{aligned}$$

with,

$$d_{1,k+\frac{1}{2},l,m} = S_{k+\frac{1}{2},l,m} - h^{-2} \left(M_{k+\frac{1}{2},l+\frac{1}{2},m} + M_{k+\frac{1}{2},l-\frac{1}{2},m} + M_{k+\frac{1}{2},l,m+\frac{1}{2}} + M_{k+\frac{1}{2},l,m-\frac{1}{2}} \right)$$

Third equation,

$$\begin{aligned}
& Fe^{i(-\frac{\theta_2}{2})} \left[\begin{array}{l} +h^{-2}M_{k+\frac{1}{2},l-\frac{1}{2},m}e^{i(\frac{\theta_1}{2}+\frac{\theta_2}{2})}\frac{(\delta u_1-\delta u_0)}{\omega_u} \\ -h^{-2}M_{k-\frac{1}{2},l-\frac{1}{2},m}e^{i(-\frac{\theta_1}{2}+\frac{\theta_2}{2})}\frac{(\delta u_2-\delta u_1)}{\omega_u} \\ -d_{2,k,l-\frac{1}{2},m}\frac{(\delta v_2-\delta v_1)}{\omega_u} \\ +h^{-2}M_{k,l-\frac{1}{2},m+\frac{1}{2}}e^{i(\frac{\theta_2}{2}+\frac{\theta_3}{2})}\frac{(\delta w_1-\delta w_0)}{\omega_w} \\ -h^{-2}M_{k,l-\frac{1}{2},m-\frac{1}{2}}e^{i(\frac{\theta_2}{2}-\frac{\theta_3}{2})}\frac{(\delta w_2-\delta w_1)}{\omega_w} \end{array} \right] \\
= & Fe^{i(-\frac{\theta_2}{2})} \left[\begin{array}{l} -h^{-2}M_{k+\frac{1}{2},l-\frac{1}{2},m}e^{i(\frac{\theta_1}{2}+\frac{\theta_2}{2})}\partial u_0 \\ +h^{-2}M_{k-\frac{1}{2},l-\frac{1}{2},m}e^{i(-\frac{\theta_1}{2}+\frac{\theta_2}{2})}\partial u_1 \\ +h^{-2}\left(M_{k+\frac{1}{2},l-\frac{1}{2},m}e^{i(\frac{\theta_1}{2}-\frac{\theta_2}{2})}-M_{k-\frac{1}{2},l-\frac{1}{2},m}e^{i(-\frac{\theta_1}{2}-\frac{\theta_2}{2})}\right)\partial u_2 \\ +h^{-2}M_{k,l-\frac{1}{2},m+\frac{1}{2}}e^{i(\theta_3)}\partial v_0 \\ +h^{-2}\left(M_{k+\frac{1}{2},l-\frac{1}{2},m}e^{i(\theta_1)}+h^2d_{2,k,l-\frac{1}{2},m}\right)\partial v_1 \\ +h^{-2}\left(M_{k,l-\frac{1}{2},m-\frac{1}{2}}e^{i(-\theta_3)}+M_{k-\frac{1}{2},l-\frac{1}{2},m}e^{i(-\theta_1)}\right)\partial v_2 \\ -h^{-2}M_{k,l-\frac{1}{2},m+\frac{1}{2}}e^{i(\frac{\theta_2}{2}+\frac{\theta_3}{2})}\partial w_0 \\ +h^{-2}\left(M_{k,l-\frac{1}{2},m-\frac{1}{2}}e^{i(\frac{\theta_2}{2}-\frac{\theta_3}{2})}+M_{k,l-\frac{1}{2},m+\frac{1}{2}}e^{i(-\frac{\theta_2}{2}+\frac{\theta_3}{2})}\right)\partial w_1 \\ -h^{-2}M_{k,l-\frac{1}{2},m-\frac{1}{2}}e^{i(-\frac{\theta_2}{2}-\frac{\theta_3}{2})}\partial w_2 \end{array} \right]
\end{aligned}$$

with,

$$d_{2,k,l-\frac{1}{2},m} = S_{k,l-\frac{1}{2},m} - h^{-2} \left(M_{k,l-\frac{1}{2},m+\frac{1}{2}} + M_{k,l-\frac{1}{2},m-\frac{1}{2}} + M_{k+\frac{1}{2},l-\frac{1}{2},m} + M_{k-\frac{1}{2},l-\frac{1}{2},m} \right)$$

Fourth equation,

$$\begin{aligned}
& F e^{i(+\frac{\theta_2}{2})} \left[\begin{array}{l} -h^2 \left(M_{k+\frac{1}{2}, l+\frac{1}{2}, m} e^{i(\frac{\theta_1}{2}-\frac{\theta_2}{2})} \right) \frac{(\delta u_1 - \delta u_0)}{\omega_u} \\ + h^{-2} M_{k-\frac{1}{2}, l+\frac{1}{2}, m} e^{i(-\frac{\theta_1}{2}-\frac{\theta_2}{2})} \frac{(\delta u_2 - \delta u_1)}{\omega_u} \\ - d_{2,k, l+\frac{1}{2}, m} \frac{(\delta v_1 - \delta v_0)}{\omega_v} \\ - h^{-2} M_{k, l+\frac{1}{2}, m+\frac{1}{2}} e^{i(-\frac{\theta_2}{2}+\frac{\theta_3}{2})} \frac{(\delta w_1 - \delta w_0)}{\omega_w} \\ + h^{-2} M_{k, l+\frac{1}{2}, m-\frac{1}{2}} e^{i(-\frac{\theta_2}{2}-\frac{\theta_3}{2})} \frac{(\delta w_2 - \delta w_1)}{\omega_w} \end{array} \right] \\
= & F e^{i(+\frac{\theta_2}{2})} \left[\begin{array}{l} h^2 \left(M_{k+\frac{1}{2}, l+\frac{1}{2}, m} \left(e^{i(\frac{\theta_1}{2}-\frac{\theta_2}{2})} - e^{i(\frac{\theta_1}{2}+\frac{\theta_2}{2})} \right) + M_{k-\frac{1}{2}, l+\frac{1}{2}, m} e^{i(-\frac{\theta_1}{2}+\frac{\theta_2}{2})} \right) \partial u_0 \\ - h^{-2} M_{k-\frac{1}{2}, l+\frac{1}{2}, m} e^{i(-\frac{\theta_1}{2}-\frac{\theta_2}{2})} \partial u_1 \\ + h^{-2} \left(M_{k, l+\frac{1}{2}, m+\frac{1}{2}} e^{i(\theta_3)} + M_{k+\frac{1}{2}, l+\frac{1}{2}, m} e^{i(\theta_1)} + h^2 d_{2,k, l+\frac{1}{2}, m} \right) \partial v_0 \\ + h^{-2} M_{k-\frac{1}{2}, l+\frac{1}{2}, m} e^{i(-\theta_1)} \partial v_1 \\ + h^{-2} M_{k, l+\frac{1}{2}, m-\frac{1}{2}} e^{i(-\theta_3)} \partial v_2 \\ + h^{-2} M_{k, l+\frac{1}{2}, m+\frac{1}{2}} \left(e^{i(-\frac{\theta_2}{2}+\frac{\theta_3}{2})} - e^{i(\frac{\theta_2}{2}+\frac{\theta_3}{2})} \right) \partial w_0 \\ + h^{-2} M_{k, l+\frac{1}{2}, m-\frac{1}{2}} \left(e^{i(\frac{\theta_2}{2}-\frac{\theta_3}{2})} - e^{i(-\frac{\theta_2}{2}-\frac{\theta_3}{2})} \right) \partial w_1 \end{array} \right]
\end{aligned}$$

with,

$$d_{2,k, l+\frac{1}{2}, m} = S_{k, l+\frac{1}{2}, m} - h^{-2} \left(M_{k, l+\frac{1}{2}, m+\frac{1}{2}} + M_{k, l+\frac{1}{2}, m-\frac{1}{2}} + M_{k+\frac{1}{2}, l+\frac{1}{2}, m} + M_{k-\frac{1}{2}, l+\frac{1}{2}, m} \right)$$

Fifth equation,

$$\begin{aligned}
& F e^{i(-\frac{\theta_3}{2})} \left[\begin{array}{l} +h^{-2} M_{k+\frac{1}{2}, l, m-\frac{1}{2}} e^{i(\frac{\theta_1}{2} + \frac{\theta_3}{2})} \frac{(\delta u_1 - \delta u_0)}{\omega_u} \\ -h^{-2} M_{k-\frac{1}{2}, l, m-\frac{1}{2}} e^{i(-\frac{\theta_1}{2} + \frac{\theta_3}{2})} \frac{(\delta u_2 - \delta u_1)}{\omega_u} \\ +h^{-2} M_{k, l+\frac{1}{2}, m-\frac{1}{2}} e^{i(\frac{\theta_2}{2} + \frac{\theta_3}{2})} \frac{(\delta v_1 - \delta v_0)}{\omega_v} \\ -h^{-2} M_{k, l-\frac{1}{2}, m-\frac{1}{2}} e^{i(-\frac{\theta_2}{2} + \frac{\theta_3}{2})} \frac{(\delta v_2 - \delta v_1)}{\omega_v} \\ -d_{3, k, l, m-\frac{1}{2}} \frac{(\delta w_2 - \delta w_1)}{\omega_w} \end{array} \right] \\
= & F e^{i(-\frac{\theta_3}{2})} \left[\begin{array}{l} -h^{-2} M_{k+\frac{1}{2}, l, m-\frac{1}{2}} e^{i(\frac{\theta_1}{2} + \frac{\theta_3}{2})} \partial u_0 \\ +h^{-2} M_{k-\frac{1}{2}, l, m-\frac{1}{2}} e^{i(-\frac{\theta_1}{2} + \frac{\theta_3}{2})} \partial u_1 \\ +h^{-2} \left(M_{k+\frac{1}{2}, l, m-\frac{1}{2}} e^{i(\frac{\theta_1}{2} - \frac{\theta_3}{2})} - M_{k-\frac{1}{2}, l, m-\frac{1}{2}} e^{i(-\frac{\theta_1}{2} - \frac{\theta_3}{2})} \right) \partial u_2 \\ -h^{-2} M_{k, l+\frac{1}{2}, m-\frac{1}{2}} e^{i(\frac{\theta_2}{2} + \frac{\theta_3}{2})} \partial v_0 \\ +h^{-2} M_{k, l-\frac{1}{2}, m-\frac{1}{2}} e^{i(-\frac{\theta_2}{2} + \frac{\theta_3}{2})} \partial v_1 \\ +h^{-2} \left(M_{k, l+\frac{1}{2}, m-\frac{1}{2}} e^{i(\frac{\theta_2}{2} - \frac{\theta_3}{2})} - M_{k, l-\frac{1}{2}, m-\frac{1}{2}} e^{i(-\frac{\theta_2}{2} - \frac{\theta_3}{2})} \right) \partial v_2 \\ +h^{-2} \left(M_{k+\frac{1}{2}, l, m-\frac{1}{2}} e^{i(\theta_1)} + M_{k, l+\frac{1}{2}, m-\frac{1}{2}} e^{i(\theta_2)} + h^2 d_{3, k, l, m-\frac{1}{2}} \right) \partial w_1 \\ +h^{-2} \left(M_{k-\frac{1}{2}, l, m-\frac{1}{2}} e^{i(-\theta_1)} + M_{k, l-\frac{1}{2}, m-\frac{1}{2}} e^{i(-\theta_3)} \right) \partial w_2 \end{array} \right]
\end{aligned}$$

with,

$$d_{3, k, l, m-\frac{1}{2}} = S_{k, l, m-\frac{1}{2}} - h^{-2} \left(M_{k+\frac{1}{2}, l, m-\frac{1}{2}} + M_{k-\frac{1}{2}, l, m-\frac{1}{2}} + M_{k, l+\frac{1}{2}, m-\frac{1}{2}} + M_{k, l-\frac{1}{2}, m-\frac{1}{2}} \right)$$

Sixth equation,

$$\begin{aligned}
& F e^{i(+\frac{\theta_3}{2})} \left[\begin{array}{l} -h^{-2} M_{k+\frac{1}{2},l,m+\frac{1}{2}} e^{i(\frac{\theta_1}{2}-\frac{\theta_3}{2})} \frac{(\delta u_1-\delta u_0)}{\omega_u} \\ +h^{-2} M_{k-\frac{1}{2},l,m+\frac{1}{2}} e^{i(-\frac{\theta_1}{2}-\frac{\theta_3}{2})} \frac{(\delta u_2-\delta u_1)}{\omega_u} \\ -h^{-2} M_{k,l+\frac{1}{2},m+\frac{1}{2}} e^{i(\frac{\theta_2}{2}-\frac{\theta_3}{2})} \frac{(\delta v_1-\delta v_0)}{\omega_v} \\ +h^{-2} M_{k,l-\frac{1}{2},m+\frac{1}{2}} e^{i(-\frac{\theta_2}{2}-\frac{\theta_3}{2})} \frac{(\delta v_2-\delta v_1)}{\omega_v} \\ -d_{3,k,l,m+\frac{1}{2}} \frac{(\delta w_1-\delta w_0)}{\omega_w} \end{array} \right] \\
= & F e^{i(+\frac{\theta_3}{2})} \left[\begin{array}{l} h^{-2} \left(M_{k+\frac{1}{2},l,m+\frac{1}{2}} \left(e^{i(\frac{\theta_1}{2}-\frac{\theta_3}{2})} - e^{i(\frac{\theta_1}{2}+\frac{\theta_3}{2})} \right) + M_{k-\frac{1}{2},l,m+\frac{1}{2}} e^{i(-\frac{\theta_1}{2}+\frac{\theta_3}{2})} \right) \partial u_0 \\ -h^{-2} M_{k-\frac{1}{2},l,m+\frac{1}{2}} e^{i(-\frac{\theta_1}{2}-\frac{\theta_3}{2})} \partial u_1 \\ +h^{-2} \left(M_{k,l+\frac{1}{2},m+\frac{1}{2}} \left(e^{i(\frac{\theta_2}{2}-\frac{\theta_3}{2})} - e^{i(\frac{\theta_2}{2}+\frac{\theta_3}{2})} \right) + M_{k,l-\frac{1}{2},m+\frac{1}{2}} e^{i(-\frac{\theta_2}{2}+\frac{\theta_3}{2})} \right) \partial v_0 \\ -h^{-2} M_{k,l-\frac{1}{2},m+\frac{1}{2}} e^{i(-\frac{\theta_2}{2}-\frac{\theta_3}{2})} \partial v_1 \\ +h^{-2} \left(M_{k+\frac{1}{2},l,m+\frac{1}{2}} e^{i(\theta_1)} + M_{k,l+\frac{1}{2},m+\frac{1}{2}} e^{i(\theta_2)} + h^2 d_{3,k,l,m+\frac{1}{2}} \right) \partial w_0 \\ +h^{-2} \left(M_{k-\frac{1}{2},l,m+\frac{1}{2}} e^{i(-\theta_1)} + M_{k,l-\frac{1}{2},m+\frac{1}{2}} e^{i(-\theta_2)} \right) \partial w_1 \end{array} \right]
\end{aligned}$$

with,

$$d_{3,k,l,m+\frac{1}{2}} = S_{k,l,m+\frac{1}{2}} - h^{-2} \left(M_{k+\frac{1}{2},l,m+\frac{1}{2}} + M_{k-\frac{1}{2},l,m+\frac{1}{2}} + M_{k,l+\frac{1}{2},m+\frac{1}{2}} + M_{k,l-\frac{1}{2},m+\frac{1}{2}} \right)$$

C.2 Line smoother

Introduce, $F = e^{i(k\theta_1 + l\theta_2 + m\theta_3)}$ and write $h^{-2} = \left(e^x_{k \pm \frac{1}{2}}\right)^2 = \left(e^y_{l \pm \frac{1}{2}}\right)^2 = \left(e^z_{m \pm \frac{1}{2}}\right)^2$.

First equation,

$$\begin{aligned}
& F e^{i(-\frac{\theta_1}{2})} \left[\begin{aligned} & -d_{1,k-\frac{1}{2},l,m} \frac{(\delta u_2 - \delta u_0)}{\omega_u} \\ & + h^{-2} \left(M_{k-\frac{1}{2},l+\frac{1}{2},m} e^{i(\frac{\theta_1}{2} + \frac{\theta_2}{2})} - M_{k-\frac{1}{2},l+\frac{1}{2},m} e^{i(-\frac{\theta_1}{2} + \frac{\theta_2}{2})} \right) \frac{(\delta v_1 - \delta v_0)}{\omega_v} \\ & + h^{-2} \left(M_{k-\frac{1}{2},l-\frac{1}{2},m} e^{i(-\frac{\theta_1}{2} - \frac{\theta_2}{2})} - M_{k-\frac{1}{2},l-\frac{1}{2},m} e^{i(\frac{\theta_1}{2} - \frac{\theta_2}{2})} \right) \frac{(\delta v_2 - \delta v_1)}{\omega_v} \\ & + h^{-2} \left(M_{k-\frac{1}{2},l,m+\frac{1}{2}} e^{i(\frac{\theta_1}{2} + \frac{\theta_3}{2})} - M_{k-\frac{1}{2},l,m+\frac{1}{2}} e^{i(-\frac{\theta_1}{2} + \frac{\theta_3}{2})} \right) \frac{(\delta w_1 - \delta w_0)}{\omega_w} \\ & + h^{-2} \left(M_{k-\frac{1}{2},l,m-\frac{1}{2}} e^{i(-\frac{\theta_1}{2} - \frac{\theta_3}{2})} - M_{k-\frac{1}{2},l,m-\frac{1}{2}} e^{i(\frac{\theta_1}{2} - \frac{\theta_3}{2})} \right) \frac{(\delta w_2 - \delta w_1)}{\omega_w} \end{aligned} \right] \\
= & F e^{i(-\frac{\theta_1}{2})} \left[\begin{aligned} & h^{-2} \left(M_{k-\frac{1}{2},l+\frac{1}{2},m} e^{i(\theta_2)} + M_{k-\frac{1}{2},l,m+\frac{1}{2}} e^{i(\theta_3)} + h^2 d_{1,k-\frac{1}{2},l,m} \right) \delta u_0 \\ & + h^{-2} \left(M_{k-\frac{1}{2},l-\frac{1}{2},m} e^{i(-\theta_2)} + M_{k-\frac{1}{2},l,m-\frac{1}{2}} e^{i(-\theta_3)} \right) \delta u_2 \\ & + h^{-2} \left(M_{k-\frac{1}{2},l+\frac{1}{2},m} e^{i(-\frac{\theta_1}{2} + \frac{\theta_2}{2})} - M_{k-\frac{1}{2},l+\frac{1}{2},m} e^{i(\frac{\theta_1}{2} + \frac{\theta_2}{2})} \right) \delta v_0 \\ & + h^{-2} \left(M_{k-\frac{1}{2},l-\frac{1}{2},m} e^{i(\frac{\theta_1}{2} - \frac{\theta_2}{2})} - M_{k-\frac{1}{2},l-\frac{1}{2},m} e^{i(-\frac{\theta_1}{2} - \frac{\theta_2}{2})} \right) \delta v_1 \\ & + h^{-2} \left(M_{k-\frac{1}{2},l,m+\frac{1}{2}} e^{i(-\frac{\theta_1}{2} + \frac{\theta_3}{2})} - M_{k-\frac{1}{2},l,m+\frac{1}{2}} e^{i(\frac{\theta_1}{2} + \frac{\theta_3}{2})} \right) \delta w_0 \\ & + h^{-2} \left(M_{k-\frac{1}{2},l,m-\frac{1}{2}} e^{i(\frac{\theta_1}{2} - \frac{\theta_3}{2})} - M_{k-\frac{1}{2},l,m-\frac{1}{2}} e^{i(-\frac{\theta_1}{2} - \frac{\theta_3}{2})} \right) \delta w_1 \end{aligned} \right]
\end{aligned}$$

with,

$$d_{1,k-\frac{1}{2},l,m} = S_{k-\frac{1}{2},l,m} - h^{-2} \left(M_{k-\frac{1}{2},l+\frac{1}{2},m} + M_{k-\frac{1}{2},l-\frac{1}{2},m} + M_{k-\frac{1}{2},l,m+\frac{1}{2}} + M_{k-\frac{1}{2},l,m-\frac{1}{2}} \right)$$

Second equation,

$$\begin{aligned}
& F e^{i(\frac{\theta_1}{2})} \left[\begin{array}{l} -d_{1,k+\frac{1}{2},l,m} \frac{(\delta u_2 - \delta u_0)}{\omega_u} \\ +h^{-2} M_{k+\frac{1}{2},l+\frac{1}{2},m} \left(e^{i(\frac{\theta_1}{2} + \frac{\theta_2}{2})} + e^{i(-\frac{\theta_1}{2} + \frac{\theta_2}{2})} \right) \frac{(\delta v_1 - \delta v_0)}{\omega_v} \\ +h^{-2} M_{k+\frac{1}{2},l-\frac{1}{2},m} \left(e^{i(-\frac{\theta_1}{2} - \frac{\theta_2}{2})} - e^{i(\frac{\theta_1}{2} - \frac{\theta_2}{2})} \right) \frac{(\delta v_2 - \delta v_1)}{\omega_v} \\ +h^{-2} M_{k+\frac{1}{2},l,m+\frac{1}{2}} \left(e^{i(\frac{\theta_1}{2} + \frac{\theta_3}{2})} - e^{i(-\frac{\theta_1}{2} + \frac{\theta_3}{2})} \right) \frac{(\delta w_1 - \delta w_0)}{\omega_w} \\ +h^{-2} M_{k+\frac{1}{2},l,m-\frac{1}{2}} \left(e^{i(-\frac{\theta_1}{2} - \frac{\theta_3}{2})} - e^{i(\frac{\theta_1}{2} - \frac{\theta_3}{2})} \right) \frac{(\delta w_2 - \delta w_1)}{\omega_w} \end{array} \right] \\
= & F e^{i(\frac{\theta_1}{2})} \left[\begin{array}{l} h^{-2} \left(M_{k+\frac{1}{2},l+\frac{1}{2},m} e^{i(\theta_2)} + M_{k+\frac{1}{2},l,m+\frac{1}{2}} e^{i(\theta_3)} + h^2 d_{1,k+\frac{1}{2},l,m} \right) \partial u_0 \\ +h^{-2} \left(M_{k+\frac{1}{2},l-\frac{1}{2},m} e^{i(-\theta_2)} + M_{k+\frac{1}{2},l,m-\frac{1}{2}} e^{i(-\theta_3)} \right) \partial u_2 \\ +h^{-2} M_{k+\frac{1}{2},l+\frac{1}{2},m} \left(e^{i(-\frac{\theta_1}{2} + \frac{\theta_2}{2})} - e^{i(\frac{\theta_1}{2} + \frac{\theta_2}{2})} \right) \partial v_0 \\ +h^{-2} M_{k+\frac{1}{2},l-\frac{1}{2},m} \left(e^{i(\frac{\theta_1}{2} - \frac{\theta_2}{2})} - e^{i(-\frac{\theta_1}{2} - \frac{\theta_2}{2})} \right) \partial v_1 \\ +h^{-2} M_{k+\frac{1}{2},l,m+\frac{1}{2}} \left(e^{i(-\frac{\theta_1}{2} + \frac{\theta_3}{2})} - e^{i(\frac{\theta_1}{2} + \frac{\theta_3}{2})} \right) \partial w_0 \\ +h^{-2} M_{k+\frac{1}{2},l,m-\frac{1}{2}} \left(e^{i(\frac{\theta_1}{2} - \frac{\theta_3}{2})} - e^{i(-\frac{\theta_1}{2} - \frac{\theta_3}{2})} \right) \partial w_1 \end{array} \right]
\end{aligned}$$

with,

$$d_{1,k+\frac{1}{2},l,m} = S_{k+\frac{1}{2},l,m} - h^{-2} \left(M_{k+\frac{1}{2},l+\frac{1}{2},m} + M_{k+\frac{1}{2},l-\frac{1}{2},m} + M_{k+\frac{1}{2},l,m+\frac{1}{2}} + M_{k+\frac{1}{2},l,m-\frac{1}{2}} \right)$$

Third equation,

$$\begin{aligned}
& F e^{i(-\frac{\theta_2}{2})} \left[\begin{aligned}
& +h^{-2} \left(M_{k+\frac{1}{2}, l-\frac{1}{2}, m} e^{i(\frac{\theta_1}{2} + \frac{\theta_2}{2})} - M_{k-\frac{1}{2}, l-\frac{1}{2}, m} e^{i(-\frac{\theta_1}{2} + \frac{\theta_2}{2})} \right) \frac{(\delta u_2 - \delta u_0)}{\omega_u} \\
& -h^{-2} \left(M_{k+\frac{1}{2}, l-\frac{1}{2}, m} e^{i(\theta_1)} + M_{k-\frac{1}{2}, l-\frac{1}{2}, m} e^{i(-\theta_1)} + h^2 d_{2,k, l-\frac{1}{2}, m} \right) \frac{(\delta v_2 - \delta v_1)}{\omega_u} \\
& \quad + h^{-2} M_{k, l-\frac{1}{2}, m+\frac{1}{2}} e^{i(\frac{\theta_2}{2} + \frac{\theta_3}{2})} \frac{(\delta w_1 - \delta w_0)}{\omega_w} \\
& \quad - h^{-2} M_{k, l-\frac{1}{2}, m-\frac{1}{2}} e^{i(\frac{\theta_2}{2} - \frac{\theta_3}{2})} \frac{(\delta w_2 - \delta w_1)}{\omega_w}
\end{aligned} \right] \\
= & F e^{i(-\frac{\theta_2}{2})} \left[\begin{aligned}
& +h^{-2} \left(M_{k-\frac{1}{2}, l-\frac{1}{2}, m} e^{i(-\frac{\theta_1}{2} + \frac{\theta_2}{2})} - M_{k+\frac{1}{2}, l-\frac{1}{2}, m} e^{i(\frac{\theta_1}{2} + \frac{\theta_2}{2})} \right) \partial u_0 \\
& +h^{-2} \left(M_{k+\frac{1}{2}, l-\frac{1}{2}, m} e^{i(\frac{\theta_1}{2} - \frac{\theta_2}{2})} - M_{k-\frac{1}{2}, l-\frac{1}{2}, m} e^{i(-\frac{\theta_1}{2} - \frac{\theta_2}{2})} \right) \partial u_2 \\
& \quad + h^{-2} M_{k, l-\frac{1}{2}, m+\frac{1}{2}} e^{i(\theta_3)} \partial v_0 \\
& +h^{-2} \left(M_{k+\frac{1}{2}, l-\frac{1}{2}, m} e^{i(\theta_1)} + M_{k-\frac{1}{2}, l-\frac{1}{2}, m} e^{i(-\theta_1)} + h^2 d_{2,k, l-\frac{1}{2}, m} \right) \partial v_1 \\
& \quad + h^{-2} M_{k, l-\frac{1}{2}, m-\frac{1}{2}} e^{i(-\theta_3)} \partial v_2 \\
& \quad - h^{-2} M_{k, l-\frac{1}{2}, m+\frac{1}{2}} e^{i(\frac{\theta_2}{2} + \frac{\theta_3}{2})} \partial w_0 \\
& +h^{-2} \left(M_{k, l-\frac{1}{2}, m-\frac{1}{2}} e^{i(\frac{\theta_2}{2} - \frac{\theta_3}{2})} + M_{k, l-\frac{1}{2}, m+\frac{1}{2}} e^{i(-\frac{\theta_2}{2} + \frac{\theta_3}{2})} \right) \partial w_1 \\
& \quad - h^{-2} M_{k, l-\frac{1}{2}, m-\frac{1}{2}} e^{i(-\frac{\theta_2}{2} - \frac{\theta_3}{2})} \partial w_2
\end{aligned} \right]
\end{aligned}$$

with,

$$d_{2,k, l-\frac{1}{2}, m} = S_{k, l-\frac{1}{2}, m} - h^{-2} \left(M_{k, l-\frac{1}{2}, m+\frac{1}{2}} + M_{k, l-\frac{1}{2}, m-\frac{1}{2}} + M_{k+\frac{1}{2}, l-\frac{1}{2}, m} + M_{k-\frac{1}{2}, l-\frac{1}{2}, m} \right)$$

Fourth equation,

$$\begin{aligned}
& F e^{i(\frac{\theta_2}{2})} \left[\begin{aligned} & h^2 \left(M_{k-\frac{1}{2}, l+\frac{1}{2}, m} e^{i(-\frac{\theta_1}{2}-\frac{\theta_2}{2})} - M_{k+\frac{1}{2}, l+\frac{1}{2}, m} e^{i(\frac{\theta_1}{2}-\frac{\theta_2}{2})} \right) \frac{(\delta u_2 - \delta u_0)}{\omega_u} \\ & - h^{-2} \left(M_{k+\frac{1}{2}, l+\frac{1}{2}, m} e^{i(\theta_1)} + M_{k-\frac{1}{2}, l+\frac{1}{2}, m} e^{i(-\theta_1)} + d_{2,k, l+\frac{1}{2}, m} \right) \frac{(\delta v_1 - \delta v_0)}{\omega_v} \\ & \quad - h^{-2} M_{k, l+\frac{1}{2}, m+\frac{1}{2}} e^{i(-\frac{\theta_2}{2}+\frac{\theta_3}{2})} \frac{(\delta w_1 - \delta w_0)}{\omega_w} \\ & \quad + h^{-2} M_{k, l+\frac{1}{2}, m-\frac{1}{2}} e^{i(-\frac{\theta_2}{2}-\frac{\theta_3}{2})} \frac{(\delta w_2 - \delta w_1)}{\omega_w} \end{aligned} \right] \\
= & F e^{i(\frac{\theta_2}{2})} \left[\begin{aligned} & h^2 \left(\begin{aligned} & M_{k+\frac{1}{2}, l+\frac{1}{2}, m} \left(e^{i(\frac{\theta_1}{2}-\frac{\theta_2}{2})} - e^{i(\frac{\theta_1}{2}+\frac{\theta_2}{2})} \right) \\ & + M_{k-\frac{1}{2}, l+\frac{1}{2}, m} \left(e^{i(-\frac{\theta_1}{2}+\frac{\theta_2}{2})} - e^{i(-\frac{\theta_1}{2}-\frac{\theta_2}{2})} \right) \end{aligned} \right) \partial u_0 \\ & + h^{-2} \left(\begin{aligned} & M_{k, l+\frac{1}{2}, m+\frac{1}{2}} e^{i(\theta_3)} + M_{k+\frac{1}{2}, l+\frac{1}{2}, m} e^{i(\theta_1)} \\ & + M_{k-\frac{1}{2}, l+\frac{1}{2}, m} e^{i(-\theta_1)} + h^2 d_{2,k, l+\frac{1}{2}, m} \end{aligned} \right) \partial v_0 \\ & \quad + h^{-2} M_{k, l+\frac{1}{2}, m-\frac{1}{2}} e^{i(-\theta_3)} \partial v_2 \\ & \quad + h^{-2} M_{k, l+\frac{1}{2}, m+\frac{1}{2}} \left(e^{i(-\frac{\theta_2}{2}+\frac{\theta_3}{2})} - e^{i(\frac{\theta_2}{2}+\frac{\theta_3}{2})} \right) \partial w_0 \\ & \quad + h^{-2} M_{k, l+\frac{1}{2}, m-\frac{1}{2}} \left(e^{i(\frac{\theta_2}{2}-\frac{\theta_3}{2})} - e^{i(-\frac{\theta_2}{2}-\frac{\theta_3}{2})} \right) \partial w_1 \end{aligned} \right]
\end{aligned}$$

with,

$$d_{2,k, l+\frac{1}{2}, m} = S_{k, l+\frac{1}{2}, m} - h^{-2} \left(M_{k, l+\frac{1}{2}, m+\frac{1}{2}} + M_{k, l+\frac{1}{2}, m-\frac{1}{2}} + M_{k+\frac{1}{2}, l+\frac{1}{2}, m} + M_{k-\frac{1}{2}, l+\frac{1}{2}, m} \right)$$

Fifth equation,

$$\begin{aligned}
& F e^{i(-\frac{\theta_3}{2})} \left[\begin{aligned}
& +h^{-2} \left(M_{k+\frac{1}{2},l,m-\frac{1}{2}} e^{i(\frac{\theta_1}{2}+\frac{\theta_3}{2})} - M_{k-\frac{1}{2},l,m-\frac{1}{2}} e^{i(-\frac{\theta_1}{2}+\frac{\theta_3}{2})} \right) \frac{(\delta u_2 - \delta u_0)}{\omega_u} \\
& + h^{-2} M_{k,l+\frac{1}{2},m-\frac{1}{2}} e^{i(\frac{\theta_2}{2}+\frac{\theta_3}{2})} \frac{(\delta v_1 - \delta v_0)}{\omega_v} \\
& - h^{-2} M_{k,l-\frac{1}{2},m-\frac{1}{2}} e^{i(-\frac{\theta_2}{2}+\frac{\theta_3}{2})} \frac{(\delta v_2 - \delta v_1)}{\omega_v} \\
& - h^{-2} \left(M_{k-\frac{1}{2},l,m-\frac{1}{2}} e^{i(-\theta_1)} + M_{k+\frac{1}{2},l,m-\frac{1}{2}} e^{i(\theta_1)} + h^2 d_{3,k,l,m-\frac{1}{2}} \right) \frac{(\delta w_2 - \delta w_1)}{\omega_w}
\end{aligned} \right] \\
= & F e^{i(-\frac{\theta_3}{2})} \left[\begin{aligned}
& h^{-2} \left(M_{k-\frac{1}{2},l,m-\frac{1}{2}} e^{i(-\frac{\theta_1}{2}+\frac{\theta_3}{2})} - M_{k+\frac{1}{2},l,m-\frac{1}{2}} e^{i(\frac{\theta_1}{2}+\frac{\theta_3}{2})} \right) \partial u_0 \\
& + h^{-2} \left(M_{k+\frac{1}{2},l,m-\frac{1}{2}} e^{i(\frac{\theta_1}{2}-\frac{\theta_3}{2})} - M_{k-\frac{1}{2},l,m-\frac{1}{2}} e^{i(-\frac{\theta_1}{2}-\frac{\theta_3}{2})} \right) \partial u_2 \\
& - h^{-2} M_{k,l+\frac{1}{2},m-\frac{1}{2}} e^{i(\frac{\theta_2}{2}+\frac{\theta_3}{2})} \partial v_0 \\
& + h^{-2} M_{k,l-\frac{1}{2},m-\frac{1}{2}} e^{i(-\frac{\theta_2}{2}+\frac{\theta_3}{2})} \partial v_1 \\
& + h^{-2} \left(M_{k,l+\frac{1}{2},m-\frac{1}{2}} e^{i(\frac{\theta_2}{2}-\frac{\theta_3}{2})} - M_{k,l-\frac{1}{2},m-\frac{1}{2}} e^{i(-\frac{\theta_2}{2}-\frac{\theta_3}{2})} \right) \partial v_2 \\
& + h^{-2} \left(\begin{aligned}
& M_{k-\frac{1}{2},l,m-\frac{1}{2}} e^{i(-\theta_1)} + M_{k+\frac{1}{2},l,m-\frac{1}{2}} e^{i(\theta_1)} \\
& + M_{k,l+\frac{1}{2},m-\frac{1}{2}} e^{i(\theta_2)} + h^2 d_{3,k,l,m-\frac{1}{2}} \\
& + h^{-2} M_{k,l-\frac{1}{2},m-\frac{1}{2}} e^{i(-\theta_3)} \partial w_2
\end{aligned} \right) \partial w_1
\end{aligned} \right]
\end{aligned}$$

with,

$$d_{3,k,l,m-\frac{1}{2}} = S_{k,l,m-\frac{1}{2}} - h^{-2} \left(M_{k+\frac{1}{2},l,m-\frac{1}{2}} + M_{k-\frac{1}{2},l,m-\frac{1}{2}} + M_{k,l+\frac{1}{2},m-\frac{1}{2}} + M_{k,l-\frac{1}{2},m-\frac{1}{2}} \right)$$

Sixth equation,

$$\begin{aligned}
& F e^{i(\frac{\theta_3}{2})} \left[\begin{aligned} & h^{-2} \left(M_{k-\frac{1}{2},l,m+\frac{1}{2}} e^{i(-\frac{\theta_1}{2}-\frac{\theta_3}{2})} - M_{k+\frac{1}{2},l,m+\frac{1}{2}} e^{i(\frac{\theta_1}{2}-\frac{\theta_3}{2})} \right) \frac{(\delta u_2 - \delta u_0)}{\omega_u} \\ & - h^{-2} M_{k,l+\frac{1}{2},m+\frac{1}{2}} e^{i(\frac{\theta_2}{2}-\frac{\theta_3}{2})} \frac{(\delta v_1 - \delta v_0)}{\omega_v} \\ & + h^{-2} M_{k,l-\frac{1}{2},m+\frac{1}{2}} e^{i(-\frac{\theta_2}{2}-\frac{\theta_3}{2})} \frac{(\delta v_2 - \delta v_1)}{\omega_v} \\ & - h^{-2} \left(M_{k-\frac{1}{2},l,m+\frac{1}{2}} e^{i(-\theta_1)} + M_{k+\frac{1}{2},l,m+\frac{1}{2}} e^{i(\theta_1)} + h^2 d_{3,k,l,m+\frac{1}{2}} \right) \frac{(\delta w_1 - \delta w_0)}{\omega_w} \end{aligned} \right] \\
= & F e^{i(\frac{\theta_3}{2})} \left[\begin{aligned} & h^{-2} \left(\begin{aligned} & M_{k+\frac{1}{2},l,m+\frac{1}{2}} \left(e^{i(\frac{\theta_1}{2}-\frac{\theta_3}{2})} - e^{i(\frac{\theta_1}{2}+\frac{\theta_3}{2})} \right) \\ & + M_{k-\frac{1}{2},l,m+\frac{1}{2}} \left(e^{i(-\frac{\theta_1}{2}+\frac{\theta_3}{2})} - e^{i(-\frac{\theta_1}{2}-\frac{\theta_3}{2})} \right) \end{aligned} \right) \partial u_0 \\ & + h^{-2} \left(\begin{aligned} & M_{k,l+\frac{1}{2},m+\frac{1}{2}} \left(e^{i(\frac{\theta_2}{2}-\frac{\theta_3}{2})} - e^{i(\frac{\theta_2}{2}+\frac{\theta_3}{2})} \right) \\ & + M_{k,l-\frac{1}{2},m+\frac{1}{2}} e^{i(-\frac{\theta_2}{2}+\frac{\theta_3}{2})} \end{aligned} \right) \partial v_0 \\ & - h^{-2} M_{k,l-\frac{1}{2},m+\frac{1}{2}} e^{i(-\frac{\theta_2}{2}-\frac{\theta_3}{2})} \partial v_1 \\ & + h^{-2} \left(\begin{aligned} & M_{k-\frac{1}{2},l,m+\frac{1}{2}} e^{i(-\theta_1)} + M_{k+\frac{1}{2},l,m+\frac{1}{2}} e^{i(\theta_1)} \\ & + M_{k,l+\frac{1}{2},m+\frac{1}{2}} e^{i(\theta_2)} + h^2 d_{3,k,l,m+\frac{1}{2}} \end{aligned} \right) \partial w_0 \\ & + h^{-2} M_{k,l-\frac{1}{2},m+\frac{1}{2}} e^{i(-\theta_2)} \partial w_1 \end{aligned} \right]
\end{aligned}$$

with,

$$d_{3,k,l,m+\frac{1}{2}} = S_{k,l,m+\frac{1}{2}} - h^{-2} \left(M_{k+\frac{1}{2},l,m+\frac{1}{2}} + M_{k-\frac{1}{2},l,m+\frac{1}{2}} + M_{k,l+\frac{1}{2},m+\frac{1}{2}} + M_{k,l-\frac{1}{2},m+\frac{1}{2}} \right)$$

D BiCGStab2

Here the algorithm of preconditioned `bicgstab2` is given.

The problem is to find the solution \mathbf{e} that leads to a zero residual $\mathbf{r} = \mathbf{s} - L\mathbf{e}$. The preconditioner is M_{pre} .

Compute the residual $\mathbf{r}^0 = \mathbf{s} - L\mathbf{e}^0$ for a starting value \mathbf{e}^0 . Set $\mathbf{p}^0 = \mathbf{r}^0$. Choose an arbitrary $\tilde{\mathbf{r}}^0$ such that $\langle \mathbf{r}^0, \tilde{\mathbf{r}}^0 \rangle \neq 0$, for instance $\tilde{\mathbf{r}}^0 = \mathbf{r}^0$. Set $\rho^0 = 1$ and $\xi^0 = 1$.

For $i = 0, 1, \dots$ until convergence do:

$$\begin{aligned}
 \hat{\mathbf{p}}^i &= M_{pre}^{-1} \mathbf{p}^i, \\
 \mathbf{v}^i &= L \hat{\mathbf{p}}^i, \\
 \rho^{i+1} &= (\tilde{\mathbf{r}}^0)^H \mathbf{r}^i \\
 \alpha^i &= \frac{\rho^{i+1}}{(\tilde{\mathbf{r}}^0)^H \mathbf{v}^i}, \\
 \mathbf{e}^{i+\frac{1}{2}} &= \mathbf{e}^i + \alpha^i \mathbf{p}^i \\
 \mathbf{d}^i &= \mathbf{r}^i - \alpha^i \mathbf{v}^i \\
 \hat{\mathbf{d}}^i &= M_{pre}^{-1} \mathbf{d}^i, \\
 \mathbf{w}^i &= L \hat{\mathbf{d}}^i, \\
 \xi^i &= \frac{(\mathbf{w}^i)^H \mathbf{d}^i}{(\mathbf{w}^i)^H \mathbf{w}^i}, \\
 \mathbf{e}^{i+1} &= \mathbf{e}^{i+\frac{1}{2}} + \xi^i \hat{\mathbf{d}}^i, \\
 \mathbf{r}^{i+1} &= \mathbf{d}^i - \xi^i \mathbf{w}^i, \\
 \beta^i &= \frac{\alpha^i \rho^{i+1}}{\xi^i \rho^i}, \\
 \mathbf{p}^{i+1} &= \mathbf{r}^i + \beta^i (\mathbf{p}^i - \xi^i \mathbf{v}^i).
 \end{aligned}$$

Here \mathbf{a}^H denotes the conjugate transpose of \mathbf{a} , so that $\mathbf{a}^H \mathbf{b}$ is a scalar product. This product is the conjugate of the usual complex scalar product $\langle \mathbf{a}, \mathbf{b} \rangle$. Convergence checks are carried out once $\mathbf{e}_{i+\frac{1}{2}}$ or \mathbf{e}_{i+1} become available by comparing the norm of $\mathbf{s} - L\mathbf{e}$ to the norm of \mathbf{s} times the convergence tolerance.

The algorithm requires two calls to the multigrid solver (M_{pre}^{-1}). The computations of the form $L\mathbf{e}$ can be carried out matrix-free by continuously evaluating the spatial discretization. Together with the residual evaluations required for the convergence checks, these add to the cost of the two multigrid iterations. Nevertheless, one full `bicgstab2` iterations is counted as two iterations when comparing to a pure multigrid iteration. Convergence checks are carried out after each solution update at the expense of an additional residual computation.

E Power-law Grid stretching

Here two types of grid stretching are considered, based on a power-law and on the hyperbolic cosine function. Here grid stretching based on power-law is discussed first.

The grid starts at $x = x_{\min}$ and ends at $x = x_{\max}$. The grid is stretched with respect to some reference point x_0 inside the domain. There are n_L cells to the left of x_0 and n_r to the right, with a total of $N_x = n_L + n_r$ cells.

The grid spacings or cell widths are given by,

$$\begin{aligned} h_k &= ha^{(n_L-1-k)}, \quad \text{for } k = 0, \dots, n_L - 1, \\ h_k &= ha^{(k-n_L)}, \quad \text{for } k = n_L, \dots, N_x - 1. \end{aligned}$$

Here $a = 1 + \alpha$. After summation the following expression is obtained,

$$\frac{a^{n_R} - 1}{a - 1} = (x_{\max} - x_0)/h, \quad \frac{a^{n_L} - 1}{a - 1} = (x_0 - x_{\min}) - h.$$

To determine n_L and n_R , determine the smallest integer n_L that minimizes

$$\left| \frac{a^{(N_x - n_L)} - 1}{a^{n_L} - 1} - r \right|, \quad r = \frac{(x_{\max} - x_0)}{(x_0 - x_{\min})} \quad (34)$$

Next, h is determined such that $\sum_{k=0}^{N_x-1} h_k = x_{\max} - x_{\min}$.

For the hyperbolic cosine function, the grid spacing obeys $h_k = h_0 \cosh[b(k - k_0)]$, $k = 0, \dots, N_x - 1$. To determine k_0 , which is not necessarily an integer, evaluate

$$\begin{aligned} \xi &= \frac{(1+r)[e^b - 1] + \sqrt{\zeta}}{2e^b[r + e^{(-bN_x)}]}, \\ \zeta &= 2e^b \left[(1+r)^2 \cosh(a) + 4r \cosh(bN_x) + (1-r)^2 \right], \end{aligned}$$

to find $k_0 = \frac{\log(\xi)}{b}$. Here r is the same as in equation (34). The parameter b can be chosen in advance, or determined iteratively such that the maximum ratio of cell widths for neighboring cells does not exceed $1 + \alpha$, or such that the minimum grid spacing equals a given value.

F L2 norm

The iterations of the multigrid solver were stopped when the l_2 -norm had dropped by a factor 10^{-8} from its original value for a zero solution.

Here l_{\max} is the maximum difference with the exact solution. The error for E_1 is measured by,

$$l_2(E_1) = \left[\sum_{k=0}^{N_x-1} \sum_{l=0}^{N_y} \sum_{m=0}^{N_z} \left| E_{1,k+\frac{1}{2},l,m} - E_{1,k+\frac{1}{2},l,m}^{exact} \right|^2 V_{k+\frac{1}{2},l,m} \right]^{1/2}$$

and similarly for components E_2 and E_3 . Here $V_{k+\frac{1}{2},l,m}$ is defined as in equation 6. The maximum error is,

$$l_{\max}(E_1) = \max_{k=0,\dots,N_x-1} \max_{l=0,\dots,N_y} \max_{m=0,\dots,N_z} \left| E_{1,k+\frac{1}{2},l,m} - E_{1,k+\frac{1}{2},l,m}^{exact} \right|.$$

The tables of results lists the maximum over *components* for l_{\max} and $l_2 = \sqrt{\sum_{k=1}^3 l_2(E_k)^2}$.

G Matlab

All the Matlab functions can be found on the CD-ROM enclosed with this thesis. For future research, in the sections below a short list of functions can be found and a schematic representation of a recursive call of the F-cycle in the Matlab multigrid solver.

G.1 Implementation Matlab code

Here the vectors, matrices and scalars are given which are used frequently in the Matlab code.

<code>clight0</code>	scalar, speed of light
<code>mu0</code>	scalar, from $i*\omega*\mu_0*\sigma*$...
<code>eps0</code>	scalar, from $\epsilon=\epsilon_0*\epsilon_{pr}$
<code>maxit</code>	scalar, maximum number of iterations
<code>om</code>	scalar, omega
<code>ex1a</code>	scalar, values 0 or 1, determines model
<code>imodel</code>	scalar, determines case in model of ex1a
<code>epsconv</code>	scalar, convergence tolerance level
<code>imethod</code>	scalar, determines type of multigrid solver
<code>mmm</code>	scalar, determines number of cells in each direction
<code>alpha</code>	scalar, amount of grid stretching
<code>xmin,xmax,...</code>	scalars, determine boundaries of domain
<code>nx,ny,nz</code>	scalar, is number of cells in each direction, $=2^{mmm+1}$
<code>xs,ys,zs</code>	scalar, placement of the source
<code>iprint</code>	scalar, 0 or 1 for printing value of variables during evaluation
<code>use_bnd</code>	scalar, 0 or 1 for adapting boundaries with PEC
<code>xx,yy,zz</code>	vectors, contain grid nodes, values from xmin to xmax with distances dx,dy,dz between each node
<code>g</code>	matrix, contains grid
<code>s1,s2,s3</code>	3D matrices, contain source term, $\dim(s1)=(nx-1,ny,nz)$, $\dim(s2)=(nx,ny-1,nz)$, $\dim(s3)=(nx,ny,nz-1)$
<code>efac</code>	3D matrix, $=(i*om*\mu_0)*(\sigma-(i*om*\epsilon_0)*\epsilon_{pr})$ $\dim(efac)=\dim(\sigma)=\dim(\epsilon_{pr})=(nx-1,ny-1,nz-1)$. Values σ , ϵ_{pr} depend on grid position. Due to PEC <code>efac</code> not defined on boundaries
<code>emur</code>	3D matrix, $=\text{volume}/\text{mur}$, mur results from model description
<code>e1,e2,e3</code>	3D matrices, contain solution of Maxwell equation $\dim(e1)=(nx-1,ny,nz)$, $\dim(e2)=(nx,ny-1,nz)$, $\dim(e3)=(nx,ny,nz-1)$
<code>eef</code>	vector, contains all components of e1, e2 and e3
<code>src</code>	vector, contains all components of s1, s2 and s3

G.2 Recursive call Matlab multigrid solver: e3db_msol1

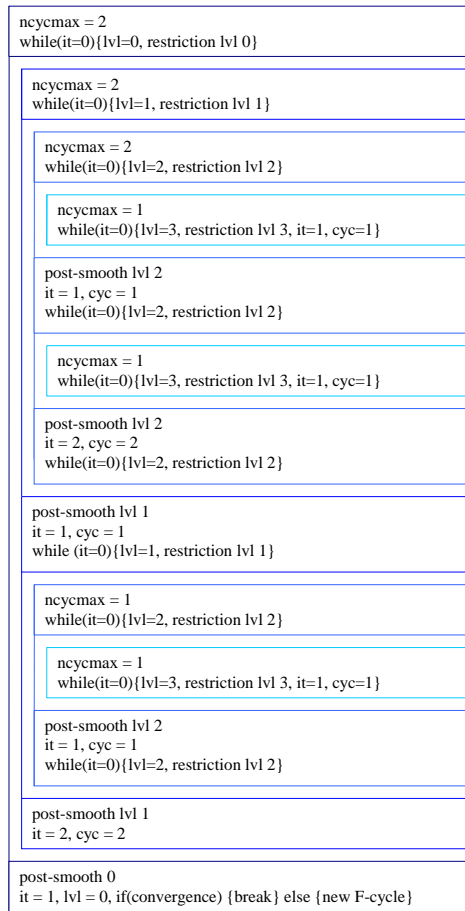


Figure 29: Example of recursive call of F-cycle for Matlab Multigrid solver, e3db_msol1. On level 0, $N_x = 17$ and level 3, $N_x = 3$.

References

- [1] R.A. Adams, Calculus, *Addison Wesley*, 2001
- [2] F. Aminzadeh, J.Brac and T. Kunz, 3-D Salt and overthrust models, *Society of Exploration Geophysicists, Tulsa, Oklahoma*, 1997
- [3] D.N. Arnold, R.S. Falk, and R. Winther, Multigrid in $H(\text{div})$ and $H(\text{curl})$. *Numerische Mathematik*, **85**, 197–217, 2000.
- [4] D.A. Aruliah, U.M. Ascher, E. Haber, and D. Oldenburg, A method for the forward modelling of 3D electromagnetic quasi-static problems. *Mathematical Models and Methods in Applied Sciences*, 2001
- [5] M. Clemens and T. Weiland, Discrete electromagnetism with the Finite Integration Technique. *Progress in Electromagnetic Research (PIER)*, **32**, 65–87, 2001.
- [6] S. Feigh, M. Clemens and T. Weiland, Geometric multigrid method for Electro- and Magnetostatic field simulation using the conformal Finite Integration Technique, *Copper Mountain Conference on Multigrid Methods*, 2003
- [7] H.H. Gutknecht, Variants of BiCGStab for matrices with complex spectrum. *SIAM Journal on Scientific and Statistical Computing*, **14**, 1020–1033, 1993.
- [8] R. Hiptmair, Multigrid for Maxwell’s equations, *SIAM J. Num. Anal.* **36**: 204-225, 1998.
- [9] M.A. Meju, L.A. Gallardo and A.K. Mohamed, Evidence for correlation of electrical resistivity and seismic velocity in heterogeneous near-surface material, *Geophys. Res. Letters*, **30** (7):1373-1376, 2003.
- [10] P. Monk and E. Süli, A convergence analysis of Yee’s scheme on nonuniform grids. *SIAM Journal of numerical analysis*, 1994
- [11] W.A. Mulder, A new multigrid approach to convection problems. *Journal of Computational Physics*, **83**, 303–323, 1989.
- [12] W.A. Mulder. A multigrid solver for 3-D electromagnetic diffusion. *Geophysical Prospecting*, in press 2006.
- [13] W.A. Mulder. Geophysical modelling of 3D electromagnetic diffusion with multigrid. *Computing and Visualization in Science*, to appear 2006.
- [14] G. Pöplau and U. van Rienen, Multigrid solvers for Poisson ’s Equation in computational electromagnetics. *Scientific Computing in Electrical Engineering* **18**, 169-176, 2001

- [15] T.M. Shah, Analysis of the multigrid method, *Thesis submitted for degree of Doctor of Philosophy at Oxford University*, 1989
- [16] C.A. Thole and U. Trottenberg, Basic smoothing procedures for the multigrid treatment of elliptic 3-D operators. *Appl. Math. Comp.* 19: 333–345, 1986.
- [17] U. Trottenberg, C.W. Oosterlee and A. Schüller: *Multigrid*, Academic Press, London (2000).
- [18] M. Unsworth, New developments in conventional hydrocarbon exploration with electromagnetic methods. *CSEG Recorder*, 2005
- [19] H.A. van der Vorst, BI-CGSTAB: a fast and smoothly converging variant of bi-CG for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, **13**, 631–644, 1992.
- [20] T. Washio and C.W. Oosterlee, Flexible multiple semicoarsening for three-dimensional singularly perturbed problems. *SIAM J. Sci. Comput.* 19: 1646–1666, 1998.
- [21] T. Weiland. A discretization method for the solution of Maxwell’s equations for six-components fields. *Electronics and Communications*, **31**, 116–120, 1977.
- [22] G. WITTUM, Multi-grid methods for Stokes and Navier-Stokes equations with transforming smoothers: algorithms and numerical results, *Num. Math.*, 54: 543-563, 1989.
- [23] Wolfram Research, <http://mathworld.wolfram.com/>, 2006
- [24] K. Yee, Numerical solution of initial boundary value problems involving Maxwell’s equations in isotropic media. *IEEE Trans. Antennas and Propagation*, **16**, 302–307, 1966.