Multigrid methods

Written by Marcus Greiff,
coauthored by Axel Södersten

6/17/2016

1 Introduction

In this project a Multigrid scheme is implemented for both the Poisson and the Helmholtz equation. As an example, the solver is the used to simulate the wave equation with homogenous boundary conditions on the unit square. In the first section, we describe the theory of the Multigrid (MG) method in one dimension and then proceed by extending the theory to 2D. The MG-solver is then applied to the wave equation, which is solved by implicit time stepping to avoid breaking the CFL-condition.

2 2D - Helmholtz Equation

In this part of the project, a MG-solver is derived for the Helmholtz equation on the unit square

$$u - \beta \Delta u = f, \quad (x,y) \in \Omega \tag{1}$$

$$u(x,y) = 0, \qquad (x,y) \in \partial\Omega \tag{2}$$

where $\Omega = \{(x,y) : 0 \leq (x,y) \leq 1\}$ denotes the computational domain. The problem is discretised using a FDM five-point stencil on an equidistant grid, such that $h = \Delta x = \Delta y$, resulting in the discrete equation

$$u_{i,j} - \beta \frac{u_{i-1,j} + u_{i,j-1} - 4u_{i,j} + u_{i+1,j} + u_{i,j+1}}{h^2} = f_{i,j} \tag{3}$$

where we denote the matrices $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{n \times n}$. As we are only interested in solutions satisfying the boundary condition (2), we see that all boundary points $u_{1,\bullet} = u_{n,\bullet} = u_{\bullet,1} = u_{\bullet,n} = 0$. Consequently, we only solve the problem for the internal grid points, a total of $N^2 = (n-2)^2$ computational points in the 2D-case. If we instead define the discretised variables in terms of the internal grid points and adopt a vector notation instead, such that $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{N^2 \times 1}$ with $u_{i,j} = u_{i+(j-1)N}$, the discrete equation can be written

$$\hat{\mathbf{T}}\mathbf{u} = \mathbf{f} \tag{4}$$

where the differential operator becomes a Toeplitz matrix on the form

$$\hat{\mathbf{T}} = \mathbf{I} \otimes \mathbf{I} - \beta(\mathbf{I} \otimes \mathbf{T}_1 + \mathbf{T}_1 \otimes \mathbf{I}) \in \mathbb{R}^{N^2 \times N^2} \tag{5}$$

with $\mathbf{T}_1 \in \mathbb{R}^{N \times N}$ being the second order FDM discretsation of the 1D Laplace operator; a tridiagonal matrix with $-2/h^2$ on the diagonal and $1/h^2$ on the super- and sub-diagonals.

As a motivating example for the MG-solver, the problem can now be solved algebraically but if the number of internal points in the discretisation is denoted N, then $\dim(\hat{\mathbf{T}}) = N^2$. This puts great constraints on how fine we may make the discretisation due to the computational cost required to solve the linear system. Instead, we wish to create an iterative method which never has to solve the system with the complete differential operator, but rather on a coarse grid where the dimensions of the $\hat{\mathbf{T}}$ operator used to solve the system depends on the number of grids used. For example, if using a total number of n_{grid} grids in the MG iteration, the system which finally needs to be solved involves an square matrix with the dimensions $N^2/2^{n_{grid}-1}$ on every iteration.

An even better solution would be never to solve the linear system exactly at any grid level but instead use a cost-effective inexact method to correct the error at each iteration, which can be done with the Jacobi Method (JM) and the Gauss-Seidel Method (GSM). In practice, solving a large system with either of these iterative methods is infeasible, as the rate of convergence is incredibly slow for low frequencies where the spectral radius of the error recursion matrix $\mathbf{P}_j$, $\rho[\mathbf{P}_j] \to 1$ for small Δx. When using a damped Jacobi recursion, the damping factor γ can be set so as to suppress high frequent error, and is therefore included in the MG-solver to smooth out the error on each iteration. This is of great interest, as moving to coarser grids effectively halves the Nyquist frequency, and the un-smoothed error with high frequency content will be aliased and corrupt the spectral content of the error.

From the discrete equation (5), the Jacobi method can be derived by splitting the discrete operator into a diagonal part $\mathbf{D}$, as well as upper and lower triangle parts $\mathbf{U}$ and $\mathbf{L}$

$$\hat{\mathbf{T}} = \mathbf{D} - \mathbf{L} - \mathbf{U} \tag{6}$$

The discrete equation can then be written as

$$(\mathbf{D} - \mathbf{L} - \mathbf{U})\mathbf{u} = \mathbf{f} \Leftrightarrow \mathbf{u} = \mathbf{D^{-1}}(\mathbf{L} + \mathbf{U})\mathbf{u} + \mathbf{D}^{-1}\mathbf{f} \overset{def}{=} \mathbf{P}_j\mathbf{u} + \mathbf{D}^{-1}\mathbf{f} \tag{7}$$

where the iteration matrix $\mathbf{P}_j$ exists as long as $\mathbf{D}$ has a non-zero diagonal and is invertible, which is the case if

$$1 + \frac{\beta}{h^2}4 \neq 0 \Rightarrow \beta \neq -\frac{h^2}{4} \tag{8}$$

which holds if we impose the condition $\beta > 0$, which is general enough for our purposes.

Using fixed point iterations, the scheme becomes

$$\mathbf{u}^{k+1} = \mathbf{P}_j\mathbf{u}^k + \mathbf{D^{-1}}\mathbf{f}. \tag{9}$$

and if we instead look at the error recursion, with the analytical solution $\mathbf{u}^*$, we see that

$$\begin{align}
\mathbf{e}^{k+1} &= \mathbf{u}^{k+1} - \mathbf{u}^* \tag{10}\\
&= \mathbf{P}_j\mathbf{u}^k + \mathbf{D}^{-1}\mathbf{f} - \mathbf{u}^* \tag{11}\\
&= \mathbf{P}_j\mathbf{u}^k + \mathbf{D}^{-1}\mathbf{f} - \mathbf{u}^* + (\mathbf{P}_j - \mathbf{P}_j)\mathbf{u}^* \tag{12}\\
&= \mathbf{P}_j(\mathbf{u}^k - \mathbf{u}^*) + \mathbf{D}^{-1}\mathbf{f} - \mathbf{u}^* + \mathbf{P}_j\mathbf{u}^* \tag{13}\\
&= \mathbf{P}_j\mathbf{e}^k + \mathbf{D}^{-1}(\mathbf{D} - \mathbf{L} - \mathbf{U})\mathbf{u}^* - \mathbf{u}^* + \mathbf{P}_j\mathbf{u}^* \tag{14}\\
&= \mathbf{P}_j\mathbf{e}^k + \mathbf{u}^* - \mathbf{u}^* - \mathbf{P}_j\mathbf{u}^* + \mathbf{P}_j\mathbf{u}^* \tag{15}\\
&= \mathbf{P}_j\mathbf{e}^k \tag{16}\\
& \tag{17}
\end{align}$$

This implies that we need to set up and compute the $\mathbf{P}_j \in \mathbf{R}^{N^2 \times N^2}$ on every iteration, which is computationally inefficient. If we instead define the residual at iteration k as

$$\mathbf{r}^k = \hat{\mathbf{T}}\mathbf{u}^k - \mathbf{f} \tag{18}$$

we get the error recursion

$$\begin{aligned}
\mathbf{e}^{k+1} &= \mathbf{u}^{k+1} - \mathbf{u}^{*} \\
&= \mathbf{D}^{-\mathbf{1}}((\mathbf{L}+\mathbf{U})\mathbf{u}^{k} + \mathbf{f}) - \mathbf{u}^{*} \\
&= \mathbf{D}^{-\mathbf{1}}((\mathbf{L}+\mathbf{U}+\mathbf{D}-\mathbf{D})\mathbf{u}^{k} + \mathbf{f}) - \mathbf{u}^{*} \\
&= \mathbf{D}^{-\mathbf{1}}((\mathbf{D}-\hat{\mathbf{T}})\mathbf{u}^{k} + \mathbf{f}) - \mathbf{u}^{*} \\
&= \mathbf{D}^{-\mathbf{1}}(\mathbf{D}\mathbf{u}^{k} - \mathbf{r}^{k}) - \mathbf{u}^{*} \\
&= \mathbf{u}^{k} - \mathbf{u}^{*} - \mathbf{D}^{-\mathbf{1}}\mathbf{r}^{k} \\
&= \mathbf{e}^{k} - \mathbf{D}^{-\mathbf{1}}\mathbf{r}^{k}
\end{aligned}$$

which is much more efficient, as it is very cheap to invert a diagonal matrix with a uniform diagonal. For Jacobi's Method we get slow convergence for both high and low frequencies with fast convergence in a mid range. This suppression can be moved to higher frequencies by adding a damping parameter γ. The error update then becomes

$$\mathbf{e}^{k+1} = \mathbf{e}^{k} - \gamma\mathbf{D}^{-\mathbf{1}}\mathbf{r}^{k} \tag{19}$$

The rate of convergence in the error recursion depends on the spectral radius of $\mathbf{P}_j$, as shown in (10). In the one dimensional case, the iteration matrix

$$\mathbf{P}_j(\gamma) = (1-\gamma)\mathbf{I} + \gamma\mathbf{P}_j = (1-\gamma)\mathbf{I} + 2\gamma\frac{\beta}{h^2 - 2\beta}\mathbf{S} \tag{20}$$

where $\mathbf{S}$ is a symmetric matrix with ones on the super- and sub-diagonals. Since

$$\lambda_k[\mathbf{S}] = 2\cos(\frac{k\pi}{N+1}), \quad k = 1...N \tag{21}$$

it follows that

$$\lambda_k[\mathbf{P}_j] = (1-\gamma) + 2\gamma\frac{\beta}{h^2 - 2\beta}\cos(\frac{k\pi}{N+1}), \quad k = 1...N. \tag{22}$$

We note that the choice of β and grid resolution N greatly affects the (see Figure 1) spectral radius of the damped iteration matrix in the 1D-case, which naturally extends to the 2D-problem by a variation of the Modulation Theorem. As the solver will be eventually be used in the for the equation where $\beta = \Delta t^2/4 \approx 10^{-4}$ the Jacob iteration was implemented with $\gamma = 2/3$, which is seen to be an apt choice for eliminating the high frequent components of the error.

However, this analysis also indicates that it might be more feasible to chose a γ-sequence, which decreases with increasing N. In fact, this sequence can be computed for specific gamma, so as to put the eigenvalue $\lambda_N = 0$ (see Figure 2) This result only applies to the 1D-case, however, choosing a γ-sequence will be investigated in the 2D-case as well.

Once the high frequency content of the error has been suppressed, the low frequency error will be dealt with in the MG-iteration by solving the linear system in residual form. As mentioned earlier, to be able to solve the linear system coarser grids the grids have to be restricted several times. In order to accomplish this, we define a set of operators to smooth the error and move solutions between the grids.

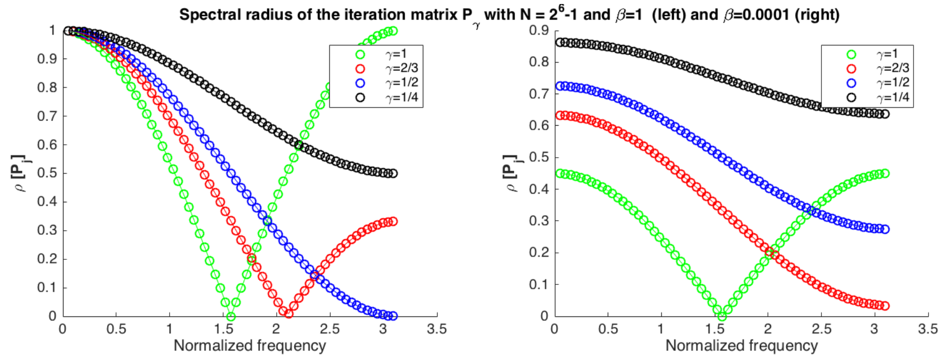

Figure 1: .

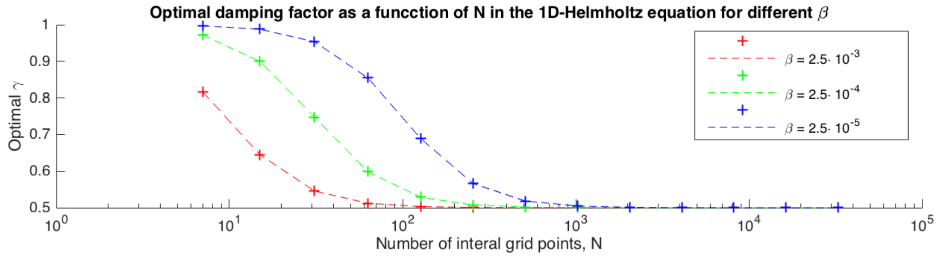

Figure 2: .

We define the operator

$$\mathbf{I}_h^H = \mathbf{I} \otimes \ [0, 1, 0]. \tag{23}$$

as the operator restricting a vector to a coarse grid by eliminating every other internal grid point. For example, the operator applied to a vector, $\mathbf{x}^h$, with $N^h = 7$ internal elements would yield a coarse grid vector, $\mathbf{x}^H$, with $N^H = 3$ internal gridpoints,

$$\mathbf{x}^H = \mathbf{I}_h^H \mathbf{x}^h = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{x}^h \tag{24}$$

When downsampling in this manner, we subject the spectral content to aliasing according to the Nyquist-Shannon sampling theorem. The high frequency content is thereby folded down at the sample frequency, corrupting the low frequency content. To suppress this effect in the error recursion and minimize the influence of aliasing, a two-dimensional digital AA-filter is applied in the multi-grid recursion. The second order lowpass filter kernel

$$F^\pi = (f_{i,j}^\pi) = \frac{1}{16} \left] \begin{matrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{matrix} \right[\tag{25}$$

is applied to the matrix, $\mathbf{A} = (a_{i,j})$, by means of a two dimensional discrete convolution

$$f_{i,j}^\pi * A_{i,j} = \sum_{a=-\infty}^{\infty} \sum_{b=-\infty}^{\infty} f\pi_{a,b} \cdot a_{i-a,j-b} \tag{26}$$

Combined with the operator $\mathbf{I}_h^H$, we have get the restriction operator

$$\mathbf{R}_h^H = \mathbf{I}_h^H F_\pi \tag{27}$$

which effectively restricts the solution and filters out its high frequency components. By defining the prolonging operator as

$$\mathbf{P}_H^h = 2^d F_\pi (\mathbf{I}_h^H)^T = 2^d (\mathbf{R}_h^H)^T \tag{28}$$

where the factor 2^d is a scaling factor depending on the problem dimension d, we see that a matrix defined on the fine grid, $\mathbf{A}_h$, can be written in terms of the same matrix on the coarse grid, $\mathbf{A}_H$, as

$$\mathbf{A}_H = \mathbf{R}_h^H \mathbf{A}_h \mathbf{P}_H^h = 2^d \mathbf{R}_h^H \mathbf{A}_h (\mathbf{R}_h^H)^T. \tag{29}$$

This is a congruence transformation, which has the property of preserving the ellipticity of the operator when moving from one grid to another. Thus, the problem remains elliptical throughout the MG-iteration, provided that $\beta > 0$ (see Figure 3).

In the solver, a `state` variable is used to pass information about the current grid as a `struct` variable. Here, information on the total number of grids used, n_{grid}, the number of pre-/post-smoothing relaxations, η_1/η_2, and the parameter β. In addition, the parameter n_{vw} determines how many times the residual should be compute on each grid, and can be set to 1 for a V-cycle iteration and to 2 for a W-cycle iteration (see Algorithm 1 for a sketch of the MG-solver). Finally, the state variable also keeps track of the grid history, logging every grid the solution has been on in chronological order. Note that the current grid, $n_{current}$, has to be an independent

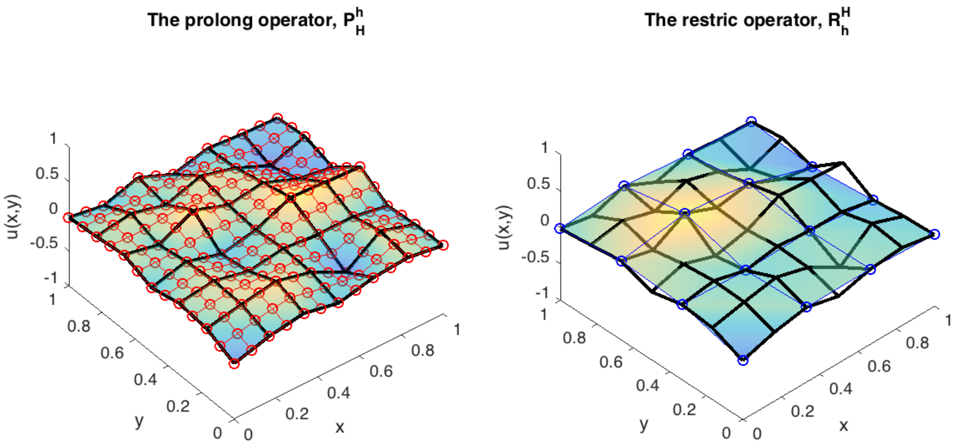

Figure 3: A randomised grid (black) subject to the prolonging operator (left) with the resulting grid (red), and the same original grid (black) subject to the restrict operator (right) and the resulting grid (blue).

variable, local to each recursion in order for the algorithm to return from the coarse grids. As an illustrative example, setting $n_{grid} = 6$ on a very large problem results in the depicted grid history (`state.gridHistory`) on the first ≈ 70 iterations (see Figure 4), which can be used for debugging purposes.

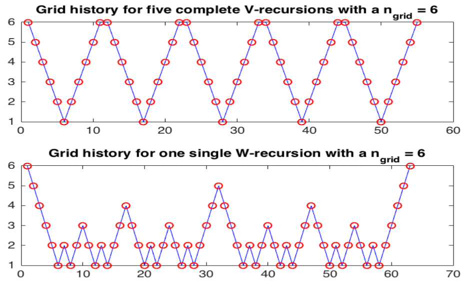

Figure 4: The grid history when applying the MG-solver the test problem with $n_{grid} = 6$, grid 6 being the finest grid and grid 1 being the most coarse grid.

```
Define differential operator kernel: $\hat{\mathbf{T}} = \mathbf{I} - \beta\mathbf{T}_h$;
if $n_{grid} = n_{current}$ then
    Solve linear system for $\mathbf{v}$: $\hat{\mathbf{T}}\mathbf{v} = \mathbf{f}$;
    return $\mathbf{v}$;
end
for $i$ = 1,...,$i_{max}$ do
    for $j$ = 1,...,$\eta_1$ do
        Compute residual: $\mathbf{r} = \hat{\mathbf{T}}\mathbf{v} - \mathbf{f}$;
        Jacobi smoothing: $\mathbf{v} := \mathbf{v} - \gamma\mathbf{D}^{-1}\mathbf{r}$;
    end
    Compute residual: $\mathbf{r} = \hat{\mathbf{T}}\mathbf{v} - \mathbf{f}$;
    Toeplitz AA-filter residual: $\mathbf{r} := F_\pi\mathbf{r}$;
    Restrict to coarser grid: $\mathbf{r}_c = \mathbf{R}_h^H\mathbf{r}$;
    for $j$ = 1,...,$n_{vw}$ do
        Coarse grid error recursion: $\mathbf{e}_c$ = FMGV(0,$\mathbf{r}_c$, state);
    end
    Prolong and remove error from solution: $\mathbf{v} := \mathbf{v} - 2^d(\mathbf{R}_h^H)^T\mathbf{e}_c$;
    for $j$ = 1,...,$\eta_2$ do
        Jacobi post-smoothing: $\mathbf{v} := \mathbf{v} - \gamma\mathbf{D}^{-1}(\hat{\mathbf{T}}\mathbf{v} - \mathbf{f})$;
    end
end
return $\mathbf{v}$, state;
```

Algorithm 1: A sketch of the Helmholtz solver

3 2D - Wave Equation

In this problem we consider the wave equation defined on a unit square

$$\frac{\partial^2 u}{\partial t^2} = c\Delta u, \qquad t \geq 0, (x,y) \in \Omega \tag{30}$$

$$u(x,y) = 0, \qquad t \geq 0, (x,y) \in \partial\Omega \tag{31}$$

where $\Omega = \{(x,y) : 0 \leq (x,y) \leq 1\}$ and $c = 1$ for the sake of simplicity. By denoting $u_t = v$, and using implicit time stepping with the trapezoidal rule to overcome the CFL-condition, we find that

$$\begin{cases} u^{n+1} = u^n + \Delta t(v^{n+1} + v^n)/2 \\ v^{n+1} = v^n + \Delta t(u^{n+1}_{xx} + u^n_{xx} + u^{n+1}_{yy} + u^n_{yy})/2 \end{cases} \tag{32}$$

where Δt is a fixed time step. Just as with the Helmholtz equation, we assume an equidistant grid in the $x-$ and $y-$directions with$\Delta x = \Delta = h$, with N^2 points in the computational domain. Using the two dimensional five-point FDM operator

$$\mathbf{T} = \mathbf{I} \otimes \mathbf{T}_1 + \mathbf{T}_1 \otimes \mathbf{I}, \tag{33}$$

where $\mathbf{T}_1$ is tidiagonal matrix with $-2/h^2$ on the diagonal, and $1/h^2$ on the super- and sub-diagonals, the discretised equation becomes

$$\begin{cases} \mathbf{u}^{n+1} = \mathbf{u}^n + \frac{\Delta t}{2}(\mathbf{v}^{n+1} + \mathbf{v}^n) \\ -\frac{\Delta t}{2}\mathbf{T}\mathbf{u}^{n+1} + \mathbf{v}^{n+1} = \mathbf{v}^n + \frac{\Delta t}{2}\mathbf{T}\mathbf{u}^n \end{cases} \tag{34}$$

$$\Updownarrow \tag{35}$$

$$\begin{cases} \mathbf{u}^{n+1} = \mathbf{u}^n + \frac{\Delta t}{2}(\mathbf{v}^{n+1} + \mathbf{v}^n) \\ \Big(\mathbf{I} - \frac{\Delta t^2}{4}\mathbf{T}\Big)\mathbf{v}^{n+1} = \Delta t\mathbf{T}\mathbf{u}^n + \Big(\mathbf{I} + \frac{\Delta t^2}{4}\mathbf{T}\Big)\mathbf{v}^n \end{cases} \tag{36}$$

The difficulty is then to solve the linear system for $\mathbf{v}^{n+1}$ on each time step, but this problem is equivalent to solving a sequence Helmholtz equaiton (1) with

$$\begin{cases} \beta = \frac{\Delta t^2}{4} \\ \mathbf{f}^n = \Delta t\mathbf{T}\mathbf{u}^n + \Big(\mathbf{I} + \frac{\Delta t^2}{4}\mathbf{T}\Big)\mathbf{v}^n \end{cases} \tag{37}$$

and for this purpose, we use MG solver derived in **Section 2**. A sketch of the algorithm

```
Initialize: $\mathbf{u}^0, \mathbf{u}^0, \Delta t, n_{max}, i_{max}, n_{grid}, n_{current}, \epsilon$;
Compute $\beta$;
for $n$ = $0,\ldots,n_{max}$ do
    Compute the funciton vector $\mathbf{f}^n$ in (37);
    Initialize MG: $\mathbf{v}^0_{MG} = \mathbf{v}^n$, $n_{current} = 1$, state;
    for $i$ = $0,\ldots,i_{max}$ do
        $\mathbf{v}^{i+1}_{MG}$ := FMGV($\mathbf{v}^i_{MG}, \mathbf{f}^n, n_{current}$,state);
    end
    Compute the solution $\mathbf{u}^{n+1}$ from in (36)
    Visualize solution;
end
```

Algorithm 2: A sketch of the 2D-Multigrid solver

4 Results

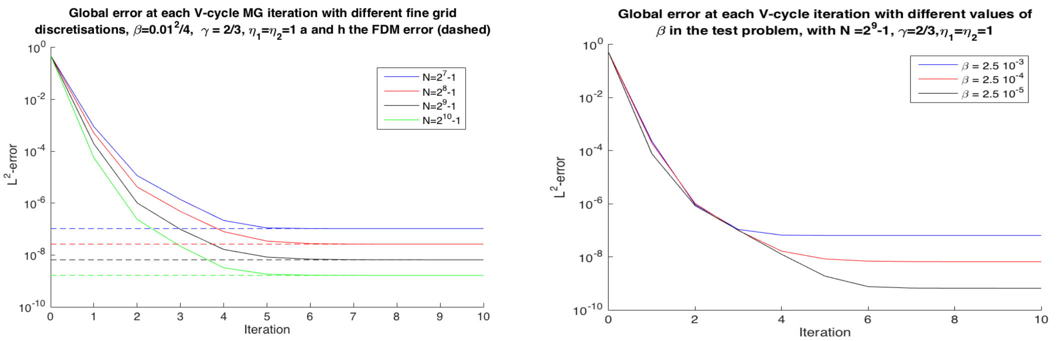

Figure 5: L^2-error of the computed solution relative the analytical solution when varying the size of the computational points $N \in \{2^7-1, 2^8-1, 2^9-1, 2^{10}-1,\}$ (left) and when varying $\beta \in \{2.5 \cdot 10^{-1}, 2.5 \cdot 10^{-3}, 2.5 \cdot 10^{-5}\}$ in the differential oprator (right).

n_{grids}	Type	Computational time [s]	Convergence iteration	η_1	η_2
8	V-cycle	20.747	9	1	1
8	V-cycle	31.7449	5	2	2
8	W-cycle	31.3731	8	1	1
8	W-cycle	44.8414	5	2	2

Table 1: Computational times and iteration at which the V-cycle MG method converges to the FDM error when using one or two η_1 pre-smoothing iterations and and η_2 post-smoothing iterations for a problem with $N = 2^{12}-1$, i.e. $\approx 17 \cdot 10^6$ unknown equations.

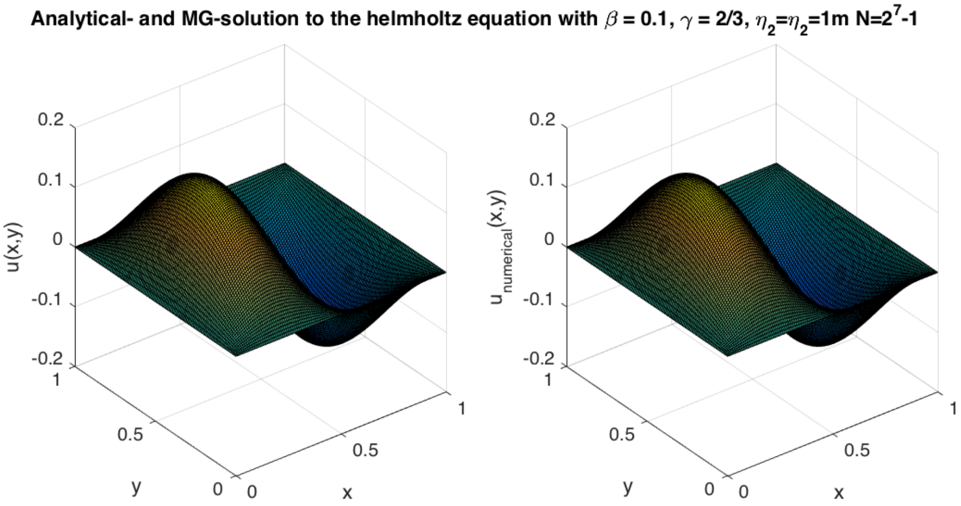
Analytical- and MG-solution to the helmholtz equation with β = 0.1, γ = 2/3, η_2=η_2=1m N=2^7-1
u(x,y)
$u_{numerical}$(x,y)
0.2
0.1
0
-0.1
-0.2
1
0.5
0
y
x

Figure 6: The analytical solution and MG-solution to the testproblem, with $\beta = 0.1, \gamma = 2/3, \eta_1 = \eta_2 = 1$ and $N = 2^7 - 1$.

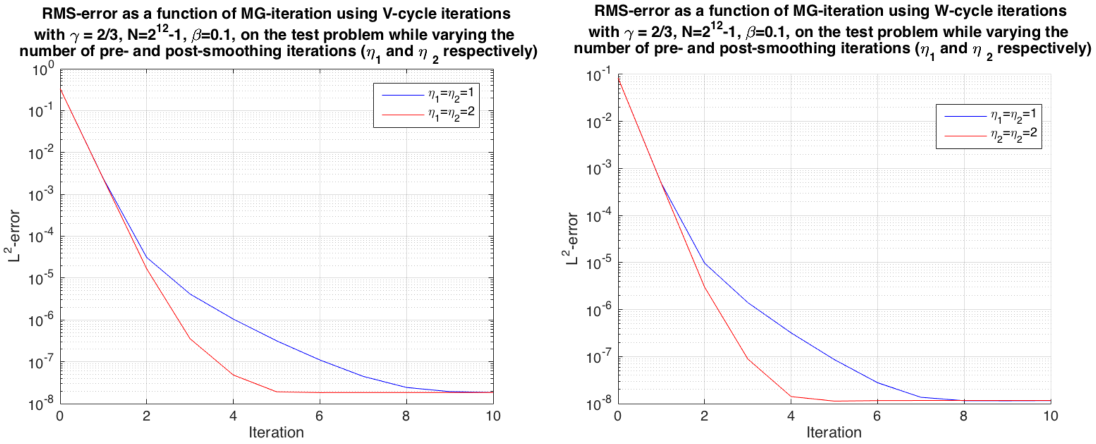

Figure 7: The RMS-error as a fucntion of MG iteration when one or two (η_1) pre-smoothing iterations and and (η_2) post-smoothing iterations with $\eta_1 = \eta_2 = 1$ (blue) and $\eta_1 = \eta_2 = 2$ (red), and a computational domain with $N = 2^{12} - 1$, i.e. $\approx 17 \cdot 10^6$ unknown equations with the V-cycle (left) and the W-cycle (right).

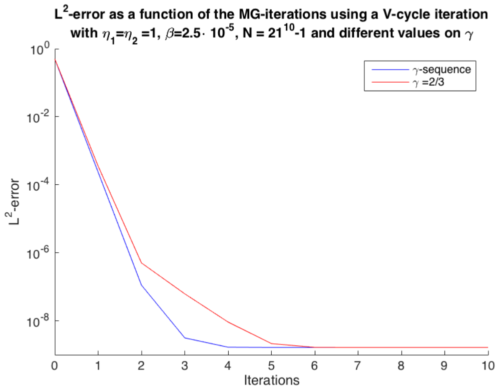

Figure 8: L^2-error as a fucntion of MG-iteration for $\beta = 0.01^2/4$, with $\gamma = 2/3$ (red) and a γ-sequence described in **Section 2**, decreasing with N.

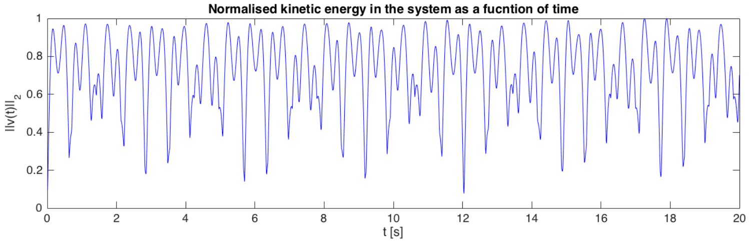

Figure 9: The normalized kinetic energy of the system (if we can assume that each node contains an equal mss.

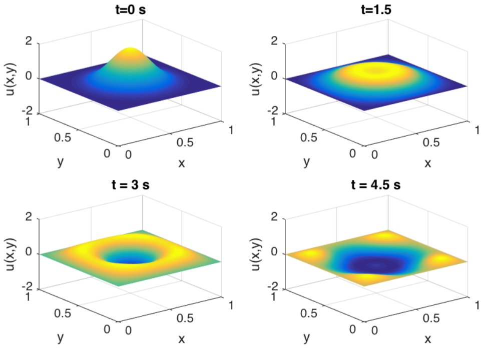

Figure 10: Time evolution of a wave in the membrane, plotted at four points in time.

5 Analysis

In order to test the MG-method, an analytical solution to the Helmholtz equation is needed, as computing the algebraic solution becomes infeasible for large N. Consider the test problem,

$$u - \beta\Delta u = f = \sin(2\pi x)\sin(\pi y), \tag{38}$$

on a unit square $\Omega = \{(x, y) : 0 \leq (x, y) \leq 1\}$, with u=0 for $(x, y) \in \partial\Omega$. Assuming that the differential operator behaves similarily to the laplace operator, the ansatz of $u^{(s)}(x, y) = C\sin(2\pi x)\sin(\pi y)$ is made for some constant C. It then is easily seen that

$$u^{(s)}(x, y) = \frac{1}{1 + 5\beta\pi^2}\sin(2\pi x)\sin(\pi y) \tag{39}$$

satisfies (38) and that $u^{(s)}(x, y) = 0$ on $\partial\Omega$. Graphically, analytical solution closely resembles solution for the MG-method (See Figure 6), and by plotting the L^2-error as a function of the MG iterations, the global error is shown to converge to the error of the FDM discretization given enough iterations (See Figure 5).

As the convergence rate of the MG scheme is to be virtually invariant of the number of points on the computational domain, the L^2-error was plotted at each iteration when solving the test problem (38) for various N with a total number of $n_{grid} = 3$ grids and $\beta = 2.5 * 10^{-5}$ (See Figure 5). This choice of β is of particular interest, as it corresponds to choosing the implemented time step $\Delta t = 0.01$ in the wave equation solver (37).

When using one pre- and post relaxation on each grid, the MG-method converges to the FDM error in ≈ 6 iterations in all cases, virtually independent of the coarseness of the initial grid. It can be seen that the convergence rate is quadratic for the first 2 iterations and on average super-linear until the discretisation error is reached, due to the damping factor of $\gamma = 2/3$ in combination a single pre-/post-relaxation not being enough to smooth out the error when β is small. Note that increasing N by a factor of 2 effectively halves h, thereby decreasing the discretisation error in the second order FDM scheme by a factor of 4. Hence, the stationary error decreases by equidistant steps in the logarithmic plot when increasing N by a factor of 2.

The convergence rate clearly depends on γ, which can in part be explained be the analysis in **Section 2**, where the smoothing properties of the Jacobi iteration was shown to depend on the choice of β. We also note that the stationary error is different for different values of β. When looking at the analytical solution (39), we see that

$$\lim_{\beta\to 0}\sup_{\Omega}|u^{(s)}| = 1 \Rightarrow \lim_{\beta\to 0} u^{(s)} = \sin(2\pi x)\sin(\pi y). \tag{40}$$

Similarily,

$$\lim_{\beta\to\infty}\sup_{\Omega}|u^{(s)}| = 0 \Rightarrow \lim_{\beta\to 0} u^{(s)} = 0 \ \ \forall(x, y) \in \Omega \tag{41}$$

In the MG-solver, the initial guess was set to $\mathbf{u}_0 = \mathbf{f} = \sin(2\pi\mathbf{x})\sin(\pi\mathbf{y})$ and it is therefore natural that a smaller error is observed for smaller β (See Figure 5).

When comparing the W- to the V-cycle iteration, we note that the computational time is much greater for the W-cycle iterations, which is evident from the recursion plot in **Section 2** (see Figure 4). Here it was shown that for $n_{grid} = 6$, one W-cycle when requires ≈ 63 translations between grids, whereas the V-cycle requires ≈ 11 grid switches. Recalling that each translation requires both prolonging, restricting, smoothing and filtering, the difference in computational time is evident. We also see that the Jacobi smoothing greatly contributes to the computational time, where an increase in 2-smoothings per level extends the computational time by ≈ 50% both in the W- and V-cycle. The rate of convergence increases marginally when using the W-cycle iteration (see Figure 7). However, when looking at the precise iteration when the global error is below the FDM error, we see that convergence (by this criteria) occurs at iteration $k = 5$ for both W and V with $\eta_1 = \eta_2 = 2$, although a difference is easily seen in the plot. However, the result is differs more in the case where $\eta_1 = \eta_2 = 1$, where the difference between W and V is a full iteration. In conclusion, the most time efficient method of computing the numerical solution (with $\beta = 0.01$) is by using V-cycle iteration, and the use of two Jacobi-relaxation iterations is the fastest by a very slim margin.

In the spectral analysis of the Jacobi iteration, it was noticed that the damping factor γ could be change dynamically, monotonically decreasing with N, to better suppress error with high frequency content. To test this theory in practice, the previous choice, $\gamma = 2/3$ was tested on the test problem, with $\beta = 0.01^2/4$. Here we again notice convergence in ≈ 6 iterations, certainly feasible. However, when introducing a $\gamma = C_1 2^{-n_{current}} + C_2$, with the constants chosen so that $\gamma \approx 0.9$ at $N = 2^3 - 1$ and $\gamma \approx 0.6$ at $N = 2^{20} - 1$, the rate of convergence is improved dramatically. For the test problem, the global error equivalent to the FDM discretisation is achieved in a mere 4 iterations, compared to the 6 iterations required when having $\gamma = 2/3$ (see Figure 8). This result is very interesting, but we do not know if the theory is accurate, or if this is just a lucky coincidence. Any input would be greatly appreciated!

When solving the wave equation, simulating with $N = 2^7 - 1$ internal points in the x- and y-directions proved more than enough, which meant that the time step, Δt, could be set as low as $0.01s$ (such that $\beta = \Delta t^2/4 = 2.5 \cdot 10^{-5}$) with convergence in ≈ 6 steps, virtually independent of the choice of N as demonstrated above (see Figure 5). With a maximum number $i_{max} = 8$ iterations, convergence is guaranteed (heuristically) if the V-cycle is used with $\eta_1 = \eta_2 = 1$. In addition, this approach was shown to be a computationally efficient choice in the above discussion (see Figure 7).

To test the method, the kinetic energy of the membrane with homogenous mass matrix, $\mathbf{m}$,

$$E_{kin}(t) = \frac{\mathbf{m}\dot{\mathbf{u}}(t)^T \dot{\mathbf{u}(t)}}{2} = \frac{\mathbf{m}\mathbf{v}(t)^T \mathbf{v}(t)}{2} \propto ||\mathbf{v}(t)||_2 \tag{42}$$

and can thus be examine by studying how the two-norm of the $\mathbf{v}$-field varies in time. The Wave equation was simulated with reasonable results, without notable energy dissipation (see Figure 7). This is expected as the spectral radius of the iteration matrix in in the trapezoidal rule can be shown to be 1, independent of the frequency content of the solution. The solution is best visualised with the script `waveMG.m`, but is also illustrated in the time plot (see Figure 10).