Multigrid Method

Fast Iterative Solvers, Project 2

Johannes Leonard Grafen, 380149

July 20, 2023

Contents

List of Figures

1 Remarks on used architecture and compiler options

The code was written in C++, compiled using the Clang compiler and executed on an Apple Silicon M1 Pro Chip featuring the ARM64 architecture. To measure timings, the high resolution clock of the std chrono library was employed. The following flags where passed to the compiler to enhance code performance of the aforementioned architecture: -march=native, -O2. To suppress the output to measure runtime accurately the flag "DISABLEIO" was introduced and passed to the compiler via the "-D" option.

2 Multigrid Method - MG

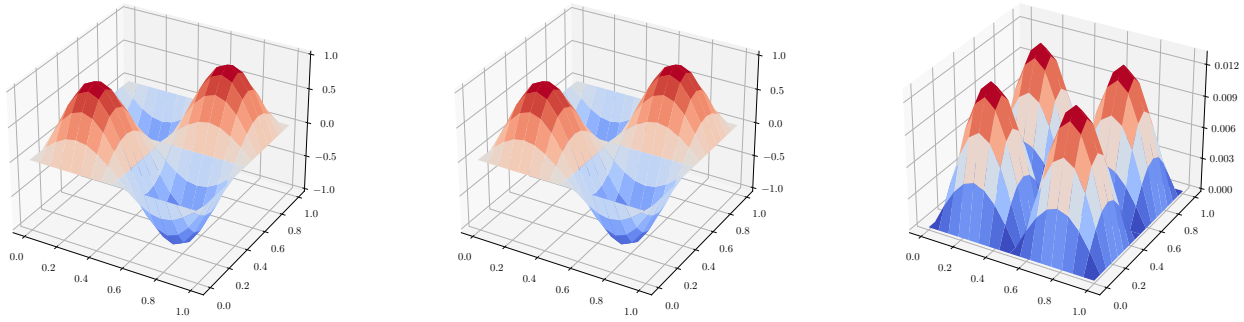
For the Multigrid iterations, W-cycles ($\gamma = 2$) were used. An iteration was considered as converged with the criterion

$$\frac{||r^{(m)}||_\infty}{||r^{(0)}||_\infty} < 10^{-10}, \tag{1}$$

were $r^{(m)}$ denotes the residual of the m-th iteration and $r^{(0)}$ the initial residual using $\mathbf{u} = \mathbf{0}$ as an initial guess for the solution vector.

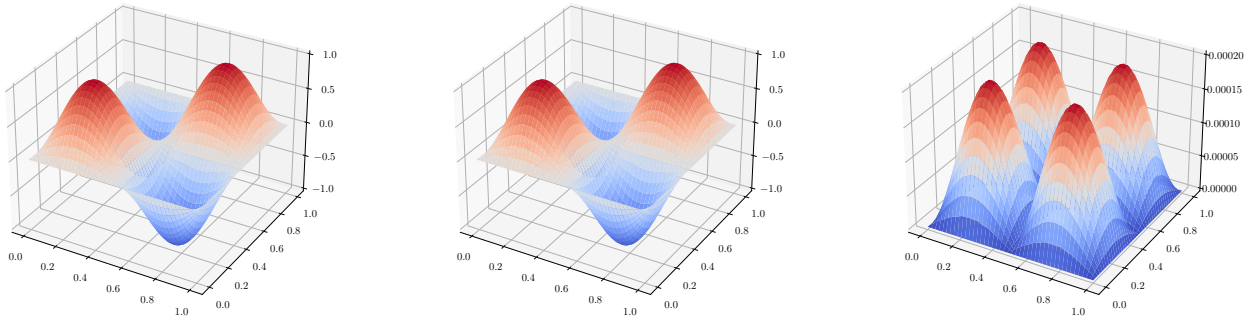
2.1 Validation of MG Method

The solution obtained by the multigrid method is compared to the exact solution $\mathbf{U}_{ex}(x, y) = sin(2\pi x)sin(2\pi y)$ in Fig. 1 for $n = 4$ and in Fig. 2 for $n = 7$. The estimated solution is in good agreement with the exact solution, the max error for $n = 4$ is $\max_{i,j} e_{i,j} = 1.295 \cdot 10^{-2}$ and for $n = 7$ the max error measures $\max_{i,j} e_{i,j} = 2.01 \cdot 10^{-4}$.



(a) $\mathbf{U}_{ex}$ (b) $\mathbf{U}_{est}$ (c) Error $\mathbf{e} = |\mathbf{U}_{ex} - \mathbf{U}_{est}|$

Figure 1: Plots of exact solution, approximated solution and corresponding error for $n = 4$

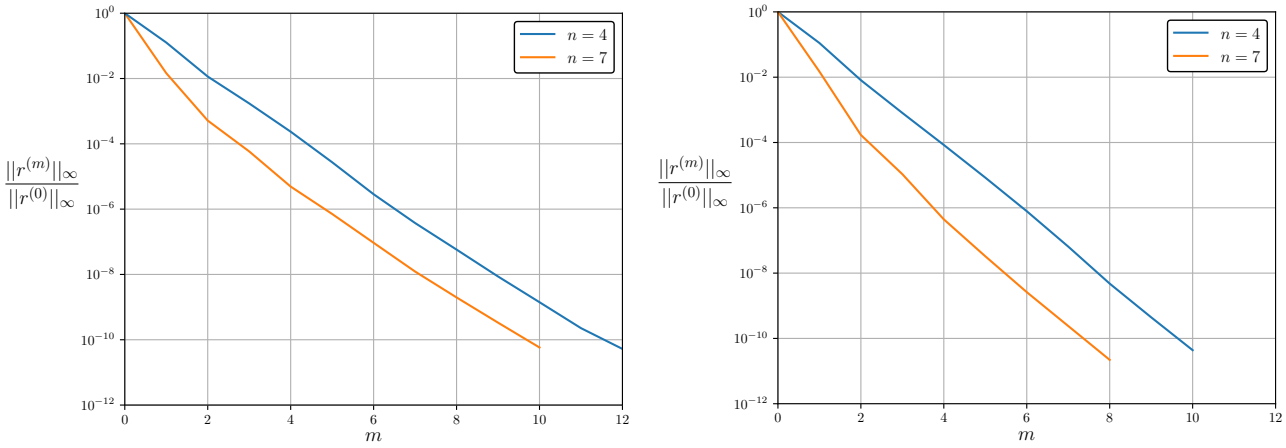


(a) $\mathbf{U}_{ex}$ (b) $\mathbf{U}_{est}$ (c) Error $\mathbf{e} = |\mathbf{U}_{ex} - \mathbf{U}_{est}|$

Figure 2: Plots of exact solution, approximated solution and corresponding error for $n = 7$

2.2 Convergence Plots

In Fig. 3 the convergence of the MG method for $\nu_1 = nu_2 = 1$ and $\nu_1 = 2, \nu_2 = 1$ for two different meshes ($n = 4, n = 7$) is shown. A simulation is considered converged, if the criterion as in Eq. (1) is met. We can observe a faster convergence for the finer mesh compared to the coarser mesh. By increasing the number of the pre smoothing iterations ν_1 a faster convergence can be achieved. The finer mesh with $n = 7$ converges after 8 iterations for $\nu_1 = 2$ opposed to just using a single pre smoothing iteration ($\nu_1 = 1$), which requires 10 iterations to reduce the residual according to Eq. (1). A similar observation can be made for the coarser mesh as well. Further influence of the number of pre smoothing iterations ν_1 is investigated in Sec. 2.3.



(a) $\nu_1 = \nu_2 = 1$ (b) $\nu_1 = 2, \nu_2 = 1$

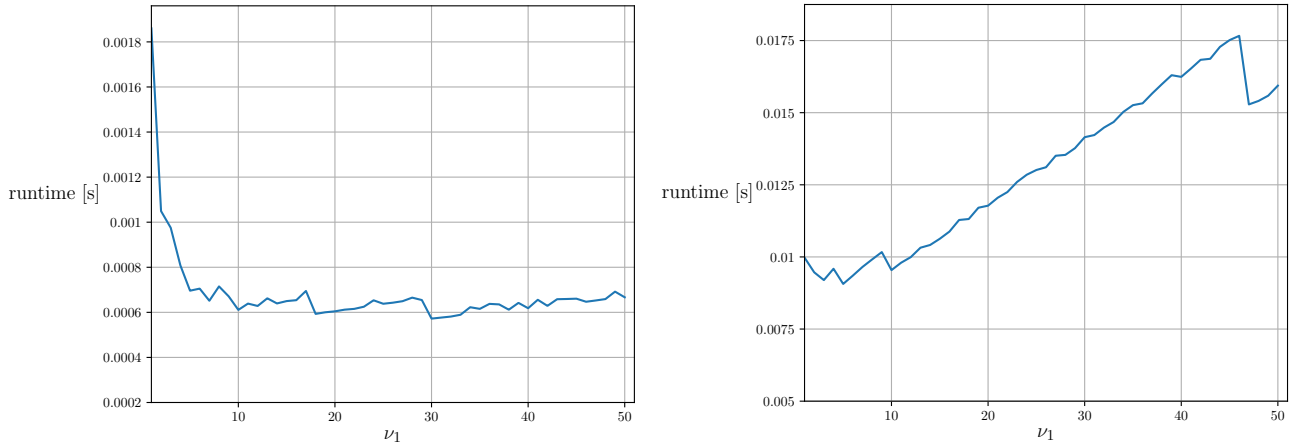
Figure 3: Convergence against multigrid iterations m for meshes $n = 4$ and $n = 7$

3 Further Analysis

This section is dedicated to the analysis of the number of pre smoothing iterations ν_1 that are used in the Gauss-Seidel Smoother and the number of coarse grid cycles γ to obtain a sufficient convergence as in Eq. (1).

3.1 Convergence for different ν_1

In Fig. 4 simulations on two meshes ($n = \{4, 7\}$) were performed for various number of pre smoothing iterations $\nu_1 = 1...50$. The number of post smoothing iterations $\nu_2 = 1$ was set constant throughout all simulations and two coarse grid cycles were used $\gamma = 2$. From Fig. 4 it can be concluded, that there is no further decrease in runtime for a $\nu_1 > 10$ for both meshes. The coarser mesh shown a relatively constant course of the runtime for $\nu_1 > 10$ (Fig. 4a), were as simulations on the finer mesh (Fig. 4b) are subject to a linear increase in runtime for the aforementioned bound.



(a) $n = 4$ (b) $n = 7$

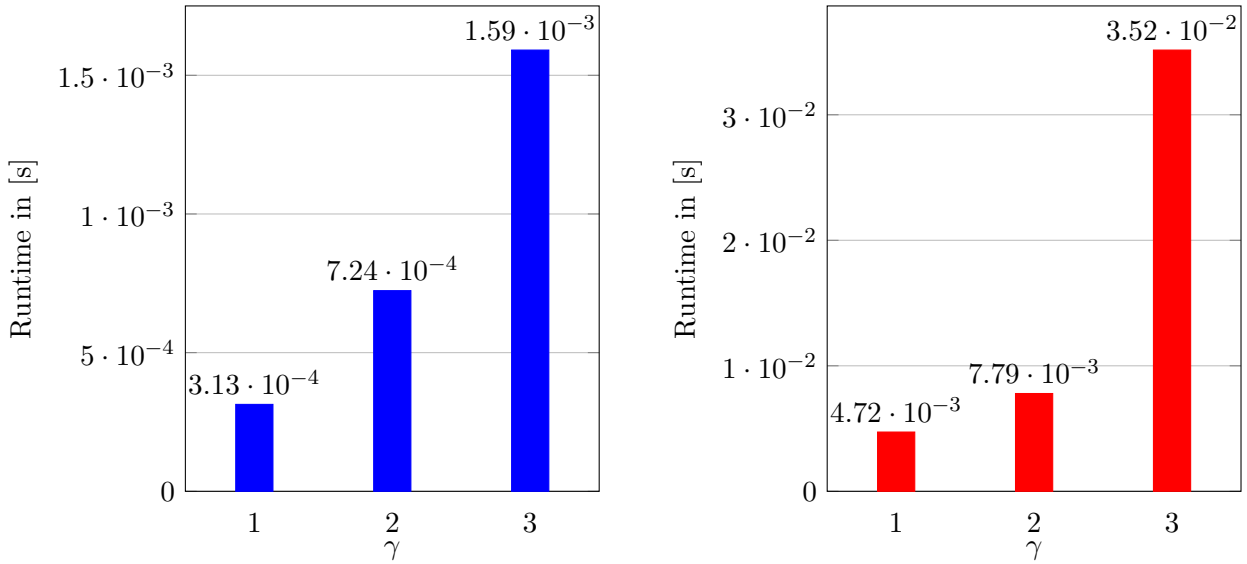
Figure 4: Runtime against number of smoothing iterations $\nu_1 = 1...50$ for meshes $n = 4$ and $n = 7$

3.2 Convergence for different γ

For the investigation of different number of coarse grid cycles $\gamma = 1...10$, the number of pre- and post smoothing iterations was set constant $\nu_1 = \nu_2 = 1$. Different γ were analysed for a coarse and fine mesh, similar to the previous section. Figure 5 displays the Runtime for different γ in the range 1...3, following the theoretical trend for the workload W_k that was discussed in the lecture

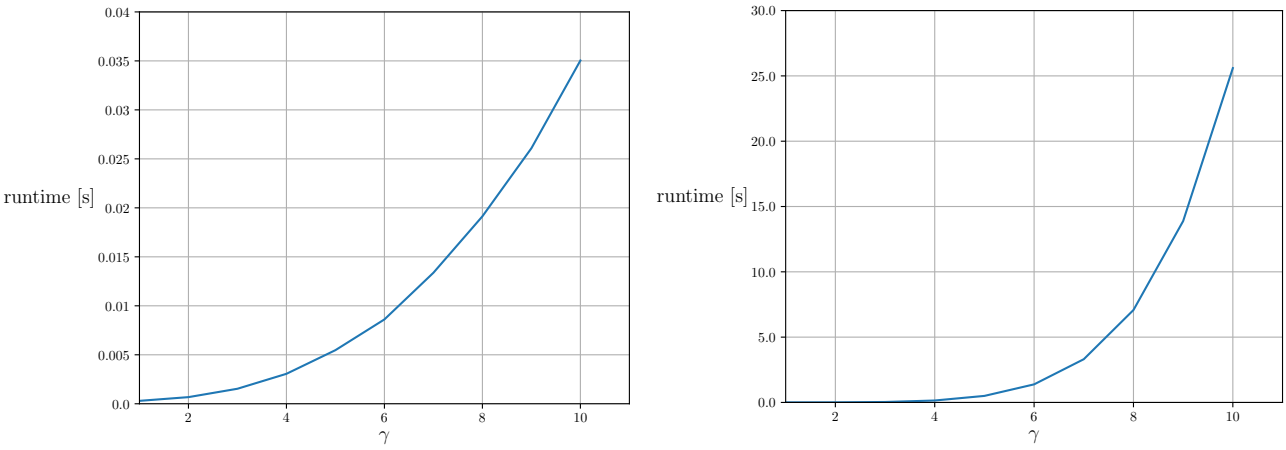
$$W_k \leq C \frac{1}{1 - \frac{\gamma}{4}} N_k. \tag{2}$$

Furthermore, in the lecture was proven that we require $\gamma < 4$, otherwise the workload becomes unbounded. Nevertheless, I conducted further investigations with $\gamma = 1...10$, resulting in an exponential evolution of the runtime as shown in Fig. 6.



(a) coarse mesh $n = 4$ (b) fine mesh $n = 7$

Figure 5: Runtime for $\gamma = \{1, 2, 3\}$ for both meshes



(a) $n = 4$ (b) $n = 7$

Figure 6: Runtime against number of γ-cycles for $\gamma = 1...10$ for meshes $n = 4$ and $n = 7$