Krylov Subspace Methods

Fast Iterative Solvers, Project 1

Johannes Leonard Grafen, 380149

June 6, 2023

Contents

List of Figures

List of Tables

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, a CPU timer object from the Boost timer library was employed. This library was specifically compiled for the aforementioned CPU architecture. Unless specified otherwise, no optimization was performed using the -O0 flag. During timing evaluations, the output was disabled in certain code sections using a preprocessor directive that was introduced as "DISABLEIO," which can be passed to the compiler using the -D option.

2 Generalized Minimal Residual Method - GMRES

2.1 Relative residuals for GMRES with and without preconditioning

In Fig. 1, the relative residuals are plotted against the iteration index of the top loop in the full GMRES algorithm. In this context, the iteration index corresponds to the Krylov vectors. The number of Krylov vectors needed to reduce the 2-norm of the initial residual by eight orders of magnitude ($||\mathbf{r}_k||/||\mathbf{r}_0|| = 10^{-8}$) is denoted as $\tilde{m}$ in the legend of the plot. All iterations were initiated with a parameter $m = 600$. If the convergence criterion was met before reaching the maximum prescribed number of Krylov vectors, the loop was prematurely terminated with $\tilde{m}$.
Comparing it to the full GMRES method without preconditioning ($\tilde{m} = 479$), the Gauss-Seidel preconditioning exhibited the best performance, requiring only $\tilde{m} = 143$ Krylov vectors to achieve convergence based on the aforementioned criterion. For Jacobi preconditioning, $\tilde{m} = 256$ Krylov vectors were necessary, while Incomplete LU Factorization (ILU(0)) required $\tilde{m} = 465$ Krylov vectors to attain convergence.

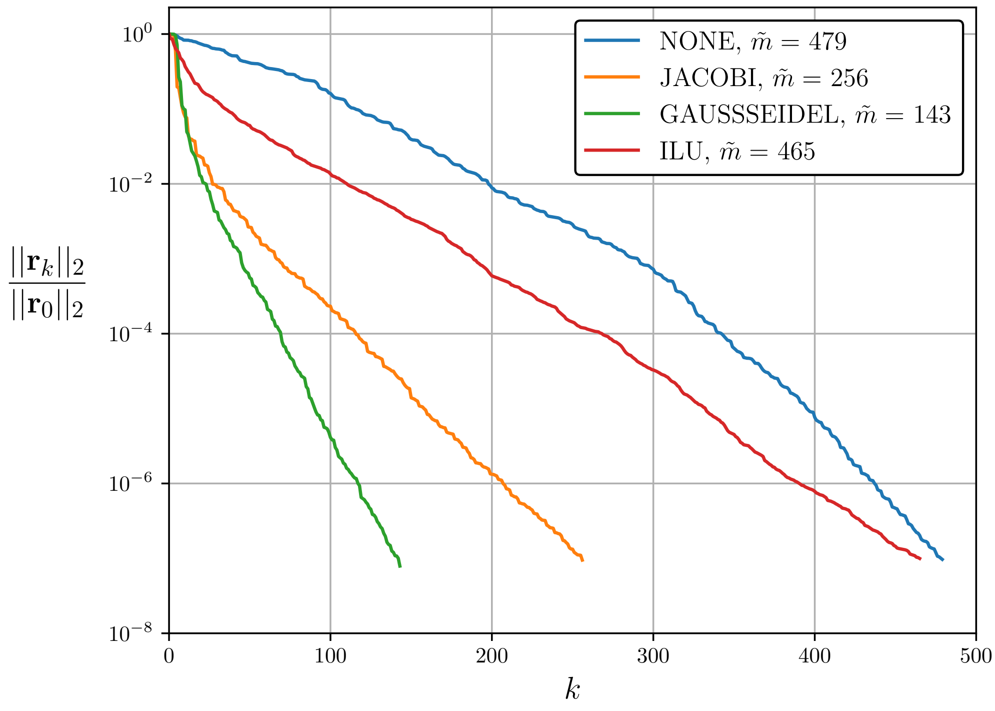

Figure 1: Comparison of the different preconditioning options for the GMRES procedure to the full GMRES Method without preconditioning using m = 600 Krylov vectors as an input.

2.2 Optimization of restart parameter for restarted GMRES

To identify an optimal restart parameter, I compared the values of $m = 30$, $m = 50$, and $m = 100$ in terms of runtime. Tab. 1 displays the results for no optimization, while Tab. 2 presents the findings for a moderate optimization level.

Upon analyzing the restarted GMRES method, I observed that restart parameters of $m = 50$ and $m = 100$ exhibited a decrease in CPU time by 12.8% and 16.1%, respectively, when compared to the full GMRES [1] approach. In contrast, a restart parameter of $m = 30$ resulted in a 43% increase in CPU time when compiler optimization was disabled.

When a moderate optimization level (-O2) was activated, the restart parameters of $m = 50$ and $m = 100$ exhibited a reduced CPU time as well, albeit to a lesser extent. The decrease was 7.7% and 12.82%, respectively, when compared to the full GMRES method.

m	Iterations	User [s]	System [s]	CPU-Time [s]
479 (full)	1	5.04	0.11	5.15
30	144	7.25	0.08	7.33
50	43	4.43	0.06	4.49
100	14	4.23	0.09	4.32

Table 1: Comparison of timings for different restart parameters using clang compiler without optimization (-O0 flag) + disabled Output (-DDISABLEIO) and without preconditioning

m	Iterations	User [s]	System [s]	CPU-Time [s]
479 (full)	1	0.29	0.1	0.39
30	143	0.46	0.06	0.52
50	43	0.29	0.07	0.36
100	14	0.27	0.07	0.34

Table 2: Comparison of timings for different restart parameters using clang compiler with optimization (-O2 flag) + disabled Output (-DDISABLEIO) and without preconditioning

Furthermore, I conducted an investigation to find the "best" restart parameter, which is presented in Fig. 2. The restarted GMRES method was executed for restart parameters $m = 20...478$, where

[1] in the following context "full" GMRES refers to a restarted GMRES method where only one loop iteration of the restarted GMRES method is necessary to establish convergence according to the given convergence criterion.

at least two iterations in the restarted GMRES method were required to reach convergence. The CPU time of the loop inside the restarted GMRES method was measured. The globally best restart parameter was identified to be $m = 63$. With this restart parameter, convergence was achieved in 25 iterations within a runtime of 3.73 seconds, marking a significant decrease of 27.6% compared to the full GMRES method. The red line in Fig. 2 denotes the CPU time of the full GMRES method ($m = 479$).

For all restart parameters m where the blue line falls below the CPU time of the full GMRES method (indicated by the red line), the restarted GMRES formulation proves to be faster than the full GMRES method. The steps in the course of CPU time can be attributed to the reduction of the required loop iterations. For example $m = 325$ necessitates 3 iterations to converge opposed to $m = 326$, which only needs 2 iterations to converge. Due to an increased number of Krylov vectors m, the runtime of the top loop in the GMRES method increases as well. Besides a decreased runtime, the restarted formulation also allows for a decrease in required memory, as the stored Hessenberg matrix $\mathbf{H}$ and also the matrix $\mathbf{V}$ that contains the Krylov vectors are smaller in the restarted formulation compared to the full GMRES method. This becomes especially beneficial when dealing with large input matrices $\mathbf{A}$, as it prevents the need for an unfeasible amount of memory.

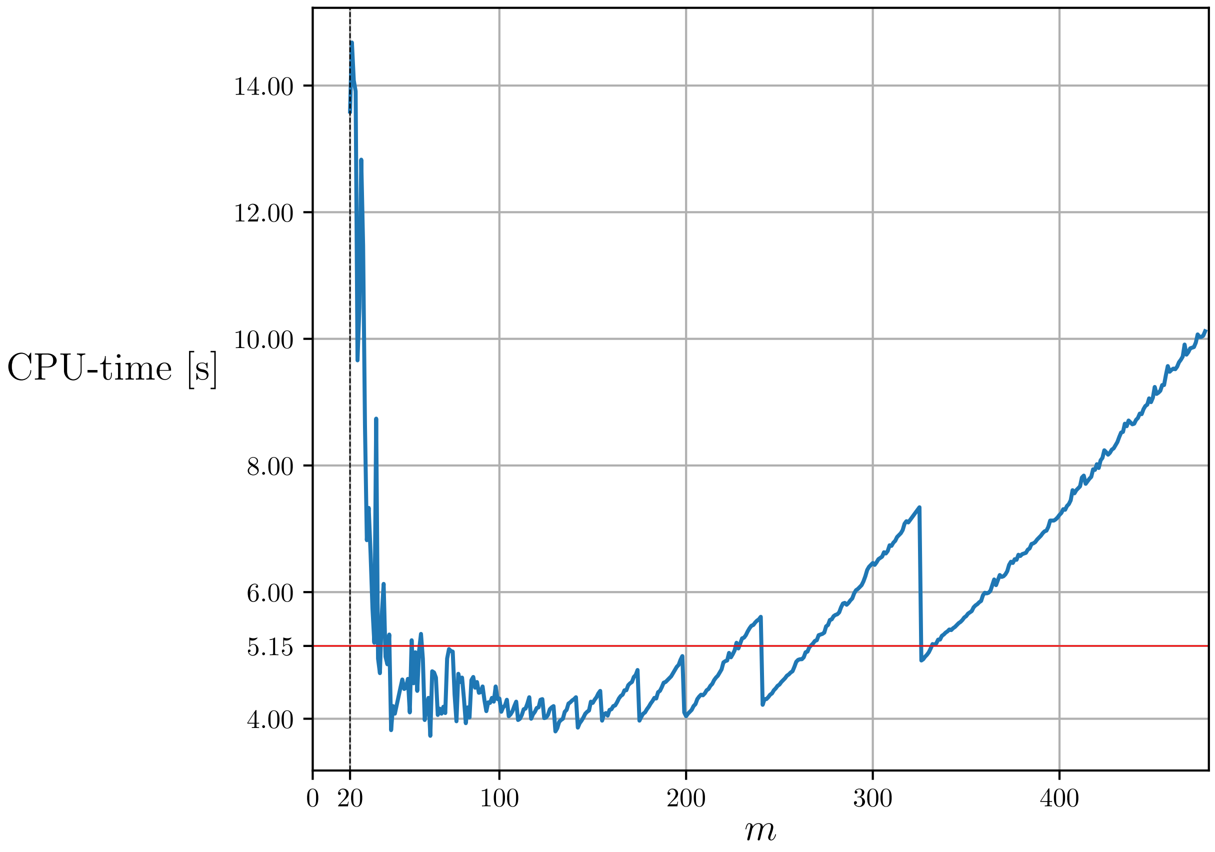

Figure 2: CPU-timings of restarted GMRES for different restart parameters $m = 20...478$ with global minimum at $m = 63$ with $3.73s$, using Clang compiler without optimization (-O0 flag) + disabled Output (-DDISABLEIO) and without preconditioning

2.3 Orthogonality of Krylov vectors

The orthogonality property of the Krylov vectors is represented by Fig. 3 for the full GMRES method. As the number of Krylov vectors increases, the dot product values deviate further from zero. This implies that the angle between the k-th Krylov vector $\mathbf{v}_k$ and the first Krylov vector $\mathbf{v}_1 = \mathbf{r}_0/||\mathbf{r}_0||_2$ deviates more from orthogonality. This deviation is primarily caused by numerical errors that accumulate with each iteration (Krylov vector) in the *getKrylov* function, responsible for determining a new Krylov vector at each step of the top loop in the full GMRES method.

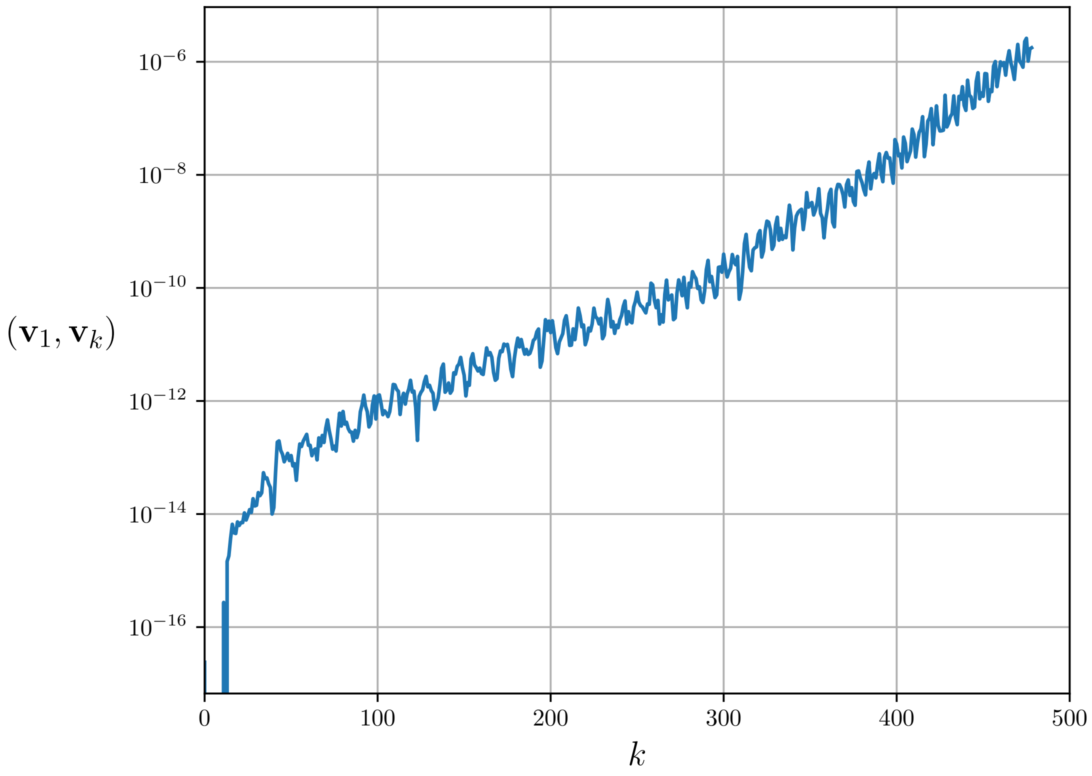

Figure 3: Orthogonality of the Krylov vectors. Plot of the dot product $(\mathbf{v}_1, \mathbf{v}_k)$ against the iteration index k of the top loop in the GMRES method

3 Conjugate Gradient Method - CG

For the given matrix the implemented CG method converges after $k = 21580$ iterations, according to the same convergence criterion that was used for the GMRES method.

The error $\mathbf{e}_k = \mathbf{x}_k - \mathbf{x}$ in A-norm $||\mathbf{e}_A|| = \sqrt{(\mathbf{Ae}, \mathbf{e})}$ as well as the residual $\mathbf{r} = \mathbf{Ax} - \mathbf{b}$ in 2-norm $||\mathbf{r}||_2 = \sqrt{(\mathbf{r}, \mathbf{r})}$ are plotted against the iteration index on a semi-log scale in Fig. 4. The known exact solution is denoted by $\mathbf{x}$.

In the graph, the error in A-norm of the conjugate gradient method exhibits a smooth progression, while the residual experiences significant fluctuations. This behavior can be explained by considering that the CG method minimizes the error $\mathbf{e}$ along all search directions $\mathbf{p}_k$. In the lecture the "Optimality" was characterized as $(\mathbf{Ae}_k, \mathbf{p}_k)_A = 0, j = 0, ..., m-1$. The conjugate gradient method is formulated such that the error is always A-orthogonal to all previous search directions $\mathcal{P}_{m+1}$. Consequently, the error in A-norm is minimized in $\mathcal{P}_{m+1}$ according to the best approximation theorem. For continuous iterations the error should thus be constantly minimized, approaching a global minimum. Where as the 2-Norm of the residual is not taking the search directions into consideration, which results in fluctuations. Nonetheless, the overall trend of the residual aims to follow that of the error in A-norm.

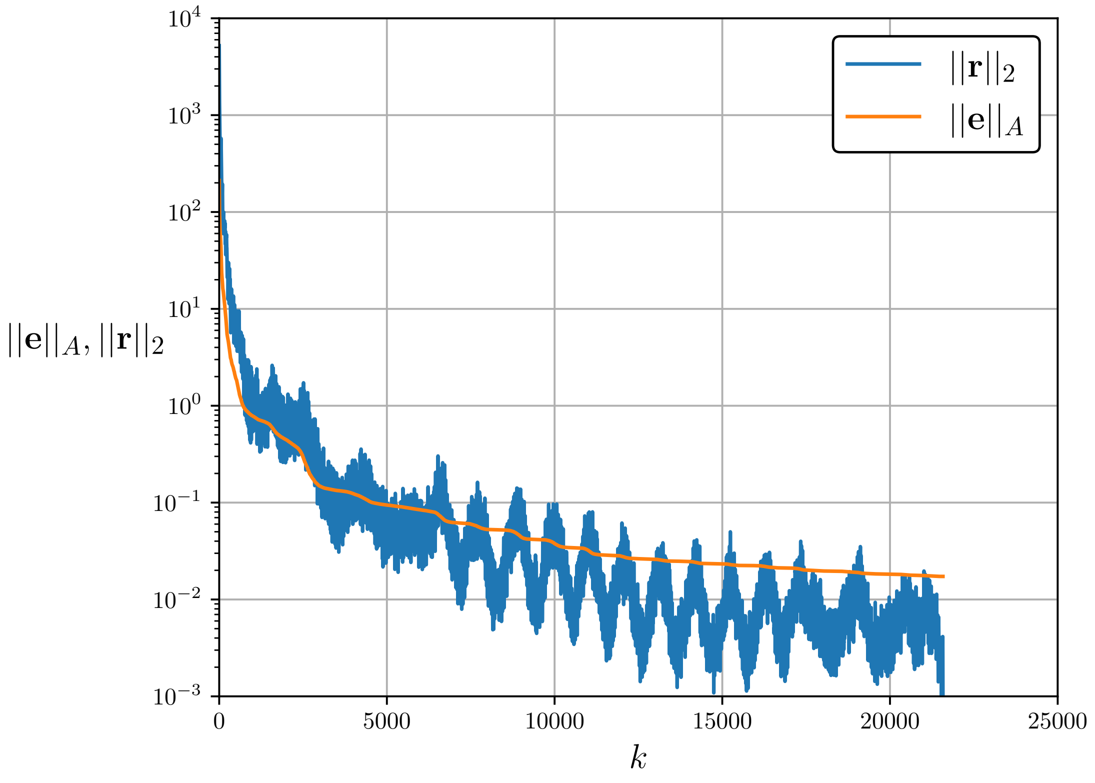

Figure 4: Error $\mathbf{e}_k = \mathbf{x}_k - \mathbf{x}$ in A-norm $||\mathbf{e}||_A = \sqrt{(\mathbf{Ae}, \mathbf{e})}$ and the residual in 2-norm $||\mathbf{r}||_2 = \sqrt{(\mathbf{r}, \mathbf{r})}$ against the iteration index k for CG-method