Fast Iterative Solvers

Marcelo Cabral Filho
Mat. 404864
marcelo.cabral.filho@rwth-aachen.de

Krylov Subspace Methods
Project 1
SS 2023

Abstract

Project in order to implement two different Krylov subspace methods which are the Generalized Minimal Residual method preconditioned (GMRES) and Conjugate Gradients (CG) to solve a linear system $Ax = b$. Both algorithms were implemented in C++ language and some conclusions will be made comparing both models as convergence rate, size of the Krylov space in order to solve the problem, and amount of restart parameters.

1 Conjugate Gradient Method

The Conjugate Gradient method was applied with convergence criteria set as 10^{-8}. The definition for the error of A-norm is: $e_k = x_k - x : ||e_k||_A$ and standard 2-norm of the residual defined as $||r||_2$ are plotted on the same diagram in Figure 1 down under. The first sight of this plot it is noticeable that the Error A-norm has a monotonically decreasing behaviour whereas the Residual 2-norm behaves in an oscillatory manner nevertheless also decreasing. The explanation for this can be introduced by the following Equation (1) where the $||x_m - x||$ is the rate of convergence in the A-norm of the error and it bounds the fluctuations.

$$||x_m - x|| \leq 2\left(\frac{\sqrt{k}-1}{\sqrt{k}+1}\right)^m ||x_0 - x||_A \quad \text{,where} \quad k_2 = \frac{\lambda_m ax(A)}{\lambda_m in(A)} \tag{1}$$

In theory, both $||e||_A$ and $||r||_2$ should converge to zero as the iteration progresses with a finite number of iterations not larger than the size of the matrix. In practice, the exact solution is never obtained since the conjugate gradient method is unstable with respect to even small perturbations, e.g., most directions are not in practice conjugate, due to the degenerative nature of generating the Krylov subspaces. This happens due to several reasons as the conditioning of the matrix A (i.e. $k_2(A)$) which directly impacts the convergence behaviour. For example, if the matrix A is ill-conditioned it can lead to slow convergence and oscillatory behaviour of the residual of 2-norm. Another point is the orthogonality of the iterates. Since the CG method relies on generating orthogonal search directions if the iterations are not perfectly orthogonal due to rounding errors or other factors, it can lead to residual oscillations. However, this does not necessarily indicate a lack of convergence as long as the overall trend of the residual norm is decreasing. It's important to note that the ultimate convergence criterion for the CG method is typically based on the residual norm $||r||_2$ reaching a predefined tolerance level rather than the error norm $||e||_A$. Therefore, monitoring the residual norm $||r||_2$ is a more reliable indicator of convergence in the CG method.

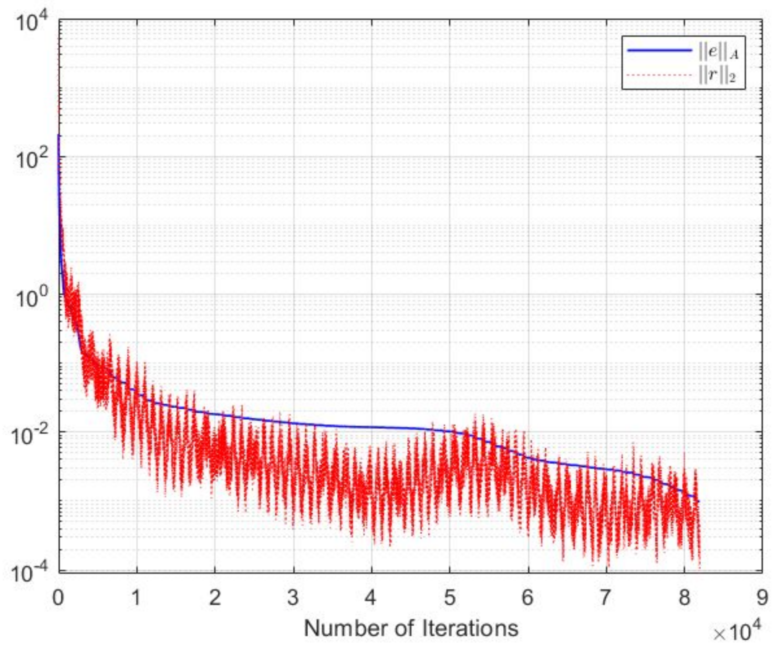

Figure 1: A-norm of Error e_k in blue and 2-norm of Residual r_k in orange at each iteration

2 GMRES Method

2.1 Full GMRES Method

Figure 2 below represents the orthogonality of the vectors V generated in the Krylov subspace. In this case, the set-up used to generate this diagram was m = 512, without preconditioners in a full GMRES space.

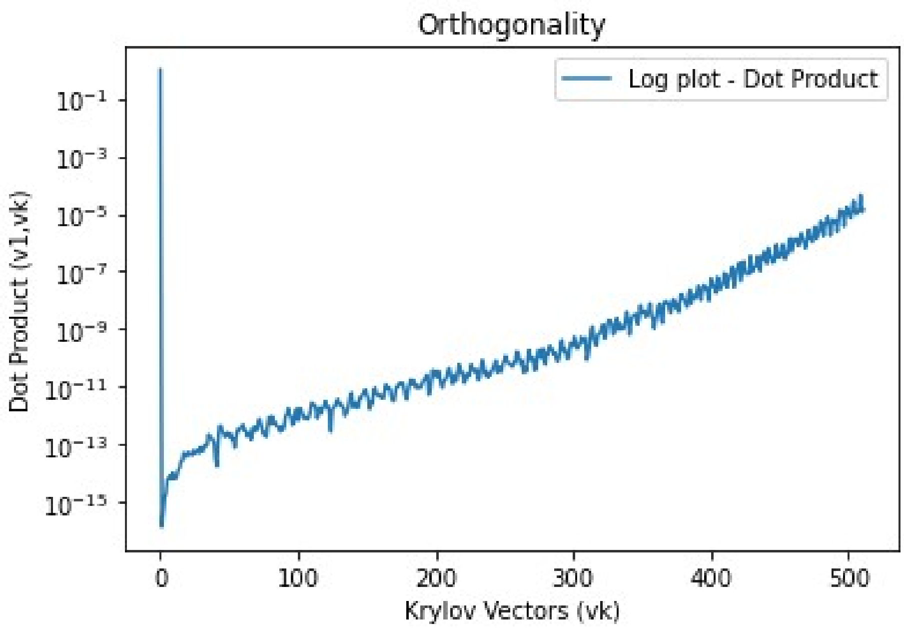

Figure 2: Orthogonality plot between vector V1 and Vk vectors.

For the Full GMRES method, m was selected as 600. The result of relative residual against iteration index on a semi-log scale for both with and without preconditioning is presented below in Figure 3. As can be seen, with the Gauss-Seidel preconditioner, the number of Krylov vectors necessary to solve the same problem with the specified tolerance (i.e. $1e^{-8}$) is smaller. For this project was also requested implement the ILU(0) precondition, however, it was not successfully implemented and thus was not considered in the diagram.

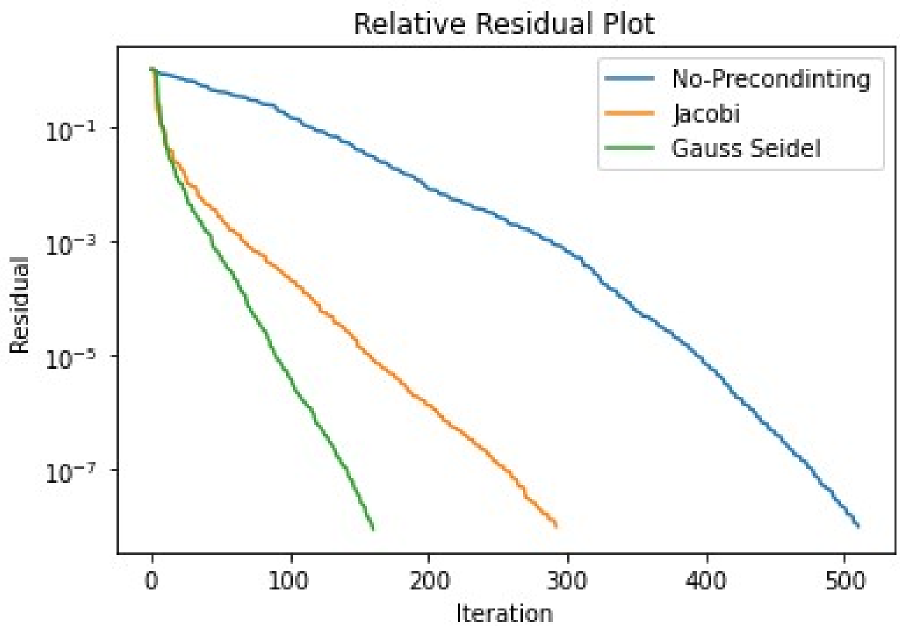

Figure 3: Relative Residual plotted against the iteration index

Table 1 summarizes the Krylov Vectors required to achieve the results within the tolerance limit.

Full Space			
Method	**Krylov Vectors need to Solve [#]**	**Time [s]**	**Restarts [#]**
No-preconditioner	512	2,566	0
Jacobi	293	0,928	0
Gauss-Seidel	161	0,288	0
ILU(0)	-	-	-

Table 1: Result overview of residual obtained depending on the iterations 'm' used as an input parameter

Since the code implementation for ILU(0) didn't work, it is here further discussed its characteristics and some results expectations. In case ILU(0) is implemented properly in the above-given conditions of Table 1, the amount of Krylov Vectors needed to solve the same problem is fewer compared to all the other methods respectively same for the time, it shrinks. The reasoning for such behaviour is, the Jacobi preconditioner approximates the original matrix by using its diagonal elements. The Gauss-Seidel preconditioner considers both the diagonal and the lower-triangular elements of the matrix. It captures some of the off-diagonal interactions and can provide a better approximation compared to the Jacobi preconditioner. However, it still neglects the upper-triangular part of the matrix, which can limit its effectiveness. ILU(0) aims to capture the main features of the original matrix by approximating its LU factorization. It factors the original matrix A into a lower-triangular matrix L and an upper-triangular matrix U, such that $A \cong L \cdot U$. However, unlike the complete LU factorization, ILU(0) only retains a limited number of non-zero entries in L and U. It considers both the diagonal and off-diagonal elements, providing a more accurate approximation compared to Jacobi and Gauss-Seidel preconditioners. ILU(0) takes into account the interactions between elements across the entire matrix and can effectively reduce the condition number and spectral radius of the preconditioned system. This improved approximation leads to faster convergence and the need for fewer iterations in the iterative solver.

2.2 Restarted GMRES Method

The performance of the Restarted GMRES Method for different values of 'm' was evaluated by noting the time taken in each case and summarized in Table 2. As it can be seen, by reducing the Krylov Space Size 'm' the restarts start to take place and then the usage of the previous iteration result as the initial solution guess for the next iteration. A reduction in total time for convergence happen. Besides the improvement in runtime, memory management is another advantage of Restarted GMRES. The reduced size of the Hessenberg matrix and reduce the number of Krylov vectors means that less memory would be required for computations and eventually, the final solution vector would be represented in a smaller Krylov subspace. The Time-Performance evaluation was carried out without any file generation, printing operations, orthogonality check or any extra operations which were not necessary for GMRES.
With the no usage of preconditioners, the best performance achievement which refers to the elapsed time of code running is given when the input 'Size of Krylov Space' (m) equals 50 with 1,042 seconds. For the Jacobi precondition, the best performance is achieved for 'm' equals 30 with 0,13 seconds. Gauss-Seidel similarly to Jacobi achieves its best performance for 'm' equals 30 with 0,074 seconds.

Krylov Space Size (m)	**30**	**50**	**100**	**600**
Run Time [s]				
No-Preconditioner	1,598	1,042	1,294	2,283
Jacobi	0,13	0,156	0,262	0,62
Gauss-Seidel	0,074	0,089	0,131	0,206
ILU(0)	-	-	-	-

Table 2: Results overview of the timings for GMRES solution for different Krylov Space Size 'm' which is used as an input parameter

With the no usage of preconditioners, the best performance achievement is given when the input 'Size of Krylov Space' (m) equals 50 with 54,4% of performance improvement on its elapsed time. For the Jacobi precondition, the best performance is achieved for 'm' equals 30 with 79,0% of performance improvement compared to the worst timing which is obtained for 'm' equals 600. Gauss-Seidel similarly to Jacobi achieves its best performance for 'm' equals 30 with 64,4% of performance improvement. The following Table 3 gives an overview of the results.

Performance Improvement				
Krylov Space Size (m)	**30**	**50**	**100**	**600**
No-Preconditioner	30,0%	54,4%	43,3%	-
Jacobi	79,0%	74,8%	57,7%	-
Gauss-Seidel	64,1%	56,8%	36,4%	-
ILU(0)	-	-	-	-

Table 3: Results overview of the performance improvement in percentage for each preconditioner according to its worst elapsed time

Even though the ILU(0) was not successfully implemented further research about it was done and the respective expectations for such a preconditioner would be for this exact case to be overall better than all the other preconditioners independently of the number of Krylov Space Size and performance improvement would happen only for 'm' smaller then 30 since ILU(0) can solve the same problem with fewer vectors and the window where the performance improvement is noticeable is shifted down when compared to other preconditioner options. Therefore between 'm' 600 to 50 no 'noticeable' improvement is seen. However, for 'm' 30 or even smaller ILU(0) keep improving up to a certain limit which could not be tested.