2 Conjugate Gradients in EIGEN

In Task 2, we were quite surprised that the preconditioned methods worked so well since they both don't work with the exact matrix A like we do in our own implementation. The diagonally preconditioned method only takes the diagonal entries into account and the incomplete Cholesky method is "only" an approximation to the Cholesky factorization itself. Nevertheless, both methods gave similar or better results compared to our own implementation. What is quite remarkable as well is the difference in execution time. Our implementation took 4375 seconds to execute 10000 iterations. The diagonally preconditioned Eigen method delivered a similar (slightly better) result in 13 seconds which makes it faster by a factor of 430.

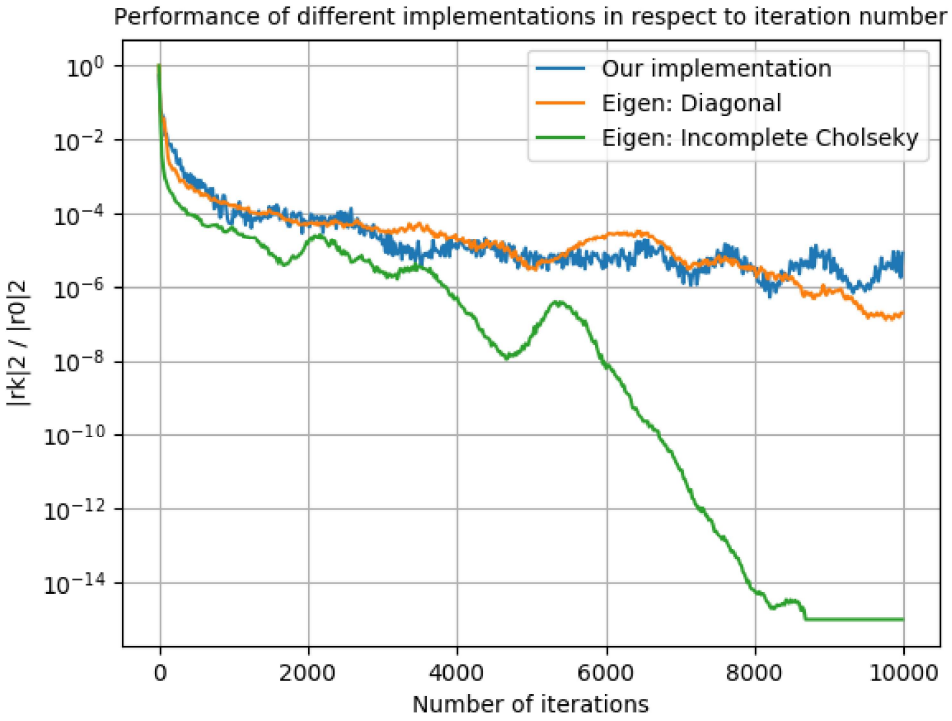

Figure 3: Comparison of three different versions of the Conjugate Gradients method

Note that for readability reasons, we only plotted every 20^{th} value. The program itself is also designed to only give back the value of every 20^{th} iteration. It can be called similar to our program in Task 1. Please consider, that the existing data files will be overwritten once the program is executed and will therefore not deliver the same plots as below when called by the python script. However, backups containing the data files for runs with 10000 iterations is provided in the subfolder *./Abgabe_2/backup_10000it/*.