

NA HW3

Christian Lascsak 01363742

December 2020

1 Conjugate Gradient Method

1.1 Methodology

1.1.1 Remarks

I used Octave version 6.1.0. For calculating the cg, I took the following approach for creating the data:

- $x_0 = rand(n, 1)$
- $b = rand(n, 1)$
- $tol = 1e - 6$

What I did:

- cg without preconditioning
- Jacobi preconditioning
- Incomplete Cholesky preconditioning (except for s3)
- Incomplete Cholesky preconditioning with threshold dropping

What I did not do:

- Block diagonal preconditioner
- Incomplete Cholesky preconditioning for s3 Matrix

I was not able to factorize the s3rmt3m3 Matrix, since I always got the error "negative pivot encountered".

1.2 Implementation

Instead of inverting the Matrix M for calculating $s_k = M^{-1}r_k$, we solve the linear system $Mr_k = s_k$. For the incomplete cholesky with threshold, I choose the value 0.5, because this was the lowest value, where the factorization of the s3rmt3m3 matrix could be done. I was not able to generate

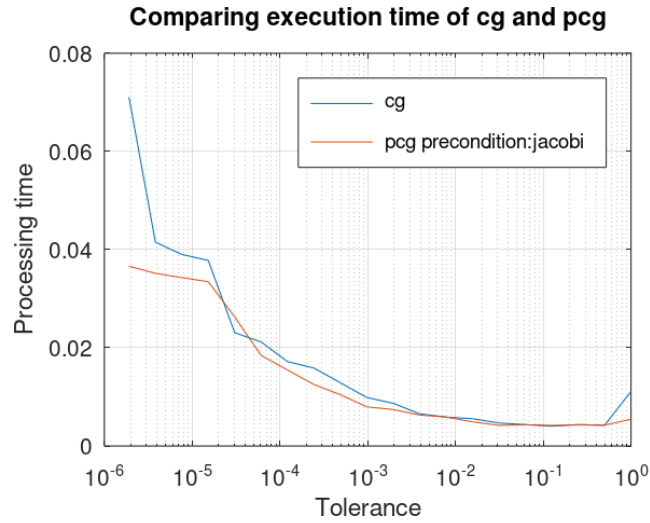


Figure 1: The execution time of the cg method with diagonal jacobi preconditioning

1.3 Results

1.3.1 Performance Metrics

I decided to do my performance measurements on the fastest of the three matrices to calculate – which is the "nos6" matrix. So all the plots regarding the performance are from this matrix. For measuring the performance, I started with a tolerance of 1 and halved this value every iteration. I iterated 20 times over a loop and calculated the pcg with the current tolerance.

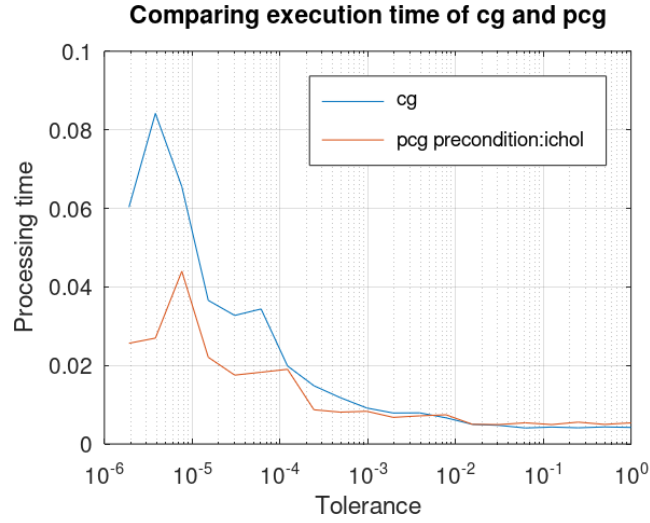


Figure 2: The execution time of the cg method with incomplete cholesky preconditioning

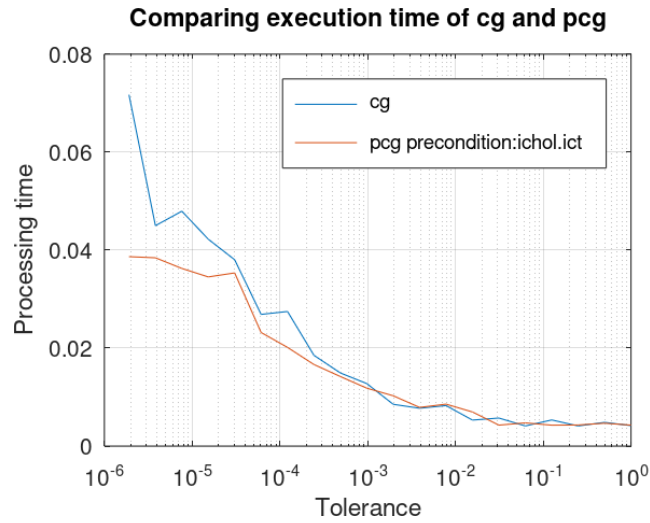


Figure 3: The execution time of the cg method with diagonal incomplete cholesky preconditioning and a threshold of 0.5

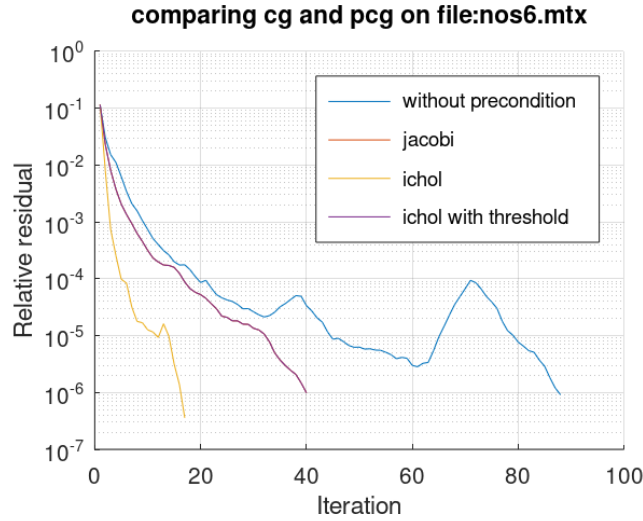


Figure 4: The residuals compared by the nos6 Matrix

As we can see in figure 1, 2 and 3 – all preconditioned cg methods run (expectedly) faster, than without preconditioning. We can also see, that the incomplete cholesky performs the best of them all. The cholesky preconditioning with a threshold performs worse, because I chose a relatively high threshold of 0.5. With a lower threshold, the cholesky can perform better than the cholesky without a threshold.

1.3.2 Comparing Residuals

For Comparing the residuals, I calculated the values for all three matrices and plotted their residual history.

We can see here, that the residual diverge much faster with preconditioning. Incomplete cholesky seems to be the fastest, because it needs less iterations than the other methods. Looking at figure 5, we can see that the jacobi preconditioner is very close to the cholesky one. However compared to cg without preconditioning, it takes more than 500 iterations less to calculate. Also the curve is logarithmic which means that we are converging much faster to the residual.

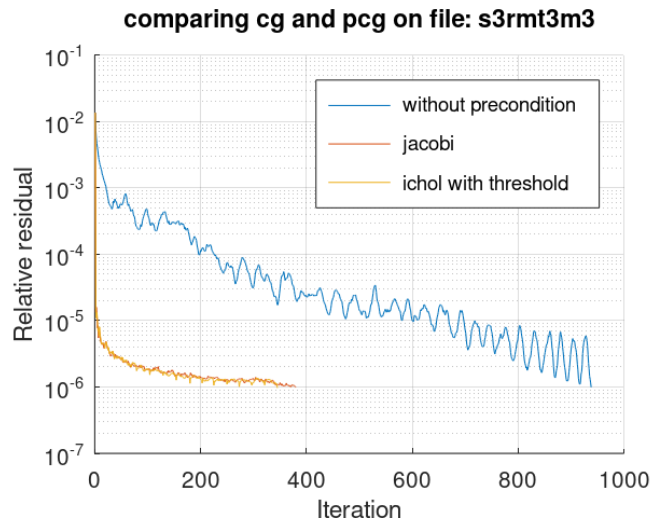


Figure 5: The residuals compared by the s3rmt3m3 Matrix

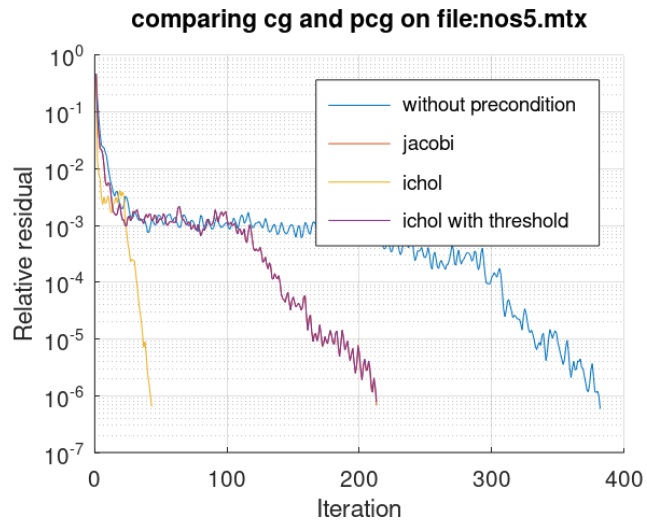


Figure 6: The residuals compared by the nos5 Matrix