Jake Betenson
u0624782 | CS 3200

Assignment 4 – Iterative Methods

This assignment explores the impact of various iterative methods on solving linear systems. All experiments were performed on my personal machine using a 2nd generation i5 processor using the MATLAB software package. Some code blocks were provided by the instructor.

I. Jacobi, Gauss-Seidel and Gradient Descent

For the first part of the experiment a linear system was solved using various iterative methods to an accuracy in the residual measured in the infinity norm of 1.0×10^{-15}. The system is given by $Ax = b$:

$$A = \begin{pmatrix} 9 & -4 & 1 & 0 & 0 & 0 & 0 & 0 \\ -4 & 12 & -4 & 1 & 0 & 0 & 0 & 0 \\ 1 & -4 & 612 & -4 & 1 & 0 & 0 & 0 \\ 0 & 1 & -4 & 12 & -4 & 1 & 0 & 0 \\ 0 & 0 & 1 & -4 & 12 & -4 & 1 & 0 \\ 0 & 0 & 0 & 1 & -4 & 12 & -4 & 1 \\ 0 & 0 & 0 & 0 & 1 & -4 & 9 & 2 \\ 0 & 0 & 0 & 0 & 0 & 1 & -2 & 9 \end{pmatrix}$$

$$b = \frac{1}{N^4}[1,1,\dots,1]^T, N = 8$$

The 3 methods achieved the same solution but took varying amounts of iterations to get there. The Jacobi method took the most iterations at 84, Gradient Descent took 41 iterations, and the Gauss-Seidel method took the least with 21 iterations.

II. Non-Convergence of Iterative Methods

The second portion of the experiment compares a similar system to part one, and explores the non-convergence of the Jacobi method when applied to the following linear system, $Ax = b$:

$$A = \begin{pmatrix} 9 & -4 & 1 & 0 & 0 & 0 & 0 & 0 \\ -4 & 6 & -4 & 1 & 0 & 0 & 0 & 0 \\ 1 & -4 & 6 & -4 & 1 & 0 & 0 & 0 \\ 0 & 1 & -4 & 6 & -4 & 1 & 0 & 0 \\ 0 & 0 & 1 & -4 & 6 & -4 & 1 & 0 \\ 0 & 0 & 0 & 1 & -4 & 6 & -4 & 1 \\ 0 & 0 & 0 & 0 & 1 & -4 & 5 & 2 \\ 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 \end{pmatrix}$$

$$b = \frac{1}{N^4}[1,1,\dots,1]^T, N = 8$$

The Jacobi method does not converge in this case because A is not diagonally dominant. The Gauss-Seidel method converged in 957 iterations.

Jake Betenson
u0624782 | CS 3200

III. Applying Iterative Methods to Ill-Conditioned Matrices

Part 3 of the experiment applies Gradient Descent and Conjugate Gradient Codes to the same system in Part 2. Gradient Descent succeeded in 2164 iterations while the Conjugate Gradient Codes fails to converge within 10000 iterations. Further inspection of A shows a wide spread of Eigenvalues, leading to a poor condition number, κ. Inspecting A from Part 1 shows a closer spread of Eigenvalues, meaning there will be a better condition number. The better condition number indicates that Conjugate Gradient Codes should converge in a fewer number of iterations. Gradient Codes should converge in a fewer number of iterations.

IV. Hilbert Matrices, Gradient Descent and Conjugate Gradient Codes

For the final part of the experiment, Hilbert Matrices of dimension varying with $n = 4,8,12,32,40$ were used to solve a system of $Ax = b$, where b consists of all ones and A is a Hilbert matrix of size n. Conjugate Gradient Codes was applied with an early out of residuals measured to the infinity norm of 1×10^{-15} or 10,000 total iterations.

This same scenario was also solved using Gradient descent, however, to avoid millions if not billions of iterations α was scaled according to the size of the Hilbert Matrix. The exact scaling and the number of iterations is shown below:

Matrix Dimensions	4	8	12	32	40
Alpha Scaling	1	0.8	2.0001	2.0001	2.0001
Iterations	1.24E+05	2.75E+06	3.52E+06	3.52E+06	3.52E+06

Comparing the results, shows a relative similarity in accuracy between the two methods when the Hilbert Matrix is small. As the Hilbert Matrix increased in size I was not able to get the Gradient Descent method to converge in a reasonable number of iterations (< 200 million) so I scaled α to get it to converge more quickly. This had the result of ravaging the accuracy of the residuals.

Residuals	Hilbert Matrix Dimensions (NxN)				
Method	4	8	12	32	40
Conjugate Gradient	7.11E-15	6.36E-12	1.98E-08	4.50E-06	1.50E-05
Gradient Descent	7.51E-13	1.29E-12	NaN	NaN	5.55E+153

V. Conclusion

I found the various types of iterative methods to be interesting. I've studied various types of Gradient Descent in other courses, and it was a good review. I'm slightly disappointed in the results of part 4, but perhaps with additional training time (hours and hours), a more powerful machine, or other forms of accelerations we could see if there was similar accuracy. I gather the point though, it would take so many more iterations to get anywhere close to the level of accuracy provided by Conjugate Gradient Codes.