Exploring GMRES-based Iterative Refinement for Solving Linear Systems

Jingxuan Liu (400490045)

liu1819@mcmaster.ca

1 Introduction

Solving the system of linear equations is a common task in science, engineering and other fields. The equation is $Ax = b$, A is $n \times n$ matrix, b is given vector, and x is the vector that needs to be solved. There are roughly two kinds of methods for solving the problem; one is a direct method (e.g., LU, QR and Cholesky decomposition), and another is an iterative method (e.g., Jacobi, Gauss-Seidel and Conjugate Gradient).

In this report, we will primarily discuss the concept of two iterative methods for solving systems of linear equations. The first method, iterative refinement, aims to enhance the accuracy of numerical solutions. It starts by obtaining an approximate solution and then iteratively refines it to mitigate rounding errors. The second method, the Generalized Minimal Residual Method (GMRES), was developed by Yousef Saad and Martin H. Schultz in 1986. GMRES operates by iteratively constructing a Krylov subspace and the initial guess's residual. The method then finds a solution within this subspace that minimizes the norm of the residual across all previous iterations. This approach is particularly effective for solving large, sparse systems where traditional methods may falter.

Although GMRES itself is an independent iterative solver that does not rely on the iterative refinement process, in some cases, combining GMRES with iterative refinement can bring additional benefits. This report will explore the GMRES-based Iterative Refinement for solving the linear system with non-symmetric and ill-conditioned matrix and analyze its performance compared to direct methods like LU decomposition.

2 Interpretations of Method

2.1 GMRES

To solve the problem, $Ax = b$ by generalized minimal residual method. First, we construct the Krylov subspace for the given matrix A and vector b, we have:

$$\mathcal{K}_n(A, r_0) = \mathrm{span}\{r_0, Ar_0, A^2r_0, \ldots, A^{n-1}r_0\}$$

where residual $r_0 = b - Ax_0$ is the initial error obtained by starting choosing the initial guess x_0 for the solution, and n is a positive integer, the GMRES suppose to find a solution $x_n \in x_0 + K_n$ by minimize the residual r_n. We use Arnoldi iterative to expand the Krylov subspace K_n by generating orthogonal basis vectors $span\{q_1, q_2, ..., q_n\}$, where $q_1 = \frac{r_0}{\|r_0\|}$. For each vector q, we use Gram-Schmidt orthogonalization process:

$$w = Aq_i,$$

$$h_{ji} = q_j^T w,$$

$$w = w - \sum_{j=1}^{i} h_{ji} q_j,$$

Normalize w to find next basis vector:

$$h_{i+1,i} = \|w\|,$$

$$q_{i+1} = \frac{w}{h_{i+1,i}}.$$

This will update the orthogonal vector q_i, and a coefficient $h_{i+1,j}$, which effectively build up the the columns of of matrix Q_{n+1} and corresponding Hessenberg matrix $\tilde{H}_n$, This lead us to relation:

$$AQ_n = Q_{n+1}\tilde{H}_n$$

And the solution we trying to find can be written as $x_n = x_0 + Q_n y_n$, where y_n is a vector contains optimal coefficients found within the Krylov subspace, they define how to combine the basis vector in Q_n to minimize the residual $\|b - Ax_n\|$ and we can rewrite it:

$$\begin{aligned}
\|r_n\| &= \|b - Ax_n\| \\
&= \|r_0 - A(x_0 + Q_n y_n)\| \\
&= \|\beta q_1 - AQ_n y_n\|, \text{ where } \beta = \|r_0\| \\
&= \|\beta q_1 - Q_{n+1}\tilde{H}_n y_n\| \\
&= \|Q_{n+1}(\beta e_1 - \tilde{H}_n y_n)\|, \text{ where } Q_{n+1} \text{ is orthogonal} \\
&= \|\beta e_1 - \tilde{H}_n y_n\|
\end{aligned}$$

Vector e_1 represents the first canonical basis vector $[1, 0, ..., 0]$ in Euclidean space to target initial residual, it has a 1 in the first component and 0s in all other components. This leads to the minimization problem for GMRES as:

$$\min_{y \in \mathbb{R}^m} \|\beta e_1 - \tilde{H}_n y_n\|,$$

The computational step for GMRES yields below in Algorithm 1.

Algorithm 1 GMRES Method

$r \leftarrow b - Ax$, $\beta \leftarrow \|r\|$, $q_1 \leftarrow r/\beta$
for $i = 1, 2, \ldots, n$ **do**
    $w_i \leftarrow Aq_i$
    $h_{ji} \leftarrow \langle w_i, q_j \rangle$, $w_i \leftarrow w_i - h_{ji}q_j$
    $h_{i+1,i} \leftarrow \|w_i\|$
    if $h_{i+1,i} == 0$ **then**
        $n \leftarrow i$, **break**
    end if
    $q_{i+1} \leftarrow w_i/h_{i+1,i}$
end for
$H_n \leftarrow \{h_{ji}\}_{1 \le j \le n+1, 1 \le i \le n}$
$y_n \leftarrow \arg\min_y \|\beta e_1 - H_n y\|$
Update: $x \leftarrow x + Q_n y_n$

2.2 GMRES based Iterative Refinement

Iterative refinement (IR) based on the Generalized Minimal Residual Method (GMRES) is an advanced numerical technique designed based on LU-based IR to solve linear systems of equations $Ax = b$ with enhanced stability and accuracy, particularly the systems that are ill-conditioned.

The initial step of the GMRES-based IR algorithm involves computing the LU factorization of matrix A, providing an initial approximation to the solution of the linear system $Ax = b$. However, unlike the traditional LU-based IR approach, each iteration of the GMRES-based IR does not directly utilize the LU decomposition to solve the correction equation. In each iteration, the algorithm first computes the residual $r_i = b - Ax_i$, which assesses the deviation of the current solution x_i from the true solution. Next, to adjust x_i, the correction equation $A\delta_i = r_i$ is solved. Here lies the critical difference from LU-based IR, we use the GMRES to solve correction equation, facilitated by a precondition derived from the

LU factorization, specifically $U^{-1}L^{-1}$. In LU-based IR, the correction equation $LUd_i = r_i$ is solved directly using LU decomposition. This direct method is effective for symmetric or well-conditioned matrices but might struggle with non-symmetric or ill-conditioned matrices. In contrast, the preconditioned correction equation $U^{-1}L^{-1}Ad_i = U^{-1}L^{-1}r_i$ is employed in GMRES-based IR. By applying the $U^{-1}L^{-1}$ preconditioner to both A and r_i, the GMRES algorithm can improve the condition number of the matrix involved in the iterations. This approach reduces the influence of condition numbers and enhances both the stability and efficiency of the solution process, making GMRES particularly suitable for non-symmetric and ill-conditioned systems. Finally, the solution is updated to $x_{i+1} = x_i + \delta_i$. this step improves the solution by adding the correction δ_i to the current solution x_i, aiming to reduce the residual in subsequent iterations further.

The algorithm also uses a Mixed-precision (MP) setting to improve computational and memory efficiency while maintaining numerical stability. As GMRES-based IR algorithm described in [2] and [1] in A common way of doing this is using a lower precision for the LU factorization, and a higher precision during the GMRES process.

The computational method for a common GMRES-based Iterative Refinement is described in Algorithm 2 below. The algorithm solves $Ax = b$ in two precision: u_l (lower precision) and u_h (higher precision).

Algorithm 2 GMRES-based Iterative Refinement (MP)

1: Compute the factorization $A = LU$ in precision u_l.
2: Solve $LUx = b$ by substitution in precision u_l.
3: **for** $i = 1$ to $imax$ or until converged **do**
4:     Compute $r_i = b - Ax_i$ in precision u_h.
5:     Solve $U^{-1}L^{-1}Ad_i = U^{-1}L^{-1}r_i$ by GMRES in precision u_h
6:     Compute $x_{i+1} = x_i + d_i$ in precision u_l.
7: **end for**

2.3 Comparison and Theoretical Result

The theorem presented in [2] states that for LU-based Iterative Refinement, the convergence condition is $\kappa(A)u < 1$, where $\kappa(A)$ is the condition number of matrix A and u is precision for LU factorization. The precision level u directly impacts the convergence speed of the algorithm and its ability to converge to an accurate solution. In pursuit of speed optimization, precision may be sacrificed. Computing the LU decomposition at a precision lower than the standard u can accelerate the process, but this may introduce numerical instability. The algorithm needs to balance between computational precision and numerical stability.

But for GMRES-based IR, the convergence condition can be $\kappa(A)^2u^2(u + \kappa(A)u) < 1$. This approach offers greater flexibility, allowing for the optimization of algorithm performance and accuracy through the adjustment of various precision parameters. This is particularly useful when dealing with highly ill-conditioned or nearly singular linear systems, as it permits the use of different precisions at different computational stages, thereby enhancing efficiency while maintaining convergence.

The most significant difference between LU-based IR and GMRES-based IR is that the latter uses Krylov subspace techniques to solve for a stable approximation of the correction equation and can significantly take advantage of mixed precision. Also, in contract, GMRES-based IR can reduce computational costs by performing the LU decomposition step at a lower precision during the iteration process. In a world, We expect GMRES-based IR should have a clear better performance when handling the non-symmetric and ill-conditioned matrix.

3 Experiment and Analysis Result

In my work, I write two algorithms to analyze the result. For my GMRES-based IR algorithm, I set lower precision to init the LU factorization and used higher precision during the iterative process, in python, this is made by np.float64 and np.float32. The same mixed-precision setting is also applied to the LU-based IR algorithm. The whole experiment is perform on a personal PC with Intel Core i5 9_{gen} CPU.

3.1 Iteration Number

The number of iterations required for convergence with a stop condition when $\frac{\|b-Ax\|}{\|b\|} \leq \text{tol}$, where tolerance is 2.2204×10^{-15} for both methods, result is presented in table 1. And all matrices in this table are of size $n = 10000$. The table 2 displays the iterations required for various matrix sizes, noting that the condition number is not constant in these cases.

Condition Number	GMRES-base IR	LU-based IR
1.56e6	3	5
1.70e6	2	4
1.837e6	2	5
1.95e6	2	5
2.16e6	2	5
2.59e6	2	4
2.85e6	2	5
4.38e6	3	5
1.01e7	2	5
1.28e7	2	6

Table 1: Iterations for LU and GMRES-based IR convergence in varying condition number.

Matrix Size	GMRES-base IR	LU-based IR
50	2	3
500	2	3
1000	2	3
2000	2	4
5000	2	4
80000	2	5
12000	2	5
15000	2	8

Table 2: Iteration for LU and GMRES-based IR convergence in varying matrix size.

The results demonstrate that across various matrix sizes and condition numbers, the GMRES-based Iterative Refinement (IR) required fewer iterations to achieve convergence compared to the LU-based IR approach. Remarkably, as both the matrix size and condition number increase, the GMRES-based IR maintains stable convergence within approximately two iterations. On the other hand, the number of iterations required for the LU-based method shows a noticeable increase with the size of the matrix, yet it is not significantly affected by the condition number.

3.2 Accuracy and Effectiveness

The results of the previous experiment indicate that, within our testing environment, GMRES-based Iterative Refinement consistently converges within approximately two iterations, whereas LU-based Iterative Refinement typically achieves convergence within five iterations. And it's good to notice that both methods are able to achieve a very similar accuracy. To further assess the superior performance of GMRES-based IR, we will conduct a comparative analysis of both methods under the constraint of a 11 iterations.

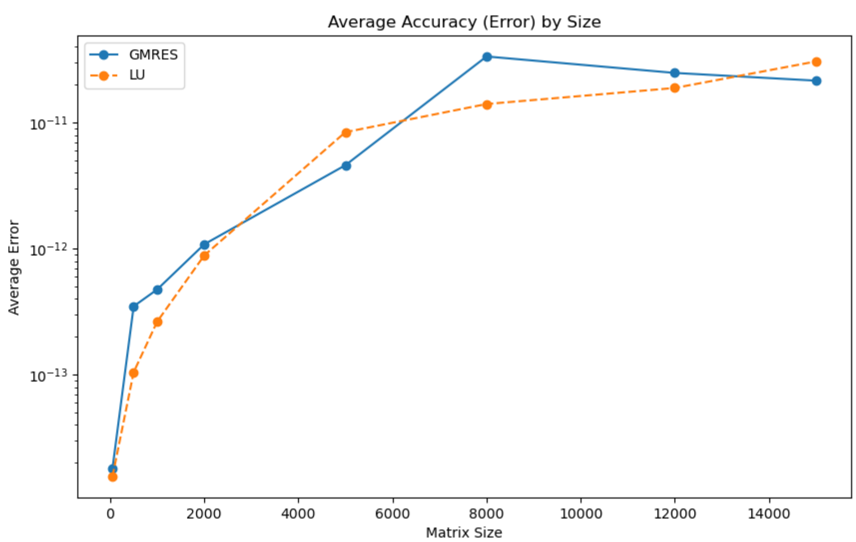

Figure 1: Average error for LU and GMRES-based IR with different matrix size.

The Figure 1 and Figure 2 illustrates that as the matrix size increases, the error and computational time associated with both LU and GMRES-based Iterative Refinement (IR) methods also increases, yet both achieve similar levels of error, but GMRES-based IR have higher computational time than LU-based IR. From the previous analysis of iterations, it is evident that both methods have converged by this number of iterations. This indicates that despite the increase in matrix size, both methods continue to converge with a reasonable error rate.

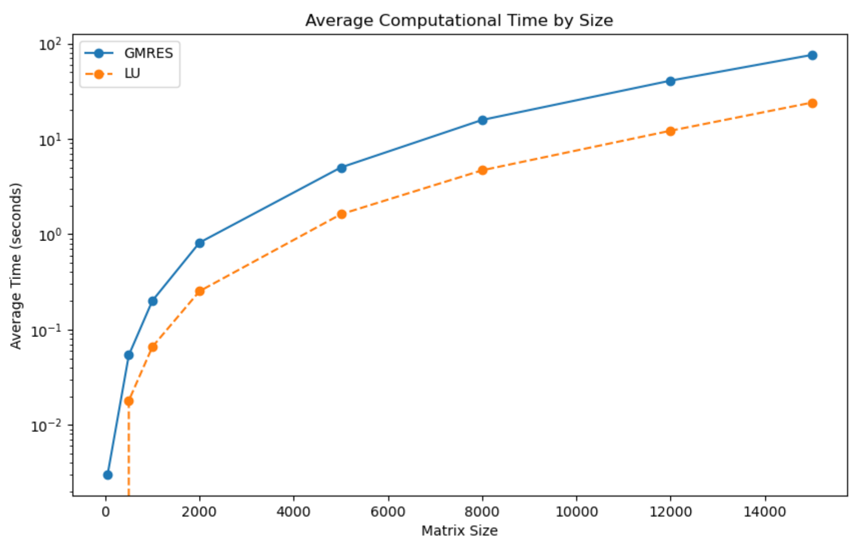

Figure 2: Computational time for LU and GMRES-based IR with different matrix size.

The Figure 3 demonstrates that as the matrix condition number increases, the error associated with both methods also rises; however, due to rounding errors, this increase is not linear. It is difficult to determine which method performs better under converged conditions, since there is no guaranty that the error of GMRES-based IR is better than LU-based IR. Therefore, we can conclude that both methods are capable of converging with good accuracy as the condition number increases. Again, in table 4 it is clear that the computational time for GMRES-based Iterative Refinement is higher than that for LU-based Iterative Refinement. However, the condition number of the matrix does not significantly affect the effectiveness of either method.

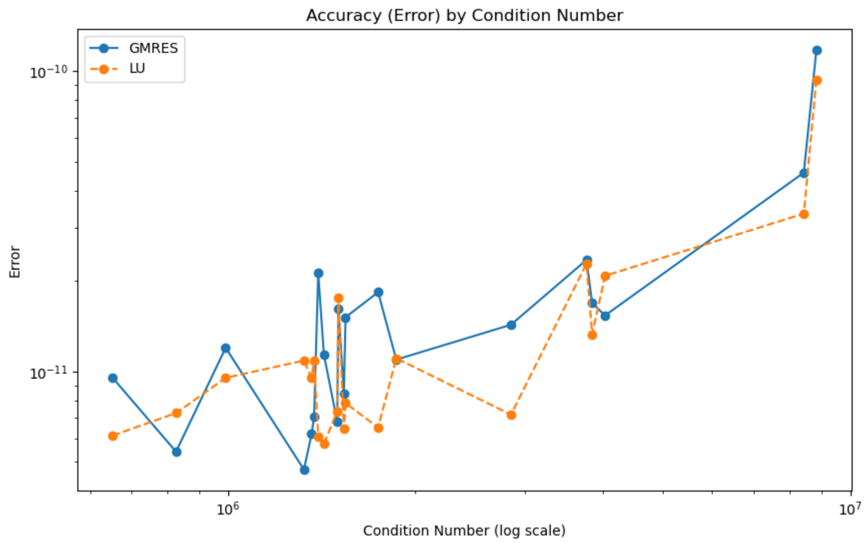

Figure 3: Error for LU and GMRES-based IR with different condition number.

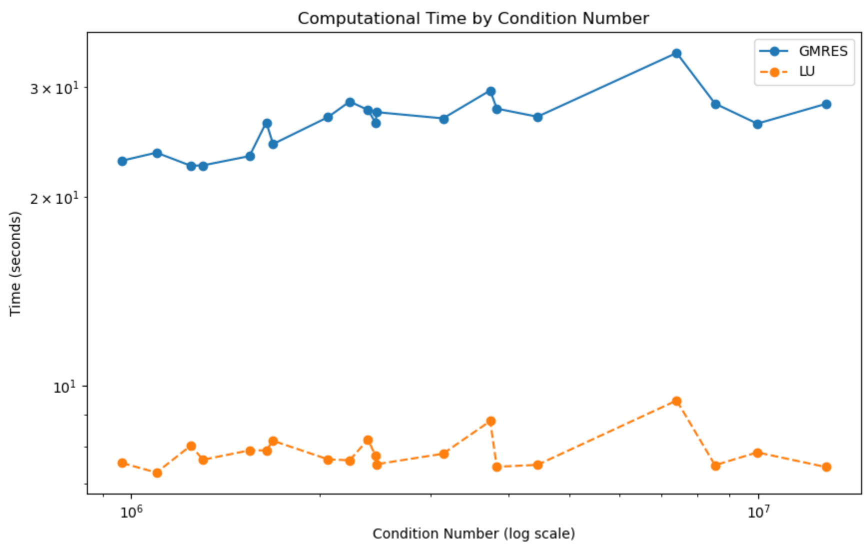

Figure 4: Computational time for LU and GMRES-based IR with different condition number.

3.3 Ill-conditioned non-symmetric Matrix

To further analyze the performance of GMRES-based Iterative Refinement (IR), we created a 1000×1000 non-symmetric matrix with a condition number of 1×10^8. As depicted in Figure 5, despite the matrix being non-symmetric and ill-conditioned, the GMRES-based IR was still able to converge within four iterations. In contrast, the LU-based IR failed to converge, with the relevant error continuing to increase.

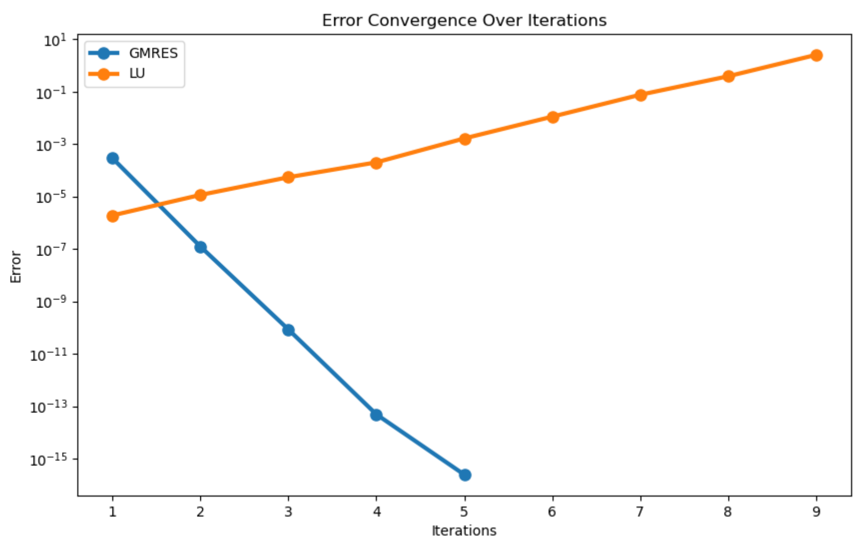

Figure 5: Iterations for LU and GMRES-based IR with $\kappa(A) = 1e^8$.

4 Conclusion

This work has demonstrated that GMRES-based Iterative Refinement (IR) achieves stable convergence in fewer iterations compared to LU-based IR. However, the computational time required for the GMRES approach is longer than that for the LU cases. This discrepancy can likely be attributed to the GMRES algorithm involving more computational steps than the direct use of LU decomposition for solving a linear system. With matrices of reasonable size and condition number, both methods can converge to satisfactory results. Remarkably, even in cases involving non-symmetric and ill-conditioned matrices, GMRES-based IR is still able to converge in just a few iterations, a feat that LU-based IR cannot achieve. Although the GMRES-based Iterative Refinement method may require longer computational times, it demonstrates significant advantages when dealing with specific types of linear systems, particularly in handling non-symmetric and highly ill-conditioned matrices, effectively enhancing the stability and accuracy of the solutions.

4.1 Future work

In the current analysis, I encounter technical difficulties in controlling the condition number due to the challenges inherent in generating matrix elements. It is worthwhile to explore methods for creating matrices with a specified condition number at a reasonable computational cost. Additionally, when analyzing non-symmetric, ill-conditioned matrices, where the condition number $\kappa(A) \geq 1e^9$, the GMRES algorithm fails to converge. This highlights the need to investigate potential improvements in the parameter settings of the GMRES algorithm to enhance its effectiveness under such challenging conditions.

References

[1] A Haidar, H Bayraktar, S Tomov, J Dongarra, and NJ Higham. Mixed-precision iterative refinement using tensor cores on gpus to accelerate solution of linear systems. *Proc Math Phys Eng Sci*, 476(2243):20200110, Nov 2020. Epub 2020 Nov 25.

[2] Nicholas J. Higham and Theo Mary. Mixed precision algorithms in numerical linear algebra. *Acta Numerica*, 31:347–414, 2022.