Math 105AL – Lab 7

Haoyu Zhang, 25327328

1 Objective

In this lab, I used the Jaboci and Gauss-Seidel iterative methods to solve two different linear systems. I investigated how each method behaves with the same initialization which is 0. I also compared the convergence rates, computed the spectral radii of the corresponding iteration matrices, and then examined whether the experimental results match theoretical expectations. Then, I generated two iteration V.S. error plots for each system and the method.

2 Procedure

First of all, I implemented the codings named "jacobi.m" and "gauss-seidel.m", which follow the standard iterative formulas:

$$x^{(k+1)} = -D^{-1}(L+U)x^{(k)} + D^{-1}b \quad \text{(Jacobi)},$$

$$x^{(k+1)} = -(D+L)^{-1}Ux^{(k)} + (D+L)^{-1}b \quad \text{(Gauss–Seidel)}.$$

For both systems, I initialized with x_0 = 0 and ran each method for at most 25 iterations. At each step I recorded the approximation error with e^(k) = || x^(k) – x^* ||∞.

I also computed the iteration matrices T_J and T_GS using: A = D + L + U; T_J = -D^-1(L+U); T_GS = -(D+L)^-1 U. And then calculated their spectral radii using "max(abs(eig(T)))".

Finally, I plotted the iteration v.s. approximation error on a semilogarithmic scale for both two systems.

3 Results

```
rhoJ = 1.1180
rhoGS = 0.5000
rhoJ = 1.2329e-05
rhoGS = 2
```

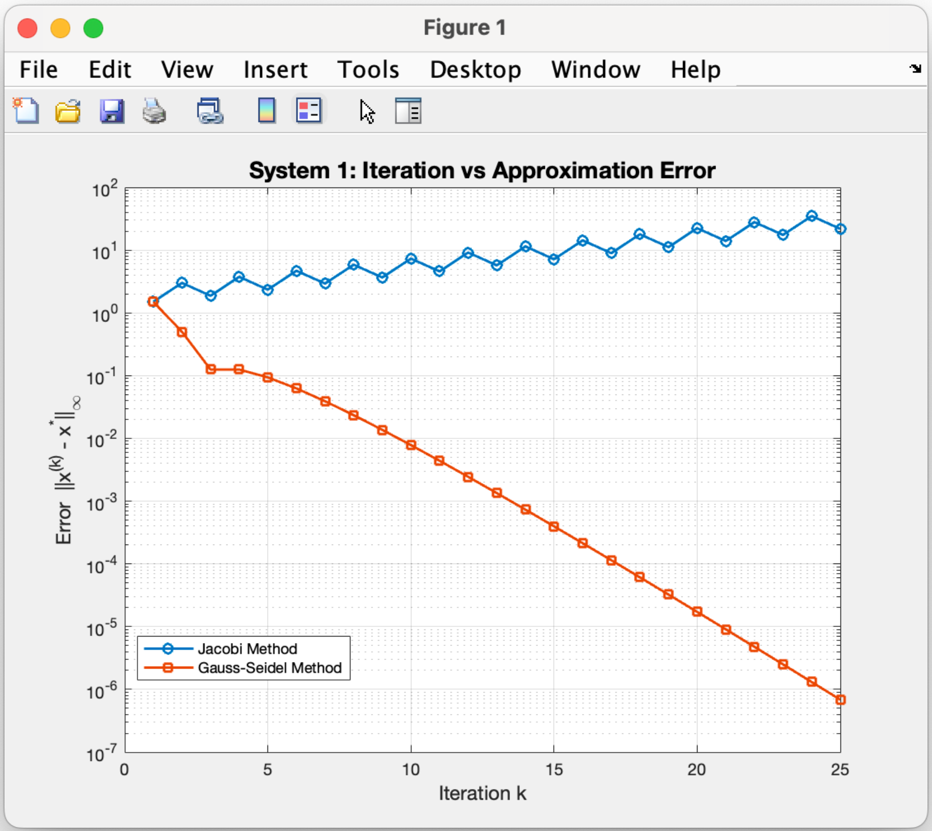
The Above screenshot accurately showed that the spectral radii of Jacobi and Gauss-Seidel for system 1 and system two separately. System 1: A1 and b1 referenced in the doc.

$$\begin{bmatrix} 2 & -1 & 1 \\ 2 & 2 & 2 \\ -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -1 \\ 4 \\ -5 \end{bmatrix}$$

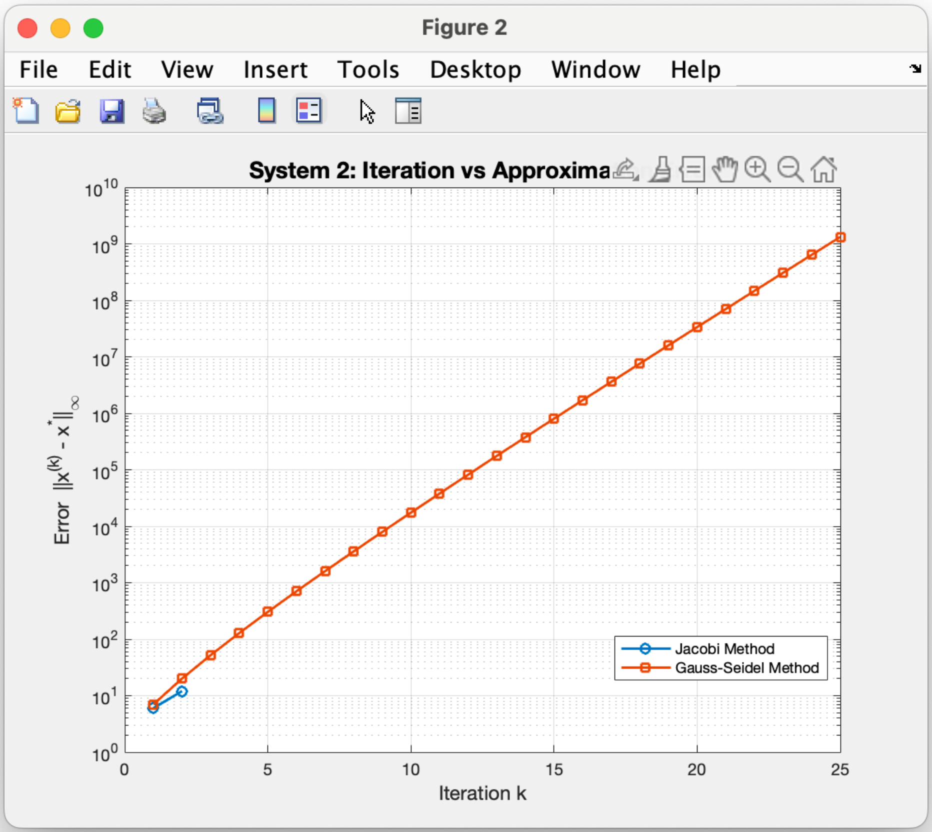
System 2: A2 and b2 referenced in the doc.

$$\begin{bmatrix} 1 & 2 & -2 \\ 1 & 1 & 1 \\ 2 & 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7 \\ 2 \\ 5 \end{bmatrix},$$

Here is the graph of the iteration v.s. approximation error for system 1:



Here is the graph of the iteration v.s. approximation error for system 2:



Summary:

(Q2) For system 1, because $p(T\_J) > 1$, the Jacobi method diverges for this system. The error plot confirms that the Jacobi error grows and does not settle toward zero.

(Q3) Because $p(T\_GS) = 0.5 < 1$, the Gauss-Seidel method converges. The error decreases roughly by a factor of 0.5 for each iteration, matching the expected geometric convergence rate. The difference occurs because the matrix is not diagonal-dominant.

For system 2, because $p(T\_J) = 0$, the Jacobi iteration matrix is nilpotent and Jacobi converges extremely quickly. Because $p(T\_GS) = 2 > 1$, the Gauss – Seidel method diverges. The error grows exponentially. I think this system showed the opposite behavior of system 1.

Jacobi is the better method.

(Q4) (a): System1: $p(T\_J) = 1.118 > 1$; $p(T\_GS) = 0.5 < 1$. System2: $p(T\_J) = 0 < 1$;

p(T_GS) = 2 > 1. These results directly classify convergence or divergence for each method.

(b): I generated two plots: for plot1, system1, GS converges, Jacobi diverges. For plot2, system2, Jacobi converges rapidly, and GS diverges.

4 Conclusion

(Q5) (a). For system1: converged one is Gauss-Seidel; The Jacobi did not converge. Gauss-Seidel is faster. For system2: Jacobi converged. Gauss-Seidel did not converge. Jacobi is faster.

(b). The experiments perfectly match the theoretical convergence rule:

p(T) < 1 if and only if the method converges.

In System1, p(T_GS) < 1 (converges) and p(T_J) > 1 (diverges).

In System2, p(T_J) < 1 (converges) and p(T_GS) >1 (diverges).

Therefore, the spectral radius predicts the behavior from the plots.

(c). I think changing x_0 would not change which method converges or diverges. For convergent cases (p<1), the method will still converge to the same solution. For divergent one (p>1), the method will diverge for almost all starting points.

5 Challenges and Bugs

1. When generating the plots, I forgot to use “hold on”.
2. The initial decomposition of A into D, L and U was incorrect. So it caused the wrong spectral radii.
3. I forgot to update. I computed the wrong error using outdated values from the old iterations.
4. My iteration history got overwritten due to reusing variable names. I incautiously write the wrong