Assignment 4

MATH391

Connor Braun

December 7, 2021

Problem 1

Compute the following for matrices

$$A = \begin{bmatrix} 1 & 1 & -1 & 2 \\ 2 & 2 & 4 & 5 \\ 1 & -1 & 1 & 7 \\ 2 & 3 & 4 & 6 \end{bmatrix}, \quad B = \begin{bmatrix} 4 & 1 & 1 & 1 \\ 1 & 3 & 0 & -1 \\ 1 & 0 & 2 & 1 \\ 1 & -1 & 1 & 4 \end{bmatrix}.$$

a)Compute the LU factorization of A with partial pivoting.

We seek to find permutation matrix P, unitary lower triangular matrix L and upper triangular matrix U such that

$$PA = LU.$$

We proceed with the process of Gaussian elimination with partial pivoting on A, noting all row interchanges between steps, and recording Gaussian multipliers below the main diagonal in red in place of zeros.

$$\begin{bmatrix} 1 & 1 & -1 & 2 \\ 2 & 2 & 4 & 5 \\ 1 & -1 & 1 & 7 \\ 2 & 3 & 4 & 6 \end{bmatrix} \xrightarrow{R_1 \leftrightarrow R_2} \begin{bmatrix} 2 & 2 & 4 & 5 \\ 1 & 1 & -1 & 2 \\ 1 & -1 & 1 & 7 \\ 2 & 3 & 4 & 6 \end{bmatrix} \overset{\substack{R_3 - \frac{1}{2}R_1 \\ R_2 - \frac{1}{2}R_1}}{\underset{R_4 - \frac{2}{2}R_1}{\longrightarrow}} \begin{bmatrix} 2 & 2 & 4 & 5 \\ \frac{1}{2} & 0 & -3 & -\frac{1}{2} \\ \frac{1}{2} & -2 & -1 & \frac{9}{2} \\ 1 & 1 & 0 & 1 \end{bmatrix} \xrightarrow{R_2 \leftrightarrow R_3} \begin{bmatrix} 2 & 2 & 4 & 5 \\ \frac{1}{2} & -2 & -1 & \frac{9}{2} \\ \frac{1}{2} & 0 & -3 & -\frac{1}{2} \\ 1 & 1 & 0 & 1 \end{bmatrix} \cdots$$

$$\cdots \overset{\substack{R_3 - \frac{0}{-2}R_2 \\ R_4 - \frac{1}{-2}R_2}}{\longrightarrow} \begin{bmatrix} 2 & 2 & 4 & 5 \\ \frac{1}{2} & -2 & -1 & \frac{9}{2} \\ \frac{1}{2} & 0 & -3 & -\frac{1}{2} \\ 1 & -\frac{1}{2} & -\frac{1}{2} & \frac{13}{4} \end{bmatrix} \xrightarrow{R_4 - \frac{1}{6}R_3} \begin{bmatrix} 2 & 2 & 4 & 5 \\ \frac{1}{2} & -2 & -1 & \frac{9}{2} \\ \frac{1}{2} & 0 & -3 & -\frac{1}{2} \\ 1 & -\frac{1}{2} & \frac{1}{6} & \frac{10}{3} \end{bmatrix}$$

We now apply precisely the same row interchanges on I_4 as we did on the system to find P.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \xrightarrow{R_1 \leftrightarrow R_2} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \xrightarrow{R_2 \leftrightarrow R_3} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = P.$$

Finally, L and U are given by separating our system with all Gaussian multipliers into a unitary lower triangular component and an upper triangular component.

$$\begin{bmatrix} 2 & 2 & 4 & 5 \\ \frac{1}{2} & -2 & -1 & \frac{9}{2} \\ \frac{1}{2} & 0 & -3 & -\frac{1}{2} \\ 1 & -\frac{1}{2} & \frac{1}{6} & \frac{10}{3} \end{bmatrix} \Rightarrow L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ \frac{1}{2} & 0 & 1 & 0 \\ 1 & -\frac{1}{2} & \frac{1}{6} & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 2 & 2 & 4 & 5 \\ 0 & -2 & -1 & \frac{9}{2} \\ 0 & 0 & -3 & -\frac{1}{2} \\ 0 & 0 & 0 & \frac{10}{3} \end{bmatrix}$$

Where matrices P, A, L and U satisfy the LU-factorization equation

$$PA = LU.$$

b) Compute the LDL^T factorization of B.

We seek to find diagonal matrix D and lower unitary triangular matrix L such that

$$B = LDL^T$$

Furthermore, D is determined by the upper triangular matrix U satisfying the LU decomposition of B

$$B = LU$$

where $diag(D) = diag(U)$. We begin with the process of Gaussian elimination without pivoting, recording Gaussian multipliers below the main diagonal in red in place of zeros.

$$\begin{bmatrix} 4 & 1 & 1 & 1 \\ 1 & 3 & 0 & -1 \\ 1 & 0 & 2 & 1 \\ 1 & -1 & 1 & 4 \end{bmatrix} \xrightarrow[R_4 - \frac{1}{4}R_1]{R_2 - \frac{1}{4}R_1 \quad R_3 - \frac{1}{4}R_1} \begin{bmatrix} 4 & 1 & 1 & 1 \\ \frac{1}{4} & \frac{11}{4} & -\frac{1}{4} & -\frac{5}{4} \\ \frac{1}{4} & -\frac{1}{4} & \frac{7}{4} & \frac{3}{4} \\ \frac{1}{4} & -\frac{5}{4} & \frac{3}{4} & \frac{15}{4} \end{bmatrix} \xrightarrow{R_3 - \frac{-1}{11}R_2 \quad R_4 - \frac{-5}{11}R_2} \begin{bmatrix} 4 & 1 & 1 & 1 \\ \frac{1}{4} & \frac{11}{4} & -\frac{1}{4} & -\frac{5}{4} \\ \frac{1}{4} & -\frac{1}{11} & \frac{76}{44} & \frac{28}{44} \\ \frac{1}{4} & -\frac{5}{11} & \frac{28}{44} & \frac{140}{44} \end{bmatrix} \xrightarrow{R_4 - \frac{7}{19}R_3} \begin{bmatrix} 4 & 1 & 1 & 1 \\ \frac{1}{4} & \frac{11}{4} & -\frac{1}{4} & -\frac{5}{4} \\ \frac{1}{4} & -\frac{1}{11} & \frac{76}{44} & \frac{28}{44} \\ \frac{1}{4} & -\frac{5}{11} & \frac{7}{19} & \frac{2464}{836} \end{bmatrix}$$

Now, as was the case in 1.a, L and U are given by separating our system with all Gaussian multipliers into a unitary lower triangular component and an upper triangular component.

$$\begin{bmatrix} 4 & 1 & 1 & 1 \\ \frac{1}{4} & \frac{11}{4} & -\frac{1}{4} & -\frac{5}{4} \\ \frac{1}{4} & -\frac{1}{11} & \frac{76}{44} & \frac{28}{44} \\ \frac{1}{4} & -\frac{5}{11} & \frac{7}{19} & \frac{2464}{836} \end{bmatrix} \Rightarrow L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{4} & 1 & 0 & 0 \\ \frac{1}{4} & -\frac{1}{11} & 1 & 0 \\ \frac{1}{4} & -\frac{5}{11} & \frac{7}{19} & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 4 & 1 & 1 & 1 \\ 0 & \frac{11}{4} & -\frac{1}{4} & -\frac{5}{4} \\ 0 & 0 & \frac{76}{44} & \frac{28}{44} \\ 0 & 0 & 0 & \frac{2464}{836} \end{bmatrix} = \begin{bmatrix} 4 & 1 & 1 & 1 \\ 0 & \frac{11}{4} & -\frac{1}{4} & -\frac{5}{4} \\ 0 & 0 & \frac{19}{11} & \frac{7}{11} \\ 0 & 0 & 0 & \frac{56}{19} \end{bmatrix}$$

But since $diag(U) = diag(D)$ and D is diagonal, we have that

$$D = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & \frac{11}{4} & 0 & 0 \\ 0 & 0 & \frac{19}{11} & 0 \\ 0 & 0 & 0 & \frac{56}{19} \end{bmatrix}$$

Where now the matrices L and D satisfy the factorization of symmetric matrix B given by

$$B = LDL^T$$

c) Compute the Cholesky factorization of B.

We seek the lower triangular matrix G with positive entries on the main diagonal such that

$$B = GG^T.$$

Such a matrix is guaranteed to exist only when B is positive definite. Let η_i, $0 \leq i \leq 4$ be the 4 principle minors of B. Then, since B is symmetric, B is positive definite if and only if $\eta_i > 0$, $0 \leq i \leq 4$. Then, we proceed by computing the principle minors of B.

$$\eta_1 = 4 > 0$$

$$\eta_2 = \begin{vmatrix} 4 & 1 \\ 1 & 3 \end{vmatrix} = 12 - 1 = 11 > 0$$

$$\eta_3 = \begin{vmatrix} 4 & 1 & 1 \\ 1 & 3 & 0 \\ 1 & 0 & 2 \end{vmatrix} = 4\begin{vmatrix} 3 & 0 \\ 0 & 2 \end{vmatrix} - \begin{vmatrix} 1 & 0 \\ 1 & 2 \end{vmatrix} + \begin{vmatrix} 1 & 3 \\ 1 & 0 \end{vmatrix} = 24 - 2 - 3 = 19 > 0$$

$$\eta_4 = \begin{vmatrix} 4 & 1 & 1 & 1 \\ 1 & 3 & 0 & -1 \\ 1 & 0 & 2 & 1 \\ 1 & -1 & 1 & 4 \end{vmatrix} = 4\begin{vmatrix} 3 & 0 & -1 \\ 0 & 2 & 1 \\ -1 & 1 & 4 \end{vmatrix} - \begin{vmatrix} 1 & 0 & -1 \\ 1 & 2 & 1 \\ 1 & 1 & 4 \end{vmatrix} + \begin{vmatrix} 1 & 3 & -1 \\ 1 & 0 & 1 \\ 1 & -1 & 4 \end{vmatrix} - \begin{vmatrix} 1 & 3 & 0 \\ 1 & 0 & 2 \\ 1 & -1 & 1 \end{vmatrix}$$

$$
\begin{aligned}
&= 4\left(3\begin{vmatrix}2 & 1\\ 1 & 4\end{vmatrix} - \begin{vmatrix}0 & 2\\ -1 & 1\end{vmatrix}\right) - \left(\begin{vmatrix}2 & 1\\ 1 & 4\end{vmatrix} - \begin{vmatrix}1 & 2\\ 1 & 1\end{vmatrix}\right) + \left(\begin{vmatrix}0 & 1\\ -1 & 4\end{vmatrix} - 3\begin{vmatrix}1 & 1\\ 1 & 4\end{vmatrix} - \begin{vmatrix}1 & 0\\ 1 & -1\end{vmatrix}\right) - \left(\begin{vmatrix}0 & 2\\ -1 & 1\end{vmatrix} - 3\begin{vmatrix}1 & 2\\ 1 & 1\end{vmatrix}\right)\\
&= 4(3(7)-2) - (7+1) + (1 - 3(3) + 1) - (2 - 3(-1))\\
&= 56 > 0
\end{aligned}
$$

So, since all positive minors are strictly greater than zero, B is positive definite. We can then use matrices L and D as computed in 1.b to find G. Let $\tilde{D}$ be the diagonal matrix obtained by taking the square root of each element in D. Then we have

$$G = L\tilde{D}.$$

We can then compute G directly.

$$
G = \begin{bmatrix} 1 & 0 & 0 & 0\\ \frac{1}{4} & 1 & 0 & 0\\ \frac{1}{4} & -\frac{1}{11} & 1 & 0\\ \frac{1}{4} & -\frac{5}{11} & \frac{7}{19} & 1 \end{bmatrix}
\begin{bmatrix} 2 & 0 & 0 & 0\\ 0 & \frac{\sqrt{11}}{2} & 0 & 0\\ 0 & 0 & \frac{\sqrt{19}}{\sqrt{11}} & 0\\ 0 & 0 & 0 & \frac{2\sqrt{14}}{\sqrt{19}} \end{bmatrix}
= \begin{bmatrix} 2 & 0 & 0 & 0\\ \frac{1}{2} & \frac{\sqrt{11}}{2} & 0 & 0\\ \frac{1}{2} & -\frac{\sqrt{11}}{22} & \frac{\sqrt{19}}{\sqrt{11}} & 0\\ \frac{1}{2} & -\frac{5\sqrt{11}}{22} & \frac{7\sqrt{19}}{19\sqrt{11}} & \frac{2\sqrt{14}}{\sqrt{19}} \end{bmatrix}
$$

Where G satisfies the Cholesky decomposition for matrix B given by

$$B = GG^T$$

Problem 2

a) Write the algorithm that computes matrices L and U such that for tridiagonal matrix A we can write $A = LU$.

Furthermore, the question specifies a form of L which precludes partial pivoting. Namely, that L is a unitary lower triangular matrix of all zeros except for on the main diagonal and first subdiagonal. Partial pivoting could result in nonzero entries of L beneath the first subdiagonal, so we exclude it from our algorithm. We can induce an algorithm by considering the case of 3×3 tridiagonal matrix B given by

$$
\begin{bmatrix} a_1 & c_1 & 0\\ b_2 & a_2 & c_2\\ 0 & b_3 & a_3 \end{bmatrix}
$$

where all of a_i, b_i and c_i are real numbers for $i \in \mathbb{N}$, $1 \le i \le 3$. Simply applying the Gaussian elimination algorithm and recording Gaussian multipliers below the main diagonal in red in place of zeros, we have

$$
\begin{bmatrix} a_1 & c_1 & 0\\ b_2 & a_2 & c_2\\ 0 & b_3 & a_3 \end{bmatrix}
\xrightarrow[R_3 - 0R_1]{R_2 - \frac{b_2}{a_1}R_1}
\begin{bmatrix} a_1 & c_1 & 0\\ \frac{b_2}{a_1} & a_2' & c_2\\ 0 & b_3 & a_3 \end{bmatrix}
\xrightarrow{R_3 - \frac{b_3}{a_2'}}
\begin{bmatrix} a_1 & c_1 & 0\\ \frac{b_2}{a_1} & a_2' & c_2\\ 0 & \frac{b_3}{a_2'} & a_3' \end{bmatrix}
$$

Where now we have the LU factorization of B such that $B = L_B U_B$ is given by

$$
L_B = \begin{bmatrix} 1 & 0 & 0\\ \frac{b_2}{a_1} & 1 & 0\\ 0 & \frac{b_3}{a_2'} & 1 \end{bmatrix}, \qquad
U_B = \begin{bmatrix} a_1 & c_1 & 0\\ 0 & a_2' & c_2\\ 0 & 0 & a_3' \end{bmatrix}
$$

Which is a solution in precisely the form required. However, we require that $a_1 \neq 0$ and $a_j' \neq 0$, $j \in \mathbb{N}$, $2 \le j \le 3$. Assuming this to be the case, we can see that

$$a_1 = a_1$$

$$a_2' = a_2 - \frac{b_2}{a_1}c_1$$
$$a_3' = a_3 - \frac{b_3}{a_2'}c_2$$

Which we can write as the recursive formula

$$a_1 = a_1 = a_1'$$
$$a_j' = a_j - \frac{b_j}{a_{j-1}'}c_{j-1}, \qquad 2 \leq j \leq 3$$

Then, for the general $n \times n$ tridiagonal matrix given by

$$A = \begin{bmatrix} u_1 & w_1 & 0 & & & \cdots & \cdots & \cdots & & 0 \\ v_2 & u_2 & w_2 & 0 & & & & & & \\ 0 & v_3 & u_3 & w_3 & 0 & & & & & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & & & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & & & \vdots \\ \vdots & & & \ddots & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & & & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ & & & & & 0 & v_{n-2} & u_{n-2} & w_{n-2} & 0 \\ & & & & & & 0 & v_{n-1} & u_{n-1} & w_{n-1} \\ 0 & & & & & & & 0 & v_n & u_n \end{bmatrix}$$

where we seek unitary lower triangular matrix L and upper triangular matrix U given by

$$L = \begin{bmatrix} 1 & 0 & & & \cdots & \cdots & 0 \\ l_2 & 1 & 0 & & & & \\ 0 & l_3 & 1 & 0 & & & \\ \vdots & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \vdots \\ & & & 0 & l_{n-1} & 1 & 0 \\ 0 & & & & 0 & l_n & 1 \end{bmatrix}, \quad U = \begin{bmatrix} z_1 & w_1 & & & \cdots & \cdots & 0 \\ 0 & z_2 & w_2 & & & & \\ 0 & 0 & z_3 & w_3 & & & \\ \vdots & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \vdots \\ & & & 0 & 0 & z_{n-1} & w_{n-1} \\ 0 & & & & 0 & 0 & z_n \end{bmatrix}$$

which satisfies the LU decomposition of A given by

$$A = LU$$

we do so by the algorithm

$$\begin{cases} z_1 & = u_1 \\ z_i & = u_i - \frac{v_i}{z_{i-1}}w_{j-1}, \qquad 2 \leq i \leq n \\ l_j & = \frac{v_j}{z_{i-1}}, \qquad 2 \leq j \leq n \end{cases}$$

Importantly, we note that this algorithm fails if any of z_1, z_2, $\ldots$, z_{n-1} is zero.

b)Write Matlab function $TriLU()$ which computes LU factorization of an arbitrary tridiagonal matrix according to the algorithm proposed in 2.a and the additional specifications in the assignment.

```
function [l, u] = TriLU(a, b, c, n)
    l = zeros(n - 1, 1);
    u = zeros(n, 1);
```

```
    u(1) = a(1);

    for i = 2:(n)
        if i ~= n
            assert(u(i - 1) ~= 0, sprintf("TriLU failure; pivot %d is 0", i - 1))
        end
        gauss_multiplier = b(i - 1)/u(i - 1);

        l(i - 1) = gauss_multiplier;
        u(i) = a(i) - gauss_multiplier*c(i - 1);
    end
end
```

Problem 3

Show that if $A = (a_{ij})_{1 \leq i,j \leq n}$ is an $n \times n$ matrix, then

$$|||A|||_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^{n} |a_{ij}|.$$

Proof. Let $x \in \mathbb{R}^n$, where $x = [x_1 \ x_2 \ \dots \ x_n]^T$. Then we proceed by first stating the definition of a matrix norm.

$$|||A|||_\infty = \max_{\substack{x \in \mathbb{R}^n \\ x \neq 0}} \frac{\|Ax\|_\infty}{\|x\|_\infty}$$

Let $x_m = \max_{1 \leq i \leq n} |x_i|$ where since $x \neq 0$, $0 < x_m$. Then, we can proceed to expand the system Ax and simplify by the definition of the infinity norm.

$$|||A|||_\infty = \max_{\substack{x \in \mathbb{R}^n \\ x \neq 0}} \left(\frac{\left\| \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \right\|_\infty}{\max_{1 \leq i \leq n} |x_i|} \right) = \max_{\substack{x \in \mathbb{R}^n \\ x \neq 0}} \left(\frac{\max_{1 \leq i \leq n} \left| \sum_{j=1}^{n} a_{ij} x_j \right|}{x_m} \right)$$

$$= \max_{\substack{x \in \mathbb{R}^n \\ x \neq 0}} \left(\max_{1 \leq i \leq n} \left| \sum_{j=1}^{n} a_{ij} \frac{x_j}{x_m} \right| \right)$$

We know that the maximum possible value of $\frac{x_j}{x_m}$, $j = 1, 2, \dots, n$ is 1, which occurs when both $|x_j| = x_m$ and x_j has the same sign as a_{ij}. Hence the vector $x \in \mathbb{R}^n$ which maximizes this sum is that for which $\forall_j$, $\frac{x_j}{x_m} = \pm 1$ and $a_{ij} x_j \geq 0$. Letting x have these properties, we get that $\forall_j$, $|a_{ij}| = a_{ij} \frac{x_j}{x_m}$ so the above simplifies to the equality of interest:

$$|||A|||_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^{n} |a_{ij}|.$$

Problem 4

Write Matlab functions *myJacobi*() and *myGaussSeidel* which iteratively approximate $x \in \mathbb{R}^n$ in linear system of equations $Ax = b$, where A is an $n \times n$ matrix and $b \in \mathbb{R}^n$. Functions must meet specifications in assignment.

```
function [x_n, n] = myJacobi(A, b, x_0, nmax, tol)
    dim = numel(b);
    x_n = x_0;
    for n = 1:nmax
        for i = 1:dim
            row_operator = 0;
            for j = 1:dim
                if j ~= i
                    row_operator = row_operator + A(i, j)*x_0(j);
                end
            end
            x_n(i) = (b(i) - row_operator)/A(i, i);
        end
        if norm(x_n - x_0)/norm(x_n) < tol
            disp("myJacobi return condition: ||x_k - x_k-1||/||x_k|| < tol")
            return
        end
        x_0 = x_n;
    end
    disp("myJacobi did not achieve convergence criterion within iteration allowance")
end

function [x_n, n] = myGaussSeidel(A, b, x_0, nmax, tol)
    dim = numel(b);
    x_n = x_0;
    for n = 1:nmax
        for i = 1:dim
            row_operator = 0;
            for j = 1:dim
                if j ~= i
                    row_operator = row_operator + A(i, j)*x_n(j);
                end
            end
            x_n(i) = (b(i) - row_operator)/A(i, i);
        end
        if norm(x_n - x_0)/norm(x_n) < tol
        disp("myGaussSeidel return condition: ||x_k - x_k-1||/||x_k|| < tol")
        return
        end
        x_0 = x_n;
    end
    disp("myGaussSeidel did not achieve convergence criterion within iteration allowance")
end
```

Problem 5

The Jacobi and Gauss-Seidel algorithms applied to linear system $Ax = b$ for $x, b \in \mathbb{R}^n$, A an $n \times n$ matrix, read

$$\begin{cases} x^{(0)} = \mathbf{0} \\ x^{(k)} = B_J x^{(k-1)} + c \end{cases} \qquad \begin{cases} x^{(0)} = \mathbf{0} \\ x^{(k)} = B_{GS} x^{(k-1)} + d \end{cases}$$

for some $c, d \in \mathbb{R}^n$.

a)Suppose that

$$A = \begin{bmatrix} 2 & -1 & 1 \\ 2 & 2 & 2 \\ -1 & -1 & 2 \end{bmatrix}, \qquad b = \begin{bmatrix} -1 \\ 4 \\ -5 \end{bmatrix}$$

i)Find matrices B_J and B_{GS}.

By definition of the Jacobi and Gauss-Seidel methods, we have that

$$B_J = (I - D^{-1}A), \qquad B_{GS} = (I - L^{-1}A).$$

Where $D \in \mathbb{R}^{n \times n}$ is the diagonal component of A, and $L \in \mathbb{R}^{n \times n}$ is the lower triangular component of A, including the main diagonal. Then we compute B_J and B_{GS} directly.

$$\begin{aligned}
B_J &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 2 & -1 & 1 \\ 2 & 2 & 2 \\ -1 & -1 & 2 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 & -\frac{1}{2} & \frac{1}{2} \\ 1 & 1 & 1 \\ -\frac{1}{2} & -\frac{1}{2} & 1 \end{bmatrix} \\
&= \begin{bmatrix} 0 & \frac{1}{2} & -\frac{1}{2} \\ -1 & 0 & -1 \\ \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix} \\
L^{-1} &= \begin{bmatrix} 2 & 0 & 0 \\ 2 & 2 & 0 \\ -1 & -1 & 2 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ -\frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{4} & \frac{1}{2} \end{bmatrix} \\
B_{GS} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ -\frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{4} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 2 & -1 & 1 \\ 2 & 2 & 2 \\ -1 & -1 & 2 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 & -\frac{1}{2} & \frac{1}{2} \\ 0 & \frac{3}{2} & \frac{1}{2} \\ 0 & 0 & \frac{3}{2} \end{bmatrix} \\
&= \begin{bmatrix} 0 & \frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & -\frac{1}{2} \end{bmatrix}
\end{aligned}$$

ii) Show that $\rho\left(B_J\right) = \frac{\sqrt{5}}{2}$ and $\rho\left(B_{GS}\right) = \frac{1}{2}$.

Since for $A \in \mathbb{R}^{n\times n}$, $\rho(A) = \underset{1\le j\le n}{max}|\lambda_j|$, we first seek the eigenvalues of B_J and B_{GS} in order to compute the spectral radius.

$$
\begin{aligned}
C_{B_J}(\lambda) &= \begin{vmatrix} -\lambda & \frac{1}{2} & -\frac{1}{2} \\ -1 & -\lambda & -1 \\ \frac{1}{2} & \frac{1}{2} & -\lambda \end{vmatrix} = -\lambda \begin{vmatrix} -\lambda & -1 \\ \frac{1}{2} & -\lambda \end{vmatrix} - \frac{1}{2}\begin{vmatrix} -1 & -1 \\ \frac{1}{2} & -\lambda \end{vmatrix} - \frac{1}{2}\begin{vmatrix} -1 & -\lambda \\ \frac{1}{2} & \frac{1}{2} \end{vmatrix} \\
&= -\lambda\left(\lambda^2 + \frac{1}{2}\right) - \frac{1}{2}\left(\lambda + \frac{1}{2}\right) - \frac{1}{2}\left(\frac{\lambda}{2} - \frac{1}{2}\right) \\
&= -\lambda^3 - \frac{\lambda}{2} - \frac{\lambda}{2} - \frac{1}{4} - \frac{\lambda}{4} + \frac{1}{4} \\
&= -\lambda^3 - \frac{5}{4}\lambda \\
&= -\lambda\left(\lambda^2 + \frac{5}{4}\right) \Rightarrow \lambda_1 = 0, \qquad \lambda_{2,3} = \frac{\pm\sqrt{-5}}{2} = \pm\frac{\sqrt{5}}{2}i \\
&\Rightarrow |\lambda_1| < |\lambda_2| = |\lambda_3| = \sqrt{\frac{5}{4}} = \frac{\sqrt{5}}{2} = \rho(B_J)
\end{aligned}
$$

$$
\begin{aligned}
C_{B_{GS}}(\lambda) &= \begin{vmatrix} -\lambda & \frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{1}{2} - \lambda & -\frac{1}{2} \\ 0 & 0 & -\frac{1}{2} - \lambda \end{vmatrix} = -\lambda \begin{vmatrix} -\frac{1}{2} - \lambda & -\frac{1}{2} \\ 0 & -\frac{1}{2} - \lambda \end{vmatrix} = -\lambda\left(-\frac{1}{2} - \lambda\right)\left(-\frac{1}{2} - \lambda\right) = -\lambda\left(\frac{1}{2} + \lambda\right)\left(\frac{1}{2} + \lambda\right) \\
&\Rightarrow \lambda_1 = 0, \qquad \lambda_2 = \lambda_3 = -\frac{1}{2} \\
&\Rightarrow |\lambda_1| < |\lambda_2| = |\lambda_3| = \frac{1}{2} = \rho(B_{GS})
\end{aligned}
$$

iii) Apply Jacobi and Gauss-Seidel to the system with $x^{(0)} = \mathbf{0}$ and 25 iterations. Comment on the findings. The exact solution is $x = [1\ 2\ -1]^T$.

Using the Matlab functions designed for problem 4, we attain the following results for the system $Ax = b$:

$$
myJacobi()\text{: } x^{(25)} = \begin{bmatrix} -20.8279 \\ 2.0000 \\ -22.8279 \end{bmatrix}, \qquad myGaussSeidel()\text{: } x^{(25)} = \begin{bmatrix} 1.0000 \\ 2.0000 \\ -1.0000 \end{bmatrix}
$$
$$
myJacobi()\text{: } \|x - x^{(25)}\| = 30.8693, \qquad myGaussSeidel()\text{: } \|x - x^{(25)}\| = 0
$$

The Jacobi algorithm fails to converge for the system $Ax = b$ within 25 iterations while the Gauss-Seidel algorithm converges handily. Increasing the number of iterations does not improve the outcome:

$$
myJacobi()\text{: } x^{(1000)} = \begin{bmatrix} -1.7106 \times 10^{48} \\ -6.8425 \times 10^{48} \\ 1.7106 \times 10^{48} \end{bmatrix}
$$

In the text by Epperson (p. 463) we have the following theorem [1]:

Theorem. Let $A \in \mathbb{R}^{n\times n}$. Let $T = M^{-1}N$, where $A = M - N$. Then

$$
(*) \qquad x^{(k+1)} = M^{-1}Nx^{(k)} + M^{-1}b
$$

converges for all initial guesses $x^{(0)}$ if and only if $\rho(T) < 1$.

The equation $(*)$, has the form of the Jacobi and Gauss-Seidel algorithms when $M^{-1}N = B_J$ and $M^{-1}N = B_{GS}$ respectively. However, we have that $\rho(B_J) = \frac{\sqrt{5}}{2} > 1$, so the Jacobi algorithm is not guaranteed to converge for all choices of $x^{(0)}$. On the other hand, $\rho(B_{GS}) = \frac{1}{2} < 1$, so the Gauss-Seidel algorithm is guaranteed to converge for all choices of $x^{(0)}$ under these conditions. This result explains the observed behavior of both algorithms for the current system $Ax = b$.

b) Suppose that

$$A = \begin{bmatrix} 1 & 2 & -2 \\ 1 & 1 & 1 \\ 2 & 2 & 1 \end{bmatrix}, \qquad b = \begin{bmatrix} 7 \\ 2 \\ 5 \end{bmatrix}$$

i)Find matrices B_J and B_{GS}.

By definition of the Jacobi and Gauss-Seidel methods, we have that

$$B_J = (I - D^{-1}A), \qquad B_{GS} = (I - L^{-1}A).$$

Where $D \in \mathbb{R}^{n\times n}$ is the diagonal component of A, and $L \in \mathbb{R}^{n\times n}$ is the lower triangular component of A, including the main diagonal. Then we compute B_J and B_{GS} directly.

$$\begin{aligned} B_J &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & -2 \\ 1 & 1 & 1 \\ 2 & 2 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 2 & -2 \\ 1 & 1 & 1 \\ 2 & 2 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & -2 & 2 \\ -1 & 0 & -1 \\ -2 & -2 & 0 \end{bmatrix} \\ L^{-1} &= \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & 2 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix} \\ B_{GS} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & -2 \\ 1 & 1 & 1 \\ 2 & 2 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 2 & -2 \\ 0 & -1 & 3 \\ 0 & 0 & -1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & -2 & 2 \\ 0 & 2 & -3 \\ 0 & 0 & 2 \end{bmatrix} \end{aligned}$$

ii) Show that $\rho\left(B_J\right) = 0$ and $\rho\left(B_{GS}\right) = 2$.

Since for $A \in \mathbb{R}^{n\times n}$, $\rho(A) = \underset{1\leq j\leq n}{max} |\lambda_j|$, we first seek the eigenvalues of B_J and B_{GS} in order to compute the spectral radius.

$$\begin{aligned} C_{B_J}(\lambda) &= \begin{vmatrix} -\lambda & -2 & 2 \\ -1 & -\lambda & -1 \\ -2 & -2 & -\lambda \end{vmatrix} = -\lambda \begin{vmatrix} -\lambda & -1 \\ -2 & -\lambda \end{vmatrix} + 2 \begin{vmatrix} -1 & -1 \\ -2 & -\lambda \end{vmatrix} + 2 \begin{vmatrix} -1 & -\lambda \\ -2 & -2 \end{vmatrix} \\ &= -\lambda\left(\lambda^2 - 2\right) + 2\left(\lambda - 2\right) + 2\left(2 - 2\lambda\right) \\ &= -\lambda^3 + 2\lambda + 2\lambda - 4 + 4 - 4\lambda \end{aligned}$$

$$= -\lambda^3 \Rightarrow \lambda_1 = \lambda_2 = \lambda_3 = 0$$
$$\Rightarrow \rho(B_J) = |\lambda_1| = |\lambda_2| = |\lambda_3| = 0$$
$$C_{B_{GS}}(\lambda) = \begin{vmatrix} -\lambda & -2 & 2 \\ 0 & 2-\lambda & -3 \\ 0 & 0 & 2-\lambda \end{vmatrix} = -\lambda \begin{vmatrix} 2-\lambda & -3 \\ 0 & 2-\lambda \end{vmatrix} = -\lambda\,(2-\lambda)\,(2-\lambda)$$
$$\Rightarrow \lambda_1 = 0, \qquad \lambda_2 = \lambda_3 = 2$$
$$\Rightarrow |\lambda_1| < |\lambda_2| = |\lambda_3| = 2 = \rho(B_{GS})$$

iii) Apply Jacobi and Gauss-Seidel to the system with $x^{(0)} = \mathbf{0}$ and 25 iterations. Comment on the findings. The exact solution is $x = [1\ 2\ {-1}]^T$.

Using the Matlab functions designed for problem 4, we attain the following results for the system $Ax = b$:

$$myJacobi()\colon\ x^{(25)} = \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix}, \qquad myGaussSeidel()\colon\ x^{(25)} = \begin{bmatrix} 1.3086 \times 10^9 \\ -1.3254 \times 10^9 \\ 0.0336 \times 10^9 \end{bmatrix}$$
$$myJacobi()\colon\ \|x - x^{(25)}\| = 0, \qquad myGaussSeidel()\colon\ \|x - x^{(25)}\| = 1.8629 \times 10^9$$

This time, the Gauss-Seidel algorithm fails to converge for the system $Ax = b$ within 25 iterations while the Jacobi algorithm converges handily. Increasing the number of iterations does not improve the outcome for Gauss-Seidel:

$$myGaussSeidel()\colon\ x^{(1000)} = \begin{bmatrix} 1.6089 \times 10^{304} \\ -1.6094 \times 10^{304} \\ 0.0011 \times 10^{304} \end{bmatrix}$$

This result is again attributable to the respective spectral radii of B_J and B_{GS}, where now $B_J = 0 < 1$ and $B_{GS} = 2 > 1$. By the same theorem as was used to argue on the convergence of these algorithms in 5.a.iii, we would expect the Jacobi algorithm to converge for all choices of $x^{(0)}$ while the Gauss-Seidel algorithm is not guaranteed to converge for all $x^{(0)}$, but most certainly does not converge for $x^{(0)} = \mathbf{0}$.

References

[1] James F. Epperson. *An introduction to numerical methods and analysis, second edition.* John Wiley & Sons, Hoboken, New Jersey, 2013.