1. Consider the matrix, right side vector, and two approximate solutions,

\[
A = \begin{bmatrix}
1.2969 & 0.8648 \\
0.2161 & 0.1441
\end{bmatrix}, \quad b = \begin{bmatrix}
0.8642 \\
0.1440
\end{bmatrix}, \quad x_1 = \begin{bmatrix}
0 \\
1
\end{bmatrix}, \quad x_2 = \begin{bmatrix}
0.9911 \\
-0.4870
\end{bmatrix}.
\]

(a) Show that \( x = [2, -2]^T \) is the exact solution of \( Ax = b \).

(b) Compute the error and residual vectors for \( x_1 \) and \( x_2 \).

(c) Use MATLAB to find \( ||A||\infty, ||A^{-1}||\infty \), and \( \kappa_\infty(A) \).

(d) In class we proved

\[
\frac{||e||}{||x||} \leq \kappa(A) \frac{||r||}{||b||}
\]

where \( \kappa(A) \) is the condition number of \( A \), \( e \) is the error and \( r \) the residual. Verify this result for the two approximate solutions \( x_1 \) and \( x_2 \) using the \( \infty \) – norm.

ANS: For (a), a direct calculation gives

\[
Ax = \begin{bmatrix}
1.2969 & 0.8648 \\
0.2161 & 0.1441
\end{bmatrix} \begin{bmatrix}
2 \\
-2
\end{bmatrix} = \begin{bmatrix}
2 \times 1.2969 - 2 \times 0.8648 \\
2 \times 0.2161 - 2 \times 0.1441
\end{bmatrix} = \begin{bmatrix}
0.8642 \\
0.1440
\end{bmatrix} = b
\]

Using MATLAB, for (b) we have

\[
\begin{align*}
A &= [1.2969 \ 0.8648; \\
   0.2161 \ 0.1441]; \\
b &= [0.8642 \ 0.1440]'; \\
x1 &= [0 \ 1]'; \\
x2 &= [0.9911 \ -0.4870]'; \\
x &= [2 \ -2]'; \\
e1 &= x-x1, \ r1 = b-A*x1
\end{align*}
\]

\[
e1 =
\begin{bmatrix}
2 \\
-3
\end{bmatrix}
\]

\[
r1 =
\begin{bmatrix}
-0.000600000000000045 \\
-0.000100000000000017
\end{bmatrix}
\]

\[
\begin{align*}
e2 &= x-x2, \ r2 = b-A*x2
\end{align*}
\]

\[
e2 =
\begin{bmatrix}
1.0089 \\
-1.513
\end{bmatrix}
\]
\textbf{For (c), again using MATLAB}

\begin{verbatim}
>> A_inf = norm(A,'inf'), Ainv_inf = norm(inv(A),'inf'), A_condinf = cond(A,'inf')
A_inf =
    2.1617

Ainv_inf =
    151300000.022015

A_condinf =
    327065210.047589
\end{verbatim}

So while the norm of $A$ is small, the norm of $A^{-1}$ is quite large, resulting in $\|A\|\|A^{-1}\| = \kappa(A) \approx 3.27 \times 10^8$.

For (d), we proved in class that for any norm and corresponding induced matrix norm

$$
\frac{\|c\|}{\|x\|} \leq \kappa(A) \frac{\|r\|}{\|b\|} = \|A\|\|A^{-1}\| \frac{\|r\|}{\|b\|}.
$$

To see that this holds here in the infinity norm,

\begin{verbatim}
>> norm(e1,'inf')/norm(x,'inf'), A_condinf*norm(r1,'inf')/norm(b,'inf')
ans =
    1.5

ans =
    227076.054507183

>> norm(e2,'inf')/norm(x,'inf'), A_condinf*norm(r2,'inf')/norm(b,'inf')
ans =
    0.7565
\end{verbatim}
ans =

3.78460096948699

which indeed it does. Note that the relative residual for \( x_1 \) is much larger than that for \( x_2 \), while the error in \( x_2 \) is larger.
2. Matrix norms

(a) Consider the matrix,

\[ A = \begin{bmatrix} 2 & -3 & 1 \\ -4 & 1 & 2 \\ 5 & 0 & 1 \end{bmatrix}. \]

Compute \( \|A\|_\infty \) and find a vector \( x \) such that \( \|A\|_\infty = \|Ax\|_\infty / \|x\|_\infty \).

(b) Find a non-zero \( 2 \times 2 \) matrix \( A \) such that \( \rho(A) = 0 \). This shows that the spectral radius \( \rho(A) \) does not define a matrix norm.

**ANS:**

(a) Since \( \|A\|_\infty \) is the maximum absolute row sum of \( A \), it is easy to see that \( \|A\|_\infty = 7 \). Letting \( x = [-1 \ 1 \ 1]^T \), we see that \( \|x\|_\infty = 1 \) and \( \|Ax\|_\infty / \|x\|_\infty = \|[-4 \ 7 \ -4]^T\|_\infty / 1 = 7 \).

(b) Let \( A = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \). Then it is easily seen that \( \|A\|_\infty = 1 \) but \( \rho(A) = 0 \). This follows since the infinity norm of \( A \) is just the maximum absolute column sum, and for a lower triangular matrix the eigenvalues lie on the diagonal. Since they are both 0, the spectral radius is 0. This show that \( \rho(A) \) is not a matrix norm, for we have found a non-zero matrix for which \( \rho(A) = 0 \). This violates one of the properties that any matrix norm must satisfy, namely, \( \|A\| = 0 \iff A = 0 \).
function x = trisolve(a,b,c,f)

% A = tridiag(b,a,c)

n = length(a); % determine system size
x = zeros(n,1); % allocate x

for i = 1:n-1 % forward elimination sweep GE
    m = b(i+1)/a(i); % compute multiplier
    a(i+1) = a(i+1)-m*c(i); % a (but not c) changes for row operation
    f(i+1) = f(i+1)-m*f(i); % row op applied to RHS f
end

x(n) = f(n)/a(n); % back substitution
for i = n-1:-1:1
    x(i) = (f(i)-c(i+1)*x(i+1))/a(i);
end

and here is the output of the 5x5 test system:

>> a = 2*ones(5,1); b = -ones(5,1); c=b; f=[1 0 0 0 1]’;
>> x = trisolve(a,b,c,f)

x =

1.0000
1.0000
1.0000
1.0000
1.0000
4. The Hilbert matrices are given by \( H_{ij} = \frac{1}{i + j - 1} \). The \( n \times n \) Hilbert matrix \( H_n \) is easily produced in MATLAB using \( \text{hilb}(n) \). Assume the true solution of \( H_nx = b \) for a given \( n \) is \( x = [1, \ldots, 1]^T \). Hence the righthand side \( b \) is simply the row sums of \( H_n \), and \( b \) is easily computed in MATLAB using \( b = \text{sum(hilb(n))}' \). Use MATLAB’s \( x = A\backslash b \) to solve the system \( H_nx = b \) for \( n = 5, 10, 15, 20 \). Qualitatively, how does the computed solution compare to the exact as \( n \) gets larger? Discuss what you think is happening here.

\[ \text{ANS: Here is the output:} \]

\[
\begin{array}{cccccc}
 n & \text{rel.err} & \text{rel.resid} & \text{cond} \\
\hline
 5.0000e+00 & 5.5467e-13 & 1.9449e-16 & 9.4366e+05 \\
 1.0000e+01 & 1.3982e-04 & 1.5162e-16 & 3.5353e+13 \\
 1.5000e+01 & 2.4345e+01 & 4.0150e-16 & 1.0975e+18 \\
 2.0000e+01 & 4.7309e+01 & 3.7031e-16 & 9.2117e+18 \\
\end{array}
\]

So we see that while for each \( n \) the relative residual is on the order of machine precision, the relative error is growing. This is directly attributable to the growth in the condition number of the Hilbert matrices. In fact, I’ve included the warnings that MATLAB displayed for \( n = 15, 20 \) to note that it is having trouble accurately computing \( H_n^{-1} \).