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

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

Using MATLAB, for (b) we have

```matlab
>> 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
```

\( e_1 = \begin{bmatrix} 2 \\ -3 \end{bmatrix} \)

\( r_1 = \begin{bmatrix} -0.000600000000000045 \\ -0.0001000000000000017 \end{bmatrix} \)

```matlab
>> e2 = x-x2, r2 = b-A*x2
```

\( e_2 = \begin{bmatrix} 1.0089 \\ -1.513 \end{bmatrix} \)
\[
r_2 = \\
\begin{align*}
1.000000001612699e-08 \\
-1.0000000022492e-08
\end{align*}
\]

For (c), again using MATLAB

\[
\text{>> } A_{\text{inf}} = \text{norm}(A,'\text{inf'}), \ A_{\text{inv inf}} = \text{norm(inv(A),'inf'}), \ A_{\text{cond inf}} = \text{cond(A,'inf')}
\]

\[
\begin{align*}
A_{\text{inf}} &= 2.1617 \\
A_{\text{inv inf}} &= 151300000.022015 \\
A_{\text{cond inf}} &= 327065210.047589
\end{align*}
\]

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

\[
\|e\| \leq \kappa(A) \frac{\|r\|}{\|x\|} = \|A\|\|A^{-1}\| \frac{\|r\|}{\|b\|}
\]

To see that this holds here in the infinity norm,

\[
\text{>> } \text{norm(e1,'inf')/norm(x,'inf'), } A_{\text{cond inf}} \text{*norm(r1,'inf')/norm(b,'inf')}
\]

\[
\text{ans = 1.5}
\]

\[
\text{ans = 227076.054507183}
\]

\[
\text{>> } \text{norm(e2,'inf')/norm(x,'inf'), } A_{\text{cond inf}} \text{*norm(r2,'inf')/norm(b,'inf')}
\]

\[
\text{ans = 0.7565}
\]
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 \( \|Ax\|_\infty /\|x\|_\infty = \|[\begin{bmatrix} -4 & 7 \\ -4 & 4 \end{bmatrix}]\|_\infty /1 = 7. \)

(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, \( \|A\| = 0 \iff A = 0 \).
3. Write a MATLAB function M-file \texttt{trisolve.m} to solve the linear system \(Ax = f\) where

\[
A = \begin{pmatrix}
  a_1 & c_1 \\
b_2 & a_2 & c_2 \\
  & \ddots & \ddots \\
  & & \ddots & c_{n-1} \\
  & & & b_n
\end{pmatrix}
\]

is a tridiagonal \(n \times n\) matrix. Assume that no partial pivoting is required. The inputs are the \(n\)-vectors \(a, b, c\) and \(f\) and returns the solution \(x\). Its first line should read:

\[
\text{function } x = \text{trisolve}(a,b,c,f)
\]

Test your code with the \(5 \times 5\) system with \(a_i = 2, b_i = -1, c_i = -1\), and RHS \(f = [1, 0, 0, 0, 1]^T\). The exact solution is \(x = [1, 1, 1, 1, 1]^T\). Use MATLAB’s \texttt{diary} command to save your MATLAB session output showing that your code works properly. Include a copy of both codes.

\textbf{ANS:} Here is the MATLAB code

\[
\text{function } x = \text{trisolve}(a,b,c,f)
\]

\[
\% \quad A = \text{tridiag}(b,a,c)
\]

\[
\text{n = length(a) \quad \% determine system size}
\]

\[
x = \text{zeros(n,1)} \quad \% \text{allocate } x
\]

\[
\text{for } i = 1:n-1 \quad \% \text{forward elimination sweep GE}
\]

\[
\text{m = b(i+1)/a(i); \quad \% compute multiplier}
\]

\[
a(i+1) = a(i+1)-m*c(i); \quad \% \text{a (but not } c\text{) changes for row operation}
\]

\[
f(i+1) = f(i+1)-m*f(i); \quad \% \text{row op applied to RHS } f
\]

\[
\text{end}
\]

\[
x(n) = f(n)/a(n); \quad \% \text{back substitution}
\]

\[
\text{for } i = n-1:-1:1
\]

\[
x(i) = (f(i)-c(i+1)*x(i+1))/a(i);
\]

\[
\text{end}
\]

and here is the output of the \(5 \times 5\) test system:

\[
>> a = 2*\text{ones}(5,1); \quad b = -\text{ones}(5,1); \quad c=b; \quad f=[1 \ 0 \ 0 \ 0 \ 1]^T;
\]

\[
>> x = \text{trisolve}(a,b,c,f)
\]

\[
x =
\]

\[
1.0000 \\
1.0000 \\
1.0000 \\
1.0000 \\
1.0000
\]

and here is the output of the \(5 \times 5\) test system:
4. Consider the 2-point BVP
\[
\begin{align*}
-u'' &= -(x^2 + 3x)e^x \\
u(0) &= u(1) = 0
\end{align*}
\]
(a) Show \(u(x) = (x^2 - x)e^x\) is the exact solution.
(b) Write a MATLAB function M-file to solve the problem using the 2nd order centered FD scheme we discussed in class, \(-D^2v_i = f_i\), that utilizes your m-file trisolve.m from problem 3 above. Assume a mesh size \(h = 1/n\) where \(n = 2^p\) for \(p\) a positive integer. For \(p = 1:12\) present a table with the following data - column 1: \(h\); column 2: \(\|u_h - v_h\|_\infty\); column 3: \(\|u_h - v_h\|_\infty/h^2\); where \(h = 1/n\). What does the trend in the third column indicate? Include a copy of your code.

ANS: (a) is just straightforward substitution.

Here is the output and code for (b). Note that the third column is approaching a constant, indicating second order convergence.

<table>
<thead>
<tr>
<th>(h)</th>
<th>inf_error</th>
<th>inf_error/h^2</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.0000e-01</td>
<td>5.1523e-02</td>
<td>2.0609e-01</td>
</tr>
<tr>
<td>2.5000e-01</td>
<td>1.3185e-02</td>
<td>2.1096e-01</td>
</tr>
<tr>
<td>1.2500e-01</td>
<td>3.3308e-03</td>
<td>2.1317e-01</td>
</tr>
<tr>
<td>6.2500e-02</td>
<td>8.4567e-04</td>
<td>2.1649e-01</td>
</tr>
<tr>
<td>3.1250e-02</td>
<td>2.1150e-04</td>
<td>2.1657e-01</td>
</tr>
<tr>
<td>1.5625e-02</td>
<td>5.2879e-05</td>
<td>2.1659e-01</td>
</tr>
<tr>
<td>7.8125e-03</td>
<td>1.3221e-05</td>
<td>2.1662e-01</td>
</tr>
<tr>
<td>3.9062e-03</td>
<td>3.3054e-06</td>
<td>2.1662e-01</td>
</tr>
<tr>
<td>1.9531e-03</td>
<td>8.2635e-07</td>
<td>2.1662e-01</td>
</tr>
<tr>
<td>9.7656e-04</td>
<td>2.0659e-07</td>
<td>2.1662e-01</td>
</tr>
<tr>
<td>4.8828e-04</td>
<td>5.1647e-08</td>
<td>2.1662e-01</td>
</tr>
<tr>
<td>2.4414e-04</td>
<td>1.2910e-08</td>
<td>2.1662e-01</td>
</tr>
</tbody>
</table>

```matlab
p = (1:12)';
n = 2.^p;
dispvars = zeros(length(p),3);

for i = 1:length(p)
    N = n(i); h = 1/N; x = h*(0:N)';
    u_h = (x.^2-x).*exp(x); % true solution u(0)=u(1)=0
    f_h = -(x.^2+3*x).*exp(x); % -u'' = f
    a = 2*ones(N-1,1)/(h^2); % create a, b & c
    b = -ones(N-1,1)/(h^2);
    c = -ones(N-1,1)/(h^2);
    ftil = f_h(2:N); % rhs is f evaluated at n-1 interior pts
...
v_h = trisolve(a,b,c,ftil); % solve
v_h = [0; v_h; 0]; % set BCs u(0)=u(1)=0

dispvars(i,1) = h;
dispvars(i,2) = max(abs(v_h-u_h));
dispvars(i,3) = dispvars(i,2)/h^2;
end

format short e
disp(' ')
disp(' h  inf_error  inf_error/h^2')
disp('-----------------------------------------')
disp(dispvars)