THE CHINESE UNIVERSITY OF HONG KONG

Department of Mathematics

2024-25 Term 2 MATH2221A Mathematics Laboratory II

Test 1 Suggested Solutions

- Full Mark: 60

1. (a) (6 marks) Consider the following system of linear equations:

$$\begin{cases} x + 2y + 3z &= 1 \\ x + 3y + 5z &= 2 \\ 3x + y + 4z &= -7 \end{cases}$$

Write down the MATLAB commands for solving the above system of linear equations and the answer obtained in the box below.

Solution:

```
A = [1,2,3;1,3,5;3,1,4];
b = [1;2;-7];
u = A\b;
```

Answer: $(x, y, z) = (-2, 3, -1)$

(b) (6 marks) Consider the following matrix:

$$B = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}.$$

Write down the MATLAB commands for constructing the matrix B <u>without directly inputting the entries one by one</u> in the box below.

Solution:

```
B = [ones(2,3), zeros(2,2); 2*eye(5)];
```

2. Let

$$s = \sqrt{1}\sin\left(\frac{\pi}{1}\right) + \sqrt{2}\sin\left(\frac{\pi}{2}\right) + \cdots + \sqrt{100}\sin\left(\frac{\pi}{100}\right).$$

(a) (6 marks) Write a MATLAB script `q2a.m` to find the value of s using `for` loop. Include the code file `q2a.m` in your submission.

Solution:

```
s = 0;
for i = 1:100
    s = s + sqrt(i)*sin(pi/i);
end
```

Answer: $s = 53.6306$

(b) (6 marks) Write a MATLAB script `q2b.m` to find the value of s <u>without using any loops</u>. Include the code file `q2b.m` in your submission.

Solution:

```
v = 1:100;
s = sum(sqrt(v).*sin(pi./v));
```

Answer: $s = 53.6306$

3. (10 marks) Consider the parametric surface $\mathcal{S}$ given by

$$\begin{cases} X(u,v) &= \cos(2u) + v\cos(u)\cos(2u), \\ Y(u,v) &= \sin(2u) + v\cos(u)\sin(2u), \\ Z(u,v) &= v\sin(u), \end{cases}$$

where $0 \le u \le \pi$ and $-0.2 \le v \le 0.2$.

Here, we consider 50 equally spaced points between 0 and π for u, and 7 equally spaced points between -0.2 and 0.2 for v.

Write a MATLAB script `q3.m` to do the following:

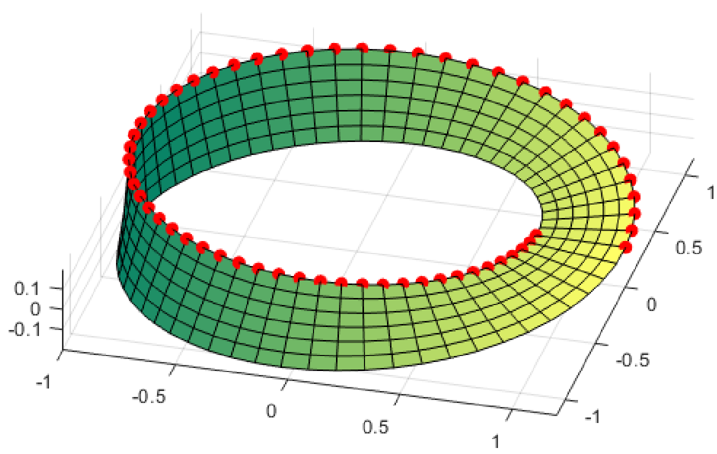
- Create a surface plot of $\mathcal{S}$ with the face color set based on the value of X, equal axis scales, the `summer` color scheme, and the view angle $(15, 30)$.
- In the same figure, plot all points (X, Y, Z) for which $v = 0.2$ with red circle markers and set the face color of the markers as red.

Include the code file `q3.m` in your submission.

Solution:

```
u = linspace(0,pi,50);
v = linspace(-0.2,0.2,7);
[U,V] = meshgrid(u,v);

X = cos(2*U) + V.*cos(U).*cos(2*U);
Y = sin(2*U) + V.*cos(U).*sin(2*U);
Z = V.*sin(U);

figure;
surf(X,Y,Z,X);
axis equal
colormap summer
view(15,30)
hold on;
% note that the last row of the 2-D array V corresponds to ...
    the points with v = 0.2
plot3(X(end,:),Y(end,:),Z(end,:),'ro','MarkerFaceColor','r');
```

The figure is as follows (the surface is a Möbius strip):



4. Let n be a given positive integer. Define a sequence $\{a_k\}$ with $a_0 = n$ and

$$a_{k+1} = \begin{cases} 3a_k + 1 & \text{if } a_k \text{ is odd,} \\ \dfrac{a_k}{2} & \text{if } a_k \text{ is even,} \end{cases}$$

for all integer $k \geq 0$.

(a) (8 marks) Write a MATLAB function `K = mysequence(n)` that takes a positive integer n as input (which serves as the value of a_0 for defining the sequence $\{a_k\}$ above) and outputs the smallest integer K such that $a_K = 1$. Include the code file `mysequence.m` in your submission.

Solution:

```
function K = mysequence(n)
a = n;
K = 0; % counter
while a ~= 1
    K = K+1;
    if mod(a,2) == 1 % the odd case
        a = 3*a+1;
    else % the even case
        a = a/2;
    end
end
end
```

(b) (4 marks) Write a MATLAB script `q4b.m` to do the following:

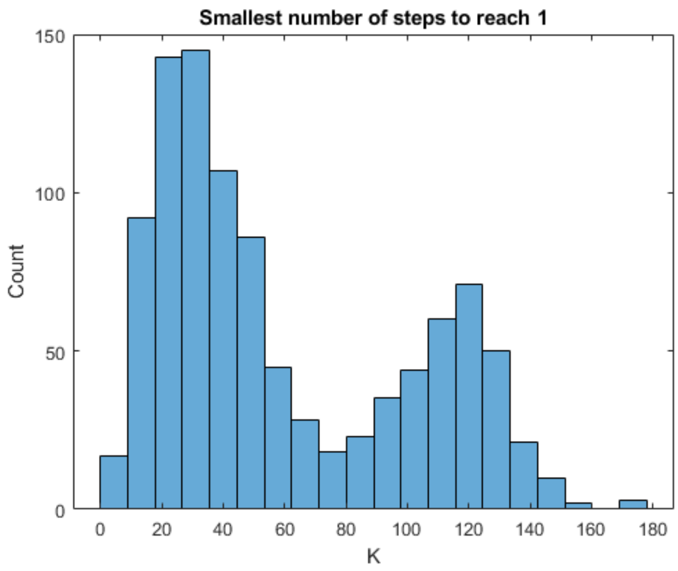
- For every $n = 1, 2, \ldots, 1000$, obtain the corresponding value of K using the `mysequence` function.
- Create a histogram plot of all the 1000 recorded values of K, and set the number of bins as 20.
- Label the x-axis as "K" and the y-axis as "Count", and add the title "Smallest number of steps to reach 1" to the figure.

Include the code file `q4b.m` in your submission.

Solution:

```
K_all = zeros(1000,1);
for n = 1:1000
    K_all(n) = mysequence(n);
end

figure;
histogram(K_all,20);
xlabel('K')
ylabel('Count')
title('Smallest number of steps to reach 1')
```

The figure is as follows:



Remark: Whether the sequence will eventually reach 1 for any input n is also known as the $3n+1$ problem (the Collatz conjecture). The conjecture has been verified computationally for all positive integers up to 2.95×10^{20}, but the general proof remains an open problem.

5. Let $A = (0,0), B = (1,0), C = \left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right)$ be three points in $\mathbb{R}^2$.

   (a) (4 marks) Write a MATLAB script `q5a.m` to do the following:

   - Compute the midpoints M_{AB}, M_{BC}, M_{CA} of the three sides of $\triangle ABC$.

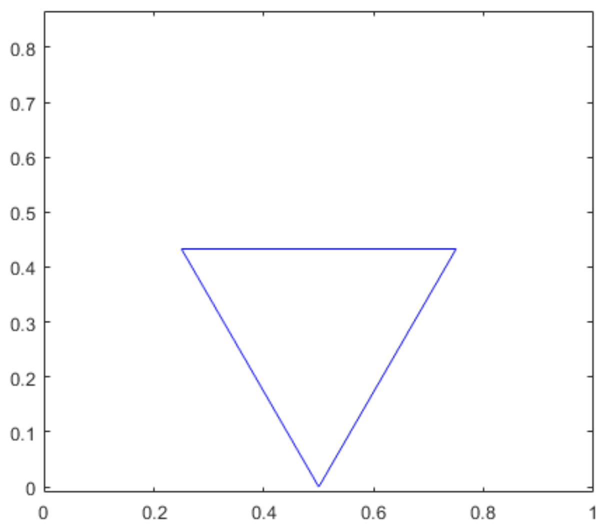
- Create a MATLAB figure and plot the triangle formed by the three midpoints (i.e., $\triangle M_{AB}M_{BC}M_{CA}$) with blue solid lines and equal axis scales. Also, change the axis limits so that the x-axis ranges from 0 to 1 and the y-axis ranges from -0.01 to $\frac{\sqrt{3}}{2}$.

Include the code file `q5a.m` in your submission.

Solution:

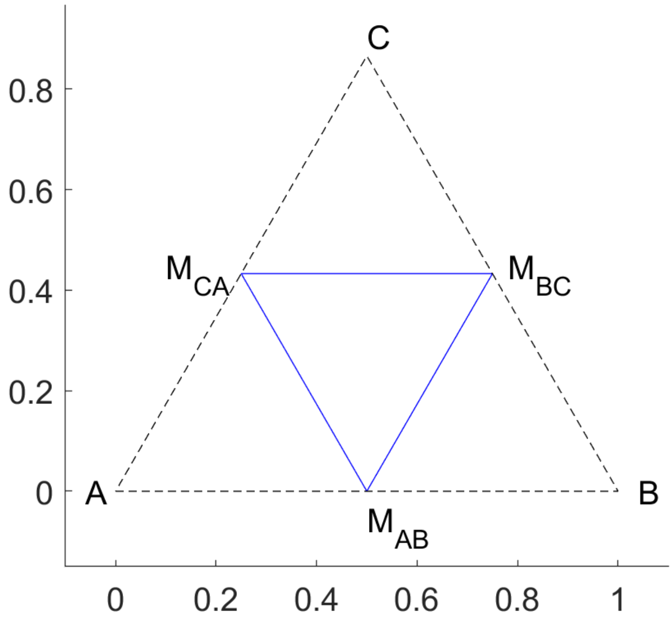
```
a = [0,0];
b = [1,0];
c = [1/2,sqrt(3)/2];

% compute the midpoints
mab = (a+b)/2;
mbc = (b+c)/2;
mca = (c+a)/2;

% plot three edges to form the triangle
figure;
plot([mab(1),mbc(1),mca(1),mab(1)], ...
    [mab(2),mbc(2),mca(2),mab(2)], 'b-');
axis equal;
axis([0 1 -0.01 sqrt(3)/2]);
```

The figure is as follows:

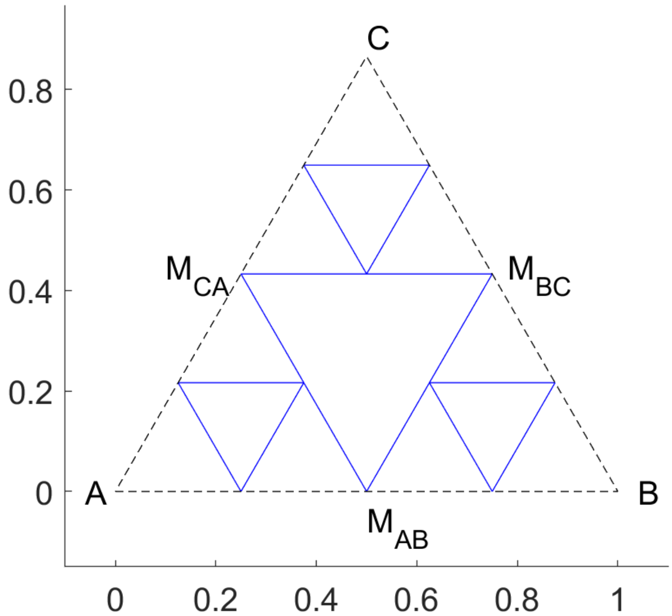


(b) (10 marks) Let L be a given positive integer. Consider the following operations:

- If $L = 1$, we draw the triangle formed by the three midpoints M_{AB}, M_{BC}, M_{CA} with blue solid lines as described in Part (a). An illustration is as follows (note that only the blue solid lines are required; the dashed lines and text labels are for reference only and are not required):

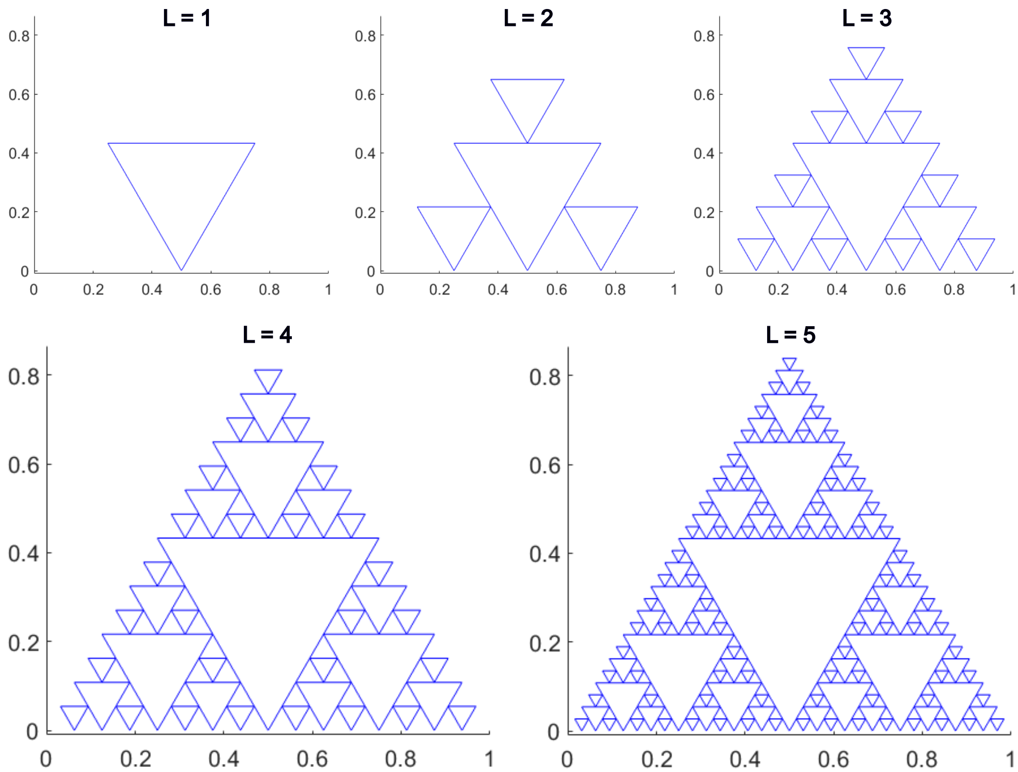


- If $L = 2$, we draw the triangle in the case $L = 1$ together with three smaller triangles with blue solid lines in the same figure, where the three smaller triangles are formed by <u>the midpoints of the sides of $\triangle AM_{AB}M_{CA}$, $\triangle BM_{BC}M_{AB}$ and $\triangle CM_{CA}M_{BC}$</u> respectively. An illustration is as follows (the dashed lines and text labels are not required):



- More generally, for the case L (where $L > 1$ is any given positive integer), we draw all triangles included in the case $(L-1)$ together with 3^{L-1} new smaller triangles with blue solid lines in the same figure in a similar manner. More precisely, every new smaller triangle will satisfy all of the following properties:
  - It is a downward-pointing, equilateral triangle with side length 2^{-L}.
  - Every vertex of it must be the midpoint of an edge formed by the vertices A, B, C and/or some vertices created in the previous operations.

See below for an illustration for $L = 1, 2, 3, 4, 5$:



Write a MATLAB function `TrianglePlot(L)` that takes a positive integer L as input and produces a MATLAB figure of the triangle plot for the case L as described above.

Include your code file `TrianglePlot.m` (and any additional code files if applicable) in your submission. Different test cases will be used for evaluating your code.

Warning: Please DO NOT try any $L \geq 8$ during the test as the computation may take a very long time.

Solution:

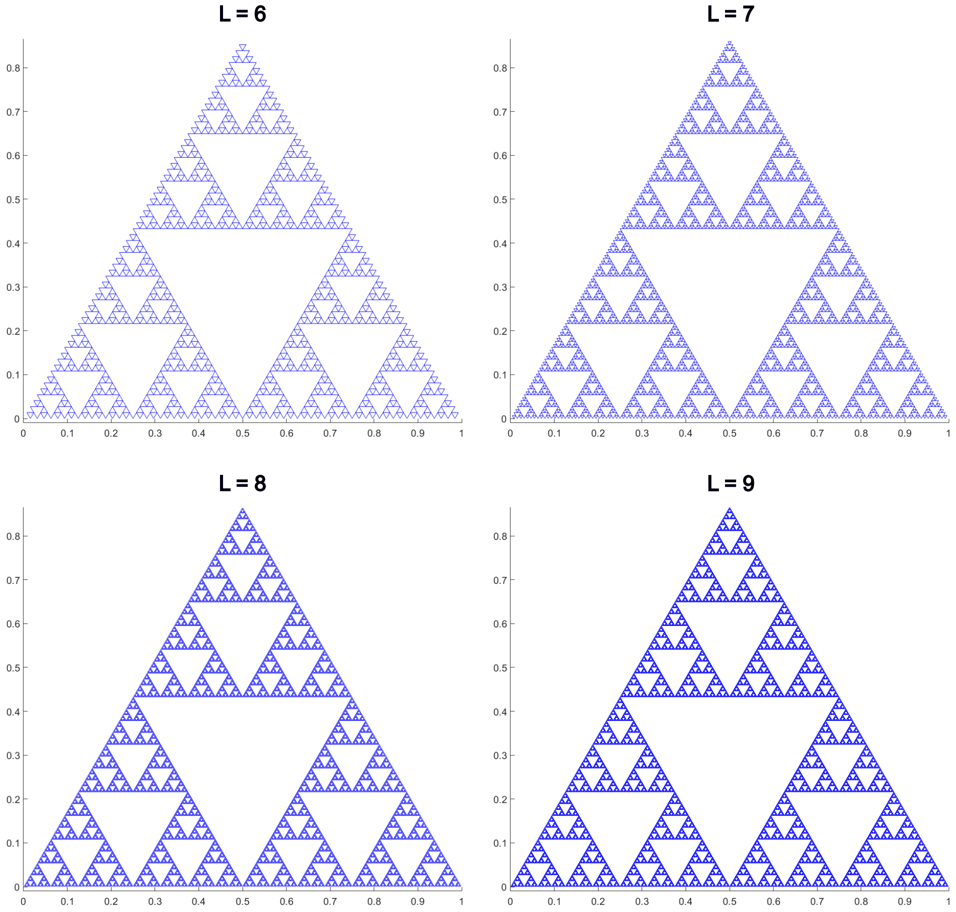
We first create a recursive function `draw_triangle(L,a,b,c)` that takes a positive integer L and three 1×2 vectors a, b, c (representing the coordinates of three vertices) as input. The function will draw a triangle using the midpoints of the three edges formed by a, b, c and then call itself three times with different inputs.

```
function draw_triangle(L,a,b,c)
if L >= 1
    % compute the midpoints
    mab = (a+b)/2;
    mbc = (b+c)/2;
    mca = (c+a)/2;
    % draw the triangle formed by the midpoints
    plot([mab(1),mbc(1),mca(1),mab(1)],...
        [mab(2),mbc(2),mca(2),mab(2)],'b-');

    % proceed to the next level for each of the three corners
    draw_triangle(L-1,a,mab,mca);
    draw_triangle(L-1,mab,b,mbc);
    draw_triangle(L-1,mca,mbc,c);
end
end
```

The required `TrianglePlot` function can then be written as follows:

```
function TrianglePlot(L)
a = [0,0];
b = [1,0];
c = [1/2,sqrt(3)/2];
figure;
hold on;
draw_triangle(L,a,b,c);
axis equal;
axis([0 1 -0.01 sqrt(3)/2])
% Note: For efficiency, it is much better to put all ...
    ``hold on'' and axis adjustments outside the recursion
end
```

Besides the results for the cases $L = 1, 2, 3, 4, 5$ as shown above, we also show the results for $L = 6, 7, 8, 9$ here:



Remarks:

- *If you would like to visualize how the triangles are drawn one by one, you may add* **`pause(0.5)`** *after the line of plotting in* **`draw_triangle.m`***, which temporarily stops the MATLAB execution for 0.5 second.*

- *You may also save the triangle drawing process as a video (see the file* **`triangles.mp4`** *on Blackboard). You will learn more about video input and output in the coming few weeks.*

○ △ □ **End of Test** ○ △ □