THE CHINESE UNIVERSITY OF HONG KONG

Department of Mathematics

2024-25 Term 2 MATH2221A Mathematics Laboratory II

Lab Assignment 5 Suggested Solutions

- Full Mark: 40

1. Consider the following 6 data points (x, y):

x	1.1	3.4	5.6	7.9	8.8	10.2
y	4.5	18.2	34.3	59.2	68.7	85.6

   (a) (6 marks) Write a MATLAB script `q1.m` to create a MATLAB figure with three subplots:

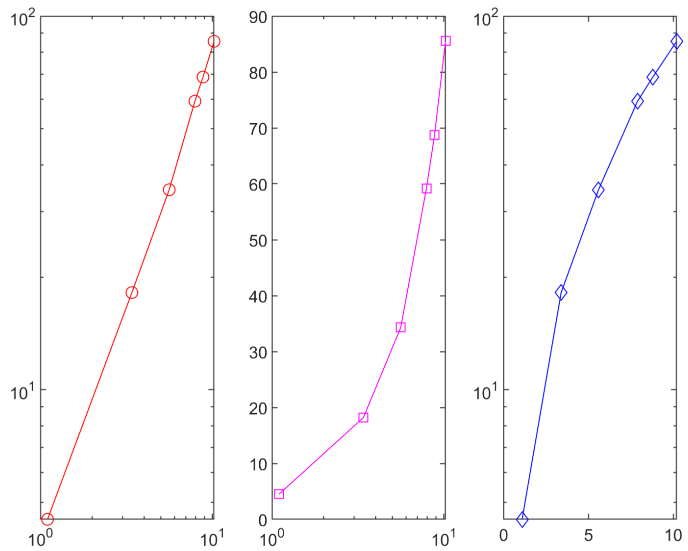
   - Subplot 1: A log-log plot of all data points (x, y) with red circle markers and solid lines.
   - Subplot 2: A semi-log plot of all data points (x, y) using a base-10 logarithmic scale on the x-axis and a linear scale on the y-axis, with magenta square markers and solid lines.
   - Subplot 3: A semi-log plot of all data points (x, y) using a linear scale on the x-axis and a base-10 logarithmic scale on the y-axis, with blue diamond markers and solid lines.

   Include the code file `q1.m` in your submission.

Solution:

```
x = [1.1, 3.4, 5.6, 7.9, 8.8, 10.2];
y = [4.5, 18.2, 34.3, 59.2, 68.7, 85.6];
figure;
subplot(1,3,1);
loglog(x,y,'ro-');
subplot(1,3,2);
semilogx(x,y,'ms-');
subplot(1,3,3);
semilogy(x,y,'bdiamond-');
```

The figure is as follows:



(b) (3 marks) Based on your plots in part (a), which of the following formulas can best describe the relationship between x and y (where a, k are some constants)?

$$y = kx^a, \qquad x = ka^y, \qquad y = ka^x$$

Write down your answer and give a brief justification in the box below.

Solution:

Since the log-log plot shows a linear trend, we consider

$$\log_{10} y = m \log_{10} x + c \Rightarrow y = 10^c x^m.$$

Therefore, we have $\boxed{y = kx^a}$.

2. (5 marks) Let θ be a vector with 2000 equally spaced points between 0 and 12π, and let

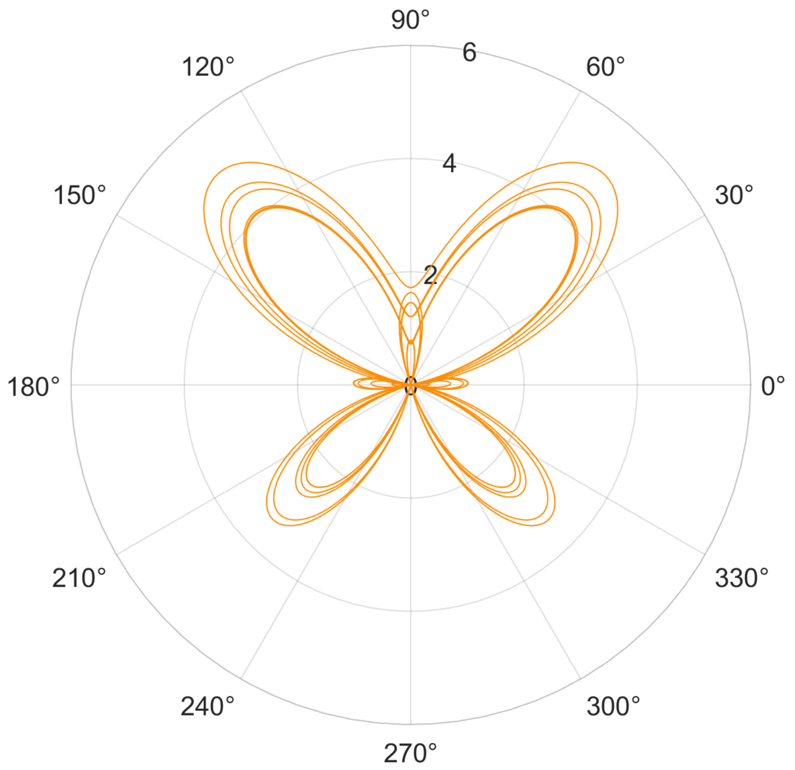
$$r(\theta) = e^{\sin\theta} - 2\cos(4\theta) + \sin^5\left(\frac{2\theta - \pi}{24}\right).$$

Write a MATLAB script `q2.m` to create a plot of $r(\theta)$ in the polar coordinate system using solid lines with RGB color value `[254,141,3]/255`. Include the code file `q2.m` in your submission.

Solution:

```
t = linspace(0,12*pi,2000);
r = exp(sin(t)) - 2*cos(4*t) + sin((2*t-pi)/24).^5;
figure;
polarplot(t,r,'-','Color',[254,141,3]/255);
```

The figure is as follows:



Remark: This curve is also known as the butterfly curve.

3. (a) (4 marks) Let $\mathcal{C}$ be a 3D curve with

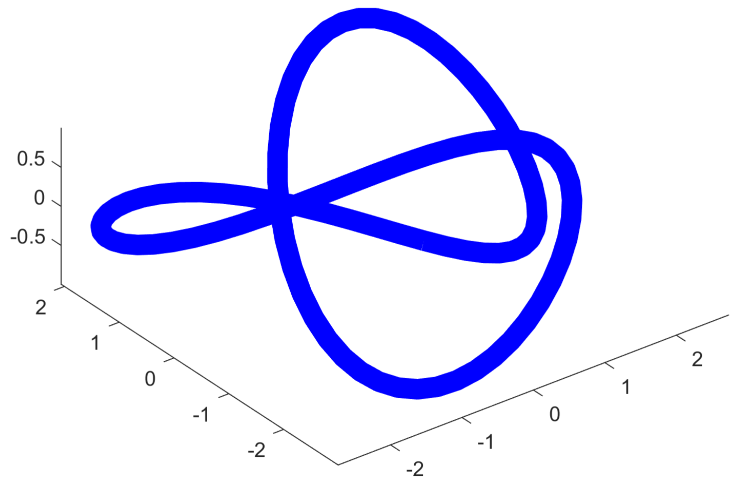
$$(x(t), y(t), z(t)) = (\sin(t) + 2\sin(2t),\ \cos(t) - 2\cos(2t),\ -\sin(3t)),$$

where t is a vector with 100 equally spaced points between 0 and 2π. Write a MATLAB script `q3a.m` to plot the 3D curve $\mathcal{C}$ with <u>blue solid lines</u>, <u>line width 10</u>, and <u>equal axis scales</u>. Include the code file `q3a.m` in your submission.

Solution:

```
t = linspace(0, 2*pi, 100);
x = sin(t) + 2*sin(2*t);
y = cos(t) - 2*cos(2*t);
z = -sin(3*t);
figure;
plot3(x,y,z,'b-','LineWidth',10);
axis equal
```

The figure is as follows:



Remark: This curve is also known as the trefoil knot.

(b) (7 marks) Consider the parametric surface $\mathcal{S}$ given by

$$\begin{cases} X(u,v) &= (4 + r\cos(v+2u))\cos u, \\ Y(u,v) &= (4 + r\cos(v+2u))\sin u, \\ Z(u,v) &= r\sin(v+2u), \end{cases}$$

where $0 \le u, v \le 2\pi$ and $r = \left(\cos^{20} v + \sin^{20} v\right)^{-1/20}$.

Write a MATLAB script `q3b.m` to do the following:

- Let u, v be two vectors, each with 100 equally spaced points between 0 and 2π. Use the `meshgrid` function to create two 2-D arrays `U`, `V` based on u and v.
- Create a surface plot of the surface $\mathcal{S}$ based on the 2-D arrays `U`, `V` and the

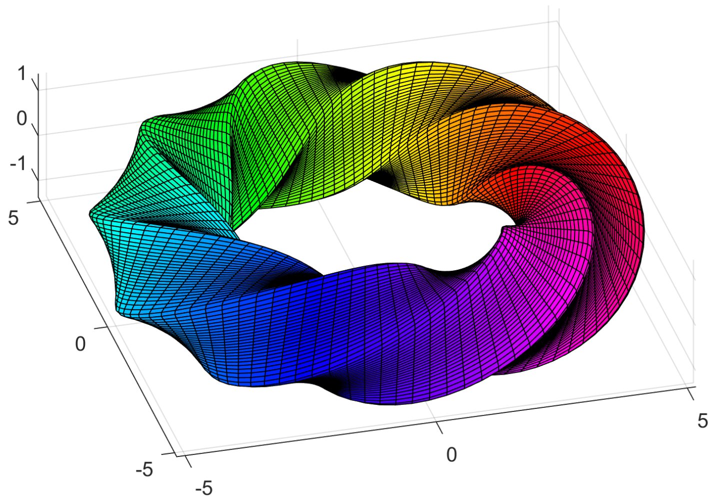
given parametric equations, with the face color set based on the `U` value, equal axis scales, the `hsv` color scheme, and the view angle $(-15, 30)$.

Include the code file `q3b.m` in your submission.

Solution:

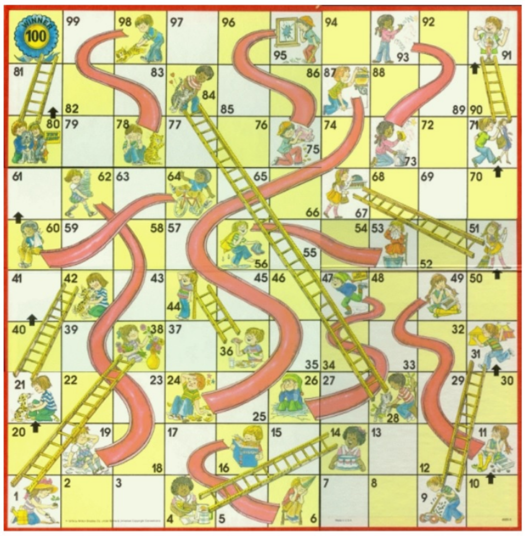
```
u = linspace(0, 2*pi, 100);
v = linspace(0, 2*pi, 100);
[U,V] = meshgrid(u,v);
a = 2;
n = 20;
r = (cos(V).^n + sin(V).^n).^(-1/n);
X = (4 + r.*cos(V+a*U)) .* cos(U);
Y = (4 + r.*cos(V+a*U)) .* sin(U);
Z = r.*sin(V+a*U);
figure;
surf(X,Y,Z,U);
axis equal
colormap hsv
view(-15,30)
```

The figure is as follows:



Remark: This surface is a twisted torus. The twist can be further adjusted by changing the values of a and n in the above code.

4. *Chutes and Ladders* (also known as *Health and Happiness Chess* or *Snakes and Ladders* in different countries) is a traditional board game played on a game board consisting of numbered boxes, with certain "chutes" and "ladders" connecting some of the boxes. The player will roll a dice repeatedly to navigate from the start (Box 1) to the goal, helped by climbing the ladders but hindered by falling down the chutes. Some typical game board examples are as follows:



In this question, we will implement the Chutes and Ladders game and perform some simulations to analyze the number of rounds needed to complete the game.

Consider the following game rules:

- The game board consists of 100 numbered boxes in total. Box 1 is the starting point and Box 100 is the goal.
- In each round, the player rolls a dice to get a random integer $d \in \{1, 2, 3, 4, 5, 6\}$.
- The player then moves from the current position (say, Box p) forward by d steps to Box $p + d$.
- If the player lands on Box 100 exactly, the game is finished.
- If the player reaches Box 100 but "overshoots" (e.g. currently at Box 99 and gets $d = 4$), he/she will need to move backward by the number of "overshot" steps (e.g. $99 \to 100 \to 99 \to 98 \to 97$; in the next round, the player will roll a dice and move forward again).

- If the player lands on one of the following special positions, he/she will immediately climb a ladder or fall down a chute to another position.

  Positions with ladders (9 in total):

  - $4 \rightarrow 14$ (if the player lands on Box 4, he/she will move to Box 14)
  - $9 \rightarrow 31$
  - $20 \rightarrow 38$
  - $28 \rightarrow 84$
  - $36 \rightarrow 44$
  - $40 \rightarrow 42$
  - $51 \rightarrow 67$
  - $71 \rightarrow 91$
  - $80 \rightarrow 81$

  Positions with chutes (10 in total):

  - $16 \rightarrow 6$ (if the player lands on Box 16, he/she will move to Box 6)
  - $47 \rightarrow 26$
  - $49 \rightarrow 11$
  - $56 \rightarrow 53$
  - $62 \rightarrow 19$
  - $64 \rightarrow 60$
  - $87 \rightarrow 24$
  - $93 \rightarrow 73$
  - $95 \rightarrow 75$
  - $98 \rightarrow 78$

(a) (10 marks) Write a MATLAB function `n = chutes_and_ladders()` that plays the Chutes and Ladders game as described above and outputs the number of rounds `n` needed for finishing the game (i.e., the number of times that the player rolls the dice in order to land on Box 100 exactly). Include the code file `chutes_and_ladders.m` in your submission.

Note: You may use the `randi` function for rolling the dice in each round.

(b) (5 marks) Write a MATLAB script `q4.m` to do the following:

- Repeat the Chutes and Ladders game for 1000 trials and record the number of rounds needed to finish the game for each trial.
- Create a histogram plot of all the 1000 recorded numbers and use the "probability" normalization.
- Label the x-axis as "Number of rounds needed" and the y-axis as "Probability", and add the title "Chutes and Ladders" to the figure.

Solution:

```
function n = chutes_and_ladders()
    pos = 1; % Initial position
    n = 0; % Round number

    % the starting points of all ladders and chutes
    special_pos = [4,9,20,28,36,40,51,71,80, ... % ladders
        16,47,49,56,62,64,87,93,95,98]; % chutes

    % the destinations of all ladders and chutes
    special_target = [14,31,38,84,44,42,67,91,81, ...  % ladders
        6,26,11,53,19,60,24,73,75,78]; % chutes

    % Repeat the game until the player lands on Box 100 exactly
    while pos ~= 100
        n = n+1; % update the round number
        d = randi(6); % roll a dice
        pos = pos + d; % update the position

        % Check whether overshoot occurs
        % Remark: This part should be put before checking ...
            chutes and ladders, as it may happen that you ...
            land on a chute position 98 after moving backward
        if pos > 100
            % overshoot, move backward
            pos = 100 - (pos - 100);
        end
```

```
        % check whether the current position is a starting ...
            point of a ladder or a chute
        id = find(special_pos == pos);
        if length(id) ~= 0 % or use ~isempty(id)
            % the current position is indeed a starting point ...
                of a ladder or a chute, so we immediately ...
                move to the target position
            pos = special_target(id);
        end

%          % Or you may also check it using a for-loop:
%          for i = 1:length(special_pos)
%              if pos == special_pos(i)
%                  pos = special_target(i);
%              end
%          end

        % Alternatively, you may also handle all ladders and ...
            chutes one by one using if-elseif-end (or ...
            switch-case) statements as follows:

%          % Climb the ladders
%          if pos == 4
%              pos = 14;
%          elseif pos == 9
%              pos = 31;
%          elseif pos == 20
%              pos = 38;
%          elseif pos == 28
%              pos = 84;
%          elseif pos == 36
%              pos = 44;
%          elseif pos == 40
%              pos = 42;
%          elseif pos == 51
%              pos = 67;
%          elseif pos == 71
```

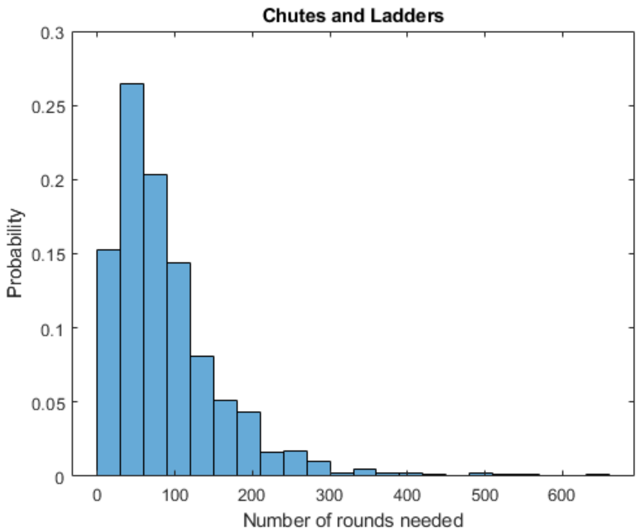
```
%             pos = 91;
%         elseif pos == 80
%             pos = 81;
%
%         % Fall down the chutes
%         elseif pos == 16
%             pos = 6;
%         elseif pos == 47
%             pos = 26;
%         elseif pos == 49
%             pos = 11;
%         elseif pos == 56
%             pos = 53;
%         elseif pos == 62
%             pos = 19;
%         elseif pos == 64
%             pos = 60;
%         elseif pos == 87
%             pos = 24;
%         elseif pos == 93
%             pos = 73;
%         elseif pos == 95
%             pos = 75;
%         elseif pos == 98
%             pos = 78;
%         end

    end
end
```

Content of `q4.m`:

```
X = zeros(1000,1);
for i = 1:1000
    X(i) = chutes_and_ladders();
end
```

```
figure;
histogram(X,'Normalization','probability');
xlabel('Number of rounds needed');
ylabel('Probability');
title('Chutes and Ladders');
```

The figure is as follows (note that the result will be different every time due to the randomness of dice rolling, but the overall trend should be similar):



Remarks:

- *For the history and the rule of different variations of the game, see Health and Happiness Chess (in Chinese only) and Snakes and Ladders.*
- *Mathematically, one can consider the game as a Markov chain and study its long-term behavior.*
- *As you can see in the game, while there may be ups and downs throughout the process, keep trying and you will eventually reach your goal (more precisely, with probability 1 as guaranteed by Mathematics)!*

End of Lab Assignment