

## Systems of Linear Equations: Solutions and Visualization

### Part A

#### Question 1: fprintf()

```
clc
% Set up A and b matrices
A = [1, 2, -3; 2, 1, -3];
b = [2; -2];
% Make augmented matrix
AM = [A, b];
% Print the size of the augmented matrix
fprintf("The size of the augmented matrix is: %d x %d\n", size(AM))
```

#### Question 2: rref()

```
clc
% Set up the A and b matrices
A = [1, 2, -3; 2, 1, -3];
b = [2; -2];
% Make the augmented matrix
AM = [A, b];
% Solve for rref of the augmented matrix
RAM = rref(AM);
% Display the RAM
disp("RAM =")
disp(RAM)
```

#### Question 3: Find the Pivots Programmatically

```
clc
% Set up the A and b matrices
A = [1, 2, -3; 2, 1, -3];
b = [2; -2];
% Make the augmented matrix
AM = [A, b];
% Solve for rref of the augmented matrix
RAM = rref(AM);
pivot_for_row1 = find(RAM(1,:), 1);
pivot_for_row2 = find(RAM(2,:), 1);
% Display results
fprintf("Pivot for row 1 is in column %d\nPivot for row 2 is in column %d\n", pivot_for_row1, pivot_for_row2)
```

#### Question 4: Write the solutions manually

```

clc
%{
    The basic variables are: x1 and x2
    The free variable is: x3
    The solution can be writte: x1=x3-2, x2=x3+2, x3=x3, where x3 is free
%}

```

## Part B

### Questions 5-6: 3D-Plot of the solutions

```

% Starter code with reference plane and axes in green.
clc, close all
figure
% Set up A and b matrices
A = [1, 2, -3; 2, 1, -3];
b = [2; -2];
% Make augmented matrix
AM = [A, b];
RAM_1 = rref(AM);
disp("RAM_1 =")
disp(RAM_1)
% Choose POLAR or CARTESIAN coordinates
use_polar_coordinates = true; % options: true (polar) or false (cartesian)
% Choose a colormap.
% try other colormaps such as cool, hot, hsv, prism, flag, jet, parula, lines
colormap cool
% choose preferred shading for surface plots
shading faceted % Shading methods are faceted, flat, and interp.
L = 5; % A default limit so we can easily control the axes.
% Set up a grid for each 3D plot.
x = -L : 1 : L; % x will range from -L to L.
y = -L : 1 : L; % y will range from -L to L.
[X,Y] = meshgrid(x,y);
% polar coordinates will be used instead, if you set use_polar_coordinates = true
if use_polar_coordinates
    [rad,theta] = meshgrid(0:1:5, 0:2*pi/36: (2*pi) );
    X = rad.*cos(theta);
    Y = rad.*sin(theta);
    Z = 0*X;
end
base = surf(X,Y,0*X + 0*Y); % Draw the xy plane. It's equation is z = 0.
hold on
grid on
% add some transparency to the base (xy-plane) for reference
base.set('facealpha', 0.4); base.set('edgealpha', 0.15);
base.set('facecolor', 'green');
darkgreen = [0 0.5 0 ];
% Adjust the axes limits for optimal view

```

```

L = L + 1;
zmin = -L; zmax = +L; % these have to be set by trial and error.
axis([-L, L, -L, L, zmin, zmax]) % set all three axis limits - we are in three space
axis equal
% Plot the coordinate axis in the background as reference objects
t = [-1, 1]; % used to compute endpoints along each axis
% plot the x-axis
x = L*t; y = 0*t; z = 0*t;
plot3( x, y, z, 'color', darkgreen, 'LineWidth', 3)
% plot the y-axis
x = 0*t; y = L*t; z = 0*t;
plot3( x, y, z, 'color', darkgreen, 'LineWidth', 3)
% plot the z-axis
x = 0*t; y = 0*t; z = [zmin, zmax];
plot3( x, y, z, 'color', darkgreen, 'LineWidth', 3)
set(gca, 'FontSize', 20)
% Add one reference marker at the tip of the positive x-axis.
plot3( L, 0, 0, 'bd', 'MarkerSize', 9, 'MarkerFaceColor', 'red')
% Add one reference marker at the tip of the positive y-axis.
plot3( 0, L, 0, 'bd', 'MarkerSize', 9, 'MarkerFaceColor', 'green')
% Add one reference marker at the tip of the positive z-axis.
plot3( 0, 0, zmax, 'bd', 'MarkerSize', 9, 'MarkerFaceColor', 'blue')
view([120, 20])

% My code:
% Make all label and text objects for plot
xlabel = 'x';
ylabel = 'y';
zlabel = 'z';
title = 'Visualization of a System of Equations';

% Define the vectors
z = -4 : 0.5 : 4;
x = z - 2;
y = z + 2;

% Plotting in 3D.
solution_set_as_line = plot3(x, y, z, 'blue-', 'LineWidth', 4);
solution_set_as_points = plot3(x, y, z, 'blueo', 'LineWidth', 1);
solution_set_as_points.set('MarkerFaceColor', 'yellow', 'MarkerSize', 12)

% Two ways to adjust the view, manually and programatically.
rotate3d on; view([100,10]); axis equal;

% Make figure full screen
set(gcf, 'Units', 'Normalized', 'OuterPosition', [0, 0.04, 1, 0.96]);
% Question 7-8 - Use surf command to show the two planes
% The meshgrid defined by arrays X and Y is still defined.
% We will use this to show each plane as a surface plot.
% First plane

```

```

a1 = A(1,1); a2 = A(1,2); a3 = A(1,3); d = b(1);
Z = (d - a1 * X - a2 * Y)/a3;
p1 = surf(X,Y,Z); % Draw the plane defined by the first equation.
p1.set('facealpha', 0.40); % Make the surface plot partially transparent.

```

% Students should add the second plane here.

% Access second row of coefficients

```

a1 = A(2,1); a2 = A(2,2); a3 = A(2,3); d = b(2);

```

% Create new Z variable for new plane

```

Z = (d - a1 * X - a2 * Y)/a3;

```

% Plot it with surf()

```

p2 = surf(X,Y,Z);

```

% Make plane transparent

```

p2.set('facealpha', 0.40);

```

% Question 9: Adding Two Test Planes

% Part a)

% Add them to the 3d-plot using the fill3 command.

```

P1 = [ 4; 0; +5];

```

```

P2 = [ 4; 0; -5];

```

```

P3 = [ 0; -4; -5];

```

```

P4 = [ 0; -4; +5];

```

```

rect1 = [P1 P2 P3 P4];

```

```

X = rect1(1, :); Y = rect1(2, :); Z = rect1(3, :);

```

```

plot3(X, Y, Z, 'bo', 'MarkerSize', 12, 'MarkerFaceColor', 'cyan')

```

```

rectangle = fill3(X, Y, Z, 'blue:');

```

```

rectangle.set('facealpha', 0.4)

```

% Add another plane on the plot

```

Q1 = [ 0; 4; +5];

```

```

Q2 = [ 0; 4; -5];

```

```

Q3 = [ -4; 0; -5];

```

```

Q4 = [ -4; 0; +5];

```

```

rect2 = [Q1 Q2 Q3 Q4];

```

```

X = rect2(1, :); Y = rect2(2, :); Z = rect2(3, :);

```

```

plot3(X, Y, Z, 'bo', 'MarkerSize', 12, 'MarkerFaceColor', 'red')

```

```

rectangle = fill3(X, Y, Z, 'magenta:');

```

```

rectangle.set('facealpha', 0.4)

```

% System 2

% Set up A and b matrices

```

A = [1, 2, -3; 2, 1, -3; 1, -1, 0];

```

```

b = [2; -2; 4];

```

% Make augmented matrix

```

AM = [A, b];

```

```

RAM_2 = rref(AM);

```

```

disp("RAM_2 =")

```

```

disp(RAM_2)

```

% System 3

```

% Set up A and b matrices
A = [1, 2, -3; 2, 1, -3; 1, -1, 0];
b = [2; -2; -4];
% Make augmented matrix
AM = [A, b];

RAM_3 = rref(AM);
disp("RAM_3 =")
disp(RAM_3)

% Question 10: Animate the graph
N = 2; % Number of complete revolutions of the azimuth. Elevation fixed at 20.
elevation = 20;
for k = 0:1:360*N
    view([k, elevation])
    pause(0.05);
end

```