Eric Wan - ezw23@drexel.edu - HW8

Table of Contents

Problem 1

```
%{
(a)
Domain: R^3
Range: R^4
Co-Domain: R^4
One-to-one or Onto: One-to-one b/c pivots same domain

(b)
Domain: R^6
Range: R^3
Co-Domain: R^3
One-to-one or Onto: Onto b/c range is the same as the codomain

(c)
Domain: R^4
Range: R^2
Co-Domain: R^2
One-to-one or Onto: Onto b/c range is the same as the codomain

(d)
Domain: R^4
Range: R^3
Co-Domain: R^3
One-to-one or Onto: Onto b/c range is the same as the codomain

(e)
Domain: R^3
Range: R^2
Co-Domain: R^3
One-to-one or Onto: Onto b/c range is the same as the codomain

(f)
Domain: R^4
Range: R^4
Co-Domain: R^4
One-to-one or Onto: Onto and One-to-one

One-to-one and Onto: To be both, the number of pivots has to be the
 same same as rows.
```

```
%}
```

Problem 2

```
%{
(a)
X = 3e1 + 2e2

(b)
No
X cannot be made

(c)
No

(d)
No as we need 4 pivots but only have 3
%}
```

Problem # 3 & 4

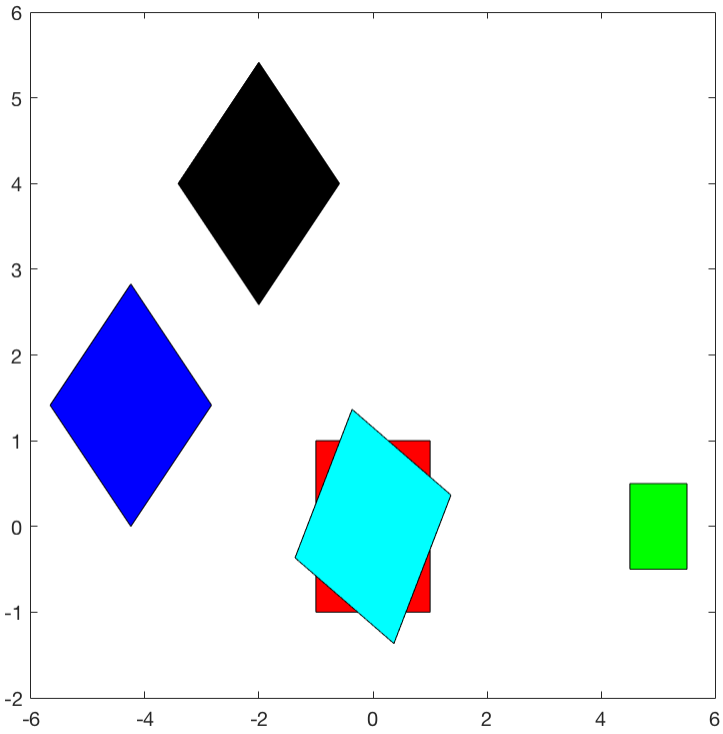
```
x_coords = [-1 1 1 -1]; % setting x coord for rectangle
y_coords = [-1 -1 1 1]; % setting y coord for rectangle
rectf = [x_coords; y_coords ; 1 1 1 1]; % creating rectangle
figure; % creating figure
fill(rectf(1,:),rectf(2,:), 'r') % filling in rectangle
axis square;
hold on;

% Part a
dx = -2;
dy = 4;
ang_deg = 45;
t = [1 0 dx; 0 1 dy; 0 0 1];
r = [cosd(ang_deg) -sind(ang_deg) 0; sind(ang_deg) cosd(ang_deg) 0; 0
 0 1];
% moved -2 in x-axis and +4 in y-axis, rotated by 45 degrees
M1 = r * t % simplified matrix
rectf1 = M1 * rectf; % new coordinates
% rectf1 = r*(t*rectf); % same equation rewritten to reference
fill(rectf1(1,:),rectf1(2,:), 'b') % filling in rectangle

% Part b
% rotated by 45 degrees, moved -2 in x-axis and +4 in y-axis,
M2 = t * r % simplified matrix
rectf2 = M2 * rectf; % new coordinates
% rectf2 = t * (r * rectf); % same equation rewritten to reference
fill(rectf2(1,:),rectf2(2,:), 'k') % filling in rectangle

% Part c
dx = 5;
dy = 0;
ang_deg = -90;
```

```
t = [1 0 dx; 0 1 dy; 0 0 1];
r = [cosd(ang_deg) -sind(ang_deg) 0; sind(ang_deg) cosd(ang_deg) 0; 0
 0 1];
s = [0.5 0 0; 0 0.5 0; 0 0 1];
% scaled to 1/2, rotated by 45 degrees, moved -2 in x-axis and +4 in
 y-axis
M3 = t * r * s % simplified matrix
rectf3 = M3 * rectf; % new coordinates
% rectf3 = (t * (r * (s * rectf) ) ); % same equation rewritten to
 reference
fill(rectf3(1,:),rectf3(2,:), 'g')

% Part d
ang_deg = 10;
r = [cosd(ang_deg) -sind(ang_deg) 0; sind(ang_deg) cosd(ang_deg) 0; 0
 0 1];
% rotated by 10 degrees 6 times
M4 = r^6
rectf4 = M4 * rectf;
% rectf4 = (r * (r * (r * (r * (r * (r * rectf) ) ) ) ) ); % same
 equation rewritten to reference
fill(rectf4(1,:),rectf4(2,:), 'cyan')

% Part e
dx = -2;
dy = 4;
t51 = [1 0 0; 0 1 dy; 0 0 1];
t52 = [1 0 dx; 0 1 0; 0 0 1];
% moving -2 in x-axis and +4 in y-axis
M51t = t51 * t52; % simplified
M52t = t52 * t51; % simplified
check1 = isequal(M51t, M52t)
% translation of points can be done in any order

ang_deg = 10;
r51 = [cosd(ang_deg) -sind(ang_deg) 0; sind(ang_deg) cosd(ang_deg) 0;
 0 0 1];
r52 = [cosd(-ang_deg) -sind(-ang_deg) 0; sind(-ang_deg) cosd(-ang_deg)
 0; 0 0 1];
% rotating 10 degrees and -10 degrees
M51r = r51 * r52; % simplified
M52r = r52 * r51; % simplified
check2 = isequal(M51r, M52r)
% just rotating points without translation can be done in any order

sd = [2 0 0; 0 2 0; 0 0 1];
sh = [0.5 0 0; 0 0.5 0; 0 0 1];
% scaling up by 2 and down by 1/2
M51s = sd * sh; % simplified
M52s = sd * sh; % simplified
check3 = isequal(M51s, M52s)
% just scaling points without translation can be done in any order

M51sr = sd * r51; % simplified
```

```
M52rs = r51 * sd; % simplified
check4 = isequal(M51sr, M52rs)
% scaling and rotating can be done in any order
```

```
M1 =

    0.7071   -0.7071   -4.2426
    0.7071    0.7071    1.4142
         0         0    1.0000


M2 =

    0.7071   -0.7071   -2.0000
    0.7071    0.7071    4.0000
         0         0    1.0000


M3 =

         0    0.5000    5.0000
   -0.5000         0         0
         0         0    1.0000


M4 =

    0.5000   -0.8660         0
    0.8660    0.5000         0
         0         0    1.0000


check1 =

  logical

   1


check2 =

  logical

   1


check3 =

  logical

   1
```

```
check4 =

  logical

   1
```

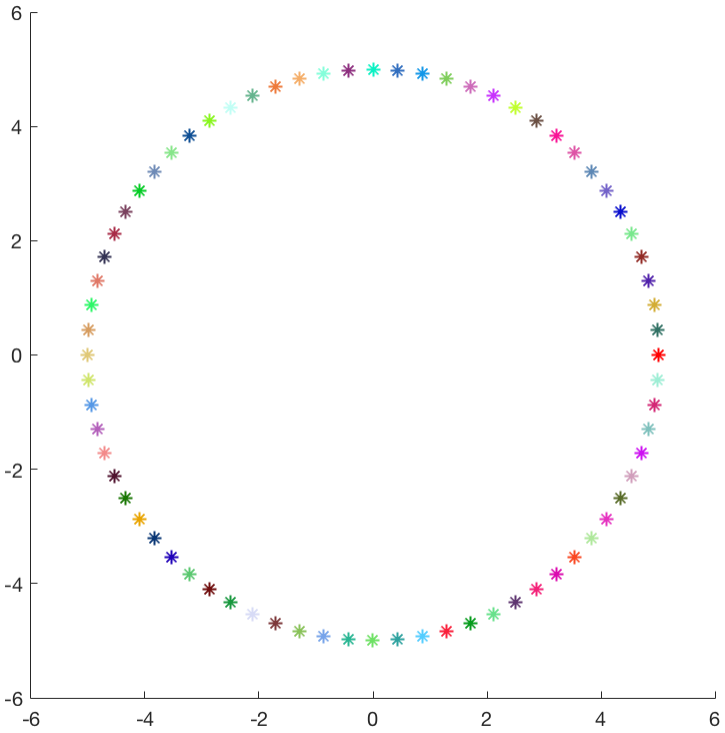


Problem # 4

Part b

```
figure;
hold on;
axis square;
p = [5; 0; 1]; % defining p
x = p(1,:); % defining x from p
y = p(2,:); % defining y from p
p1 = plot(x, y); % ploting x and y as initial point
p1.Marker = '*';
p1.Color = [1 0 0];

for i = 1: 5: 355 % for loop
    p = root(5)* p; % new p values every iteration
    x = p(1,:); % new x every iteration from p
    y = p(2,:); % new y every iteration from p
    p2 = plot(x, y); % plotting new x and y in circle
```

```
    p2.Marker = '*';
    p2.Color = [rand rand rand];
end
```



Published with MATLAB® R2017a