

**ENCE 201 Midterm 1, Open Notes and Open Book**

Name : Austin

E-mail (print neatly!): austin@umd.edu,

**Exam Format and Grading.** This take home midterm exam is open notes and open book. You need to comply with the **university regulations for academic integrity.**

There are three questions. Partial credit will be given for partially correct answers, so please show all your working.

Please see the class web page for instructions on how to submit your exam paper.

Question	Points	Score
1	10	
2	15	
3	15	
Total	40	

**Question 1: 10 points.** This question covers linear matrix equations and their solution.

Consider the family of equations:

$$\begin{aligned}x + ky &= 1 \\ kx + y &= 1\end{aligned}\tag{1}$$

where  $x$  and  $y$  are variables and  $k$  is a number.

[1a] (2 pts) Write equations 1 in matrix form  $A \cdot X = B$ .

$$\begin{bmatrix} 1 & k \\ k & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

[1b] (2 pts) In part [1a], what are the dimensions of matrices  $A$ ,  $X$  and  $B$ ?

$$\begin{aligned}[A][X] &= [B] \\ (2 \times 2) &(2 \times 1) \quad (2 \times 1)\end{aligned}$$

[1c] (2 pts) Write down the augmented matrix  $[A|B]$ .

$$\left[ \begin{array}{cc|c} 1 & k & 1 \\ k & 1 & 1 \end{array} \right]$$

[1d] (4 pts) Find values of  $k$  for which equations 1 will have exactly one solution. What are the corresponding values of  $x$  and  $y$ ?

Unique sol'n when  $\det(A) \neq 0$ , i.e.,

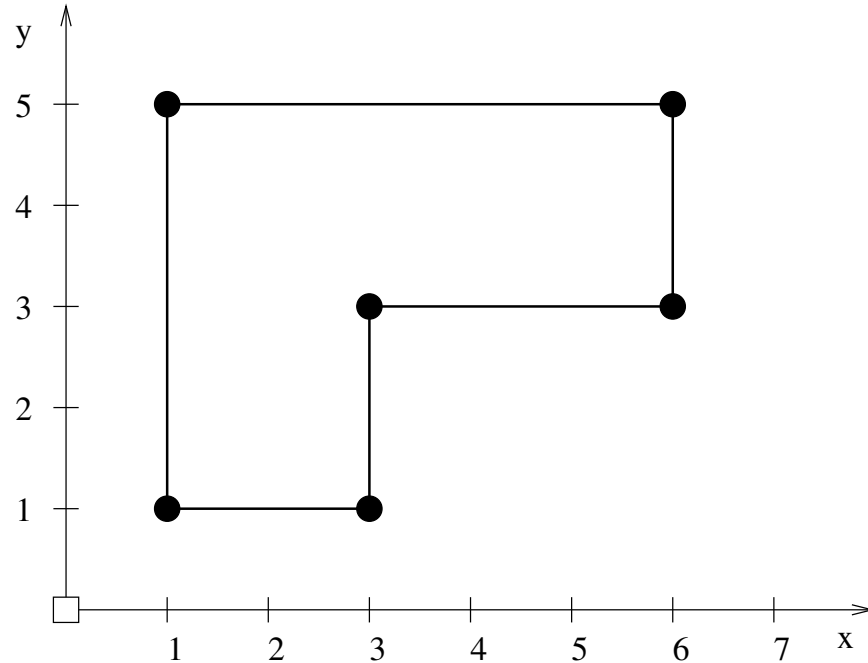
$$\det \begin{bmatrix} 1 & k \\ k & 1 \end{bmatrix} = 1 - k^2 \neq 0 \Rightarrow k \neq 1, k \neq -1.$$

Let  $x = y = \frac{1}{k+1}$ .

$$x + ky = \frac{1}{k+1} + \frac{k}{k+1} = \frac{1+k}{k+1} = 1 \quad \checkmark$$

$$kx + y = \frac{k}{k+1} + \frac{1}{k+1} = \frac{k+1}{k+1} = 1 \quad \checkmark$$

**Question 2 (15 points).** This question covers use of Python to compute and print the maximum and minimum coordinate positions, and perimeter of the six-sided polygon shown in Figure 1.



**Figure 1.** Six-sided irregular polygon.

Look the Python code over carefully and answer the questions that follow:

```
# =====
# TestPolygonSixSides.py: Compute max/min nodal coordinates and
# perimeter of six-sided polygon.
#
# Written by: Mark Austin                                     March 2024
# =====

import math
import numpy as np

# Function to print two-dimensional matrices ...

def PrintMatrix(name, a):
    print("Matrix: {:s} ".format(name) );
    for row in a:
        for col in row:
            print("{:8.4f}".format(col), end=" ")
        print("")

# Compute polygon perimeter ...

def perimeter( coord ):
    norows = coord.shape[0];
```

```

dperimeter = 0.0;
for i in range(1,norows):
    dx = coord[i][0] - coord[i-1][0];
    dy = coord[i][1] - coord[i-1][1];
    dperimeter = dperimeter + math.sqrt( dx*dx + dy*dy );

dx = coord[norows-1][0] - coord[0][0];
dy = coord[norows-1][1] - coord[0][1];
dperimeter = dperimeter + math.sqrt( dx*dx + dy*dy );

return dperimeter;

# =====
# main method ...
# =====

def main():
    print("---- Enter TestPolygonSixSides.main()           ... ");
    print("---- ===== ... ");

    print("--- Part 1: Initialize coefficients for matrix equations ... ");

    coord = np.array( [ ( 1.0, 1.0 ),
                        ( 1.0, 5.0 ),
                        ( 6.0, 5.0 ),
                        ( 6.0, 3.0 ),
                        ( 3.0, 3.0 ),
                        ( 3.0, 1.0 ) ] );

    PrintMatrix("Polygon Coordinates", coord);

    print("--- Part 2: Max/min coordinate positions ... ");

    print("--- Min x = {:f} ...".format( min ( coord[:,0] ) ) );
    print("--- Max x = {:f} ...".format( max ( coord[:,0] ) ) );
    print("--- Min y = {:f} ...".format( min ( coord[:,1] ) ) );
    print("--- Max y = {:f} ...".format( max ( coord[:,1] ) ) );

    print("--- Part 3: Compute and print distance of coords from origin ... ");

    distance = np.zeros( coord.shape[0] )
    for i in range(6):
        x = coord[i][0];
        y = coord[i][1];
        distance[i] = math.sqrt(x**2 + y**2)

    print("--- Min distance from origin = {:f} ...".format( min ( distance ) ) );
    print("--- Max distance from origin = {:f} ...".format( max ( distance ) ) );

    print("--- Part 4: Compute and print perimeter and area of polygon ... ");

    print("---- Polygon perimeter = {:f} ...".format( perimeter(coord) ) );

    print("---- ===== ... ");

```

```

    print("--- Enter TestPolygonSixSides.main()          ... ");

# call the main method ...

if __name__ == "__main__":
    main()

```

The program output is:

```

--- Enter TestPolygonSixSides.main()          ...
--- ===== ...
--- Part 1: Initialize coefficients for matrix equations ...
Matrix: Polygon Coordinates
  1.0000  1.0000
  1.0000  5.0000
  6.0000  5.0000
  6.0000  3.0000
  3.0000  3.0000
  3.0000  1.0000
--- Part 2: Max/min coordinate positions ...
--- Min x = 1.000000 ...
--- Max x = 6.000000 ...
--- Min y = 1.000000 ...
--- Max y = 5.000000 ...
--- Part 3: Compute and print distance of coords from origin ...
--- Min distance from origin = 1.414214 ...
--- Max distance from origin = 7.810250 ...
--- Part 4: Compute and print perimeter and area of polygon ...
--- Polygon perimeter = 18.000000 ...
--- ===== ...
--- Enter TestPolygonSixSides.main()          ...

```

[2a] (2 pts) What does the line:

```
import numpy as np
```

do, and why is it needed in this program?

Imports numpy (a library that performs common mathematical operations such as array functions & linear algebra) as np.

np.zeros() returns a new array filled with zeros, and is subsequently used to store the distance between the origin & nodal coordinates.

[2b] (2 pts) What does the line:

```
import math
```

do, and why is it needed in this program?

Imports the math library; this program uses `math.sqrt(...)` in the computation of distances between the origin & nodal coordinates.

[2c] (2 pts) Write a small script of Python code to retrieve and print the matrix dimensions, and number of rows and columns in the coord matrix.

```
dimensions = coord.shape;
print("--- number of rows:", dimensions[0]);
print("--- number of columns:", dimensions[1]);
```

[2d] (3 pts) Briefly explain how the various parts (i.e., use of builtin functions and formatting specification) of the statement:

```
print("--- Min x = {:.f} ...".format( min ( coord[:,0] ) ) );
```

work.

- (1) `coord[:,0]` selects the first column of the coord matrix.
- (2) `min(coord[:,0])` finds the minimum value in the selected column.
- (3) `{:.f}` formats the minimum value as a floating-point value.
- (4) `print(...)` prints string + formatted numerical value.

[2e] (3 pts) Now let's consider the block of code:

```
distance = np.zeros( coord.shape[0] )
for i in range(6):
    x = coord[i][0];
    y = coord[i][1];
    distance[i] = math.sqrt(x**2 + y**2)
```

Create a table that shows the iteration value  $i$ , and values of  $x$ ,  $y$  and  $distance[i]$ , for each iteration of the loop computation.

Iteration $i$	$x$	$y$	$distance[i]$
0	1	1	1.414 ← min distance
1	1	5	5.099
2	6	5	7.810 ← max distance
3	6	3	6.708
4	3	3	4.243
5	3	1	3.162

[2f] (3 pts) Finally, consider the statement:

```
print("--- Polygon perimeter = {:.f} ...".format( perimeter(coord) ) );
```

and the block of code:

```
dperimeter = 0.0;
for i in range(1,norows):
    dx = coord[i][0] - coord[i-1][0];
    dy = coord[i][1] - coord[i-1][1];
    dperimeter = dperimeter + math.sqrt( dx*dx + dy*dy );

dx = coord[norows-1][0] - coord[0][0];
dy = coord[norows-1][1] - coord[0][1];
dperimeter = dperimeter + math.sqrt( dx*dx + dy*dy );
```

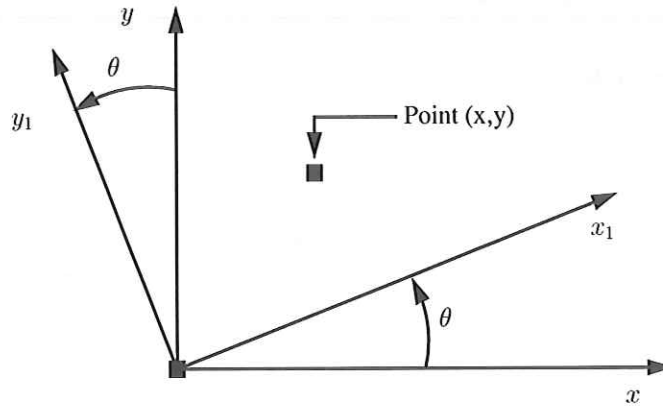
within the function perimeter. Construct a table that shows how the perimeter computation systematically assembles the polygon perimeter.

*dperimeter = 0*

Iteration (i)	dx	dy	dperimeter.
1	$1-1=0$	$5-1=4$	$0+4=4$
2	$6-1=5$	$5-5=0$	$4+5=9$
3	$6-6=0$	$3-5=-2$	$9+2=11$
4	$3-6=-3$	$3-3=0$	$11+3=14$
5	$3-3=0$	$1-3=-2$	$14+2=16$
<u>Last step:</u> connect nodes 1 & 6.	$3-1=2$	$1-1=0$	$16+2=18 \leftarrow \text{Result.}$



**Question 3 (15 points).** Suppose that a  $x$ - $y$  coordinate system is rotated anticlockwise by an angle  $\theta$  to create a new coordinate system  $x_1$ - $y_1$ .



The matrix product:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (2)$$

describes how points in the  $x$ - $y$  coordinate system are transformed into the  $x_1$ - $y_1$  coordinate system. Let us denote the 2-by-2 coordinate transformation matrix  $A(\theta)$ .

[3a] (2 pts) Is matrix  $A(\theta)$  skew-symmetric?

Skew symmetric when  $A^T = -A$ .

$$A(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \rightarrow \begin{matrix} -A = \begin{bmatrix} -\cos(\theta) & -\sin(\theta) \\ \sin(\theta) & -\cos(\theta) \end{bmatrix} \\ A^T = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \end{matrix} \left. \vphantom{\begin{matrix} -A \\ A^T \end{matrix}} \right\} \begin{matrix} -A \neq A^T \\ \text{Not skew} \\ \text{symmetric} \end{matrix}$$

[3b] (3 pts) For two rotations  $\theta_1$  and  $\theta_2$  verify that:

$$A(\theta_1)A(\theta_2) = A(\theta_1 + \theta_2). \quad (3)$$

Show all of your working ...

$$\begin{aligned} A(\theta_1)A(\theta_2) &= \begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) \\ -\sin(\theta_1) & \cos(\theta_1) \end{bmatrix} \begin{bmatrix} \cos(\theta_2) & \sin(\theta_2) \\ -\sin(\theta_2) & \cos(\theta_2) \end{bmatrix} \\ &= \begin{bmatrix} \cos(\theta_1)\cos(\theta_2) - \sin(\theta_1)\sin(\theta_2), & \cos(\theta_1)\sin(\theta_2) + \sin(\theta_1)\cos(\theta_2) \\ -\sin(\theta_1)\cos(\theta_2) + \cos(\theta_1)(-\sin(\theta_2)), & -\sin(\theta_1)\sin(\theta_2) + \cos(\theta_1)\cos(\theta_2) \end{bmatrix} \end{aligned}$$

[3b] continued ...

$$\approx \begin{bmatrix} \cos(\theta_1 + \theta_2) & \sin(\theta_1 + \theta_2) \\ -\sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) \end{bmatrix}$$

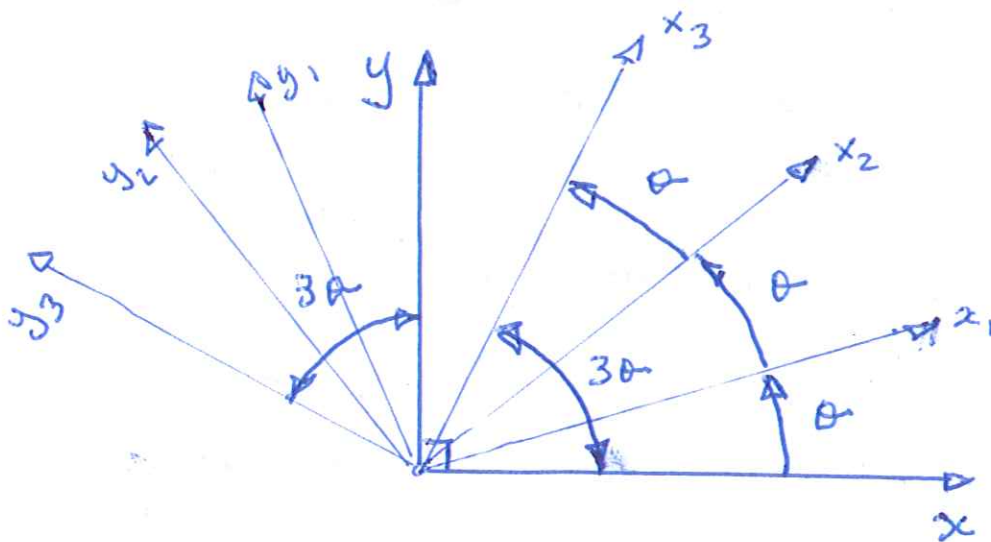
$$\approx A(\theta_1 + \theta_2).$$

Note: If  $\theta_1 = \theta_2 \Rightarrow A(\theta_1) A(\theta_1) = A(2\theta_1)$ .

Let  $s = \sin(\theta_1)$ ,  $c = \cos(\theta_1)$

$$A(2\theta) = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix} = \begin{bmatrix} c^2 - s^2 & 2cs \\ -2cs & c^2 - s^2 \end{bmatrix}.$$

[3c] (2 pts) Now suppose that the coordinate axes are rotated anticlockwise by angle  $\theta$  three times. Draw and label a sequence of coordinate systems (similar to the diagram above) that illustrate the essential features of this scenario.



[3d] (5 pts) Using the ideas and results of Parts [3b] and [3c], derive a formula for  $\cos(3\theta)$  in terms of  $\cos(\theta)$  alone, i.e.,

$$\cos(3\theta) = 4\cos^3(\theta) - 3\cos(\theta). \quad (4)$$

Show all of your working.

From 3b,  $A(3\theta) = A(2\theta) \cdot A(\theta)$ .

$$= \begin{bmatrix} c^2 - s^2 & 2cs \\ -2cs & c^2 - s^2 \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix}$$

$$= \begin{bmatrix} \cos(3\theta) & \sin(3\theta) \\ -\sin(3\theta) & \cos(3\theta) \end{bmatrix}.$$

Equating terms:

$$\cos(3\theta) = (c^2 - s^2)c - 2cs^2$$

$$= c^3 - cs^2 - 2cs^2.$$

$$= c^3 - 3cs^2. \quad \text{--- (A)}$$

Recall:  $\sin^2(\theta) + \cos^2(\theta) = 1$  --- (B)

Combining (A) & (B):

$$\cos(3\theta) = \cos^3(\theta) - 3\cos(\theta)(1 - \cos^2(\theta))$$

$$= 4\cos^3(\theta) - 3\cos(\theta).$$

[3e] (3 pts) Show that for all values of  $x, y, z$  and  $a \geq 0$  the triple product of matrices:

$$\begin{bmatrix} x & y & z \end{bmatrix} \cdot \begin{bmatrix} a & -a & 0 \\ -a & 1+2a & -a \\ 0 & -a & 1+2a \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \geq 0 \quad (5)$$

Multiply 2nd & third matrices:

$$= \begin{bmatrix} x & y & z \end{bmatrix} \cdot \begin{bmatrix} ax - ay \\ -ax + (1+2a)y - az \\ -ay + (1+2a)z \end{bmatrix}$$

$$= ax^2 - axy - axy + (1+2a)y^2 - ayz - ayz + (1+2a)z^2$$

Collecting terms:

$$= a(x^2 - 2xy + y^2) + y^2 + a(y^2 - 2yz + z^2) + (1+a)z^2$$

$$= a(x-y)^2 + a(y-z)^2 + y^2 + (1+a)z^2 \geq 0$$