UECM1703 Test 1 Marking Guide

Name: Student ID: Mark: /15

Course Code & Course Title: UECM1703 Introduction to Scientific Computing
Faculty: LKC FES, UTAR Course: AM, FM
Session: Oct 2019 Lecturer: Liew How Hui

Instruction: Answer all questions in the space provided. **If you do not write your answer in the space provided, you will get ZERO mark**. An answer without working steps may also receive ZERO mark.

1. CO1: Perform vector and matrix operation using computer software

   (a) Given the following matrix which is encoded as Numpy array `A`.

$$\texttt{A} = \begin{bmatrix} 25 & 35 & 26 & 36 & 11 & 36 & 29 & 10 & 36 \\ 18 & 34 & 13 & 26 & 10 & 37 & 45 & 43 & 39 \\ 38 & 11 & 13 & 13 & 21 & 16 & 42 & 28 & 44 \\ 39 & 44 & 35 & 29 & 17 & 29 & 37 & 22 & 40 \\ 28 & 37 & 33 & 38 & 11 & 42 & 25 & 35 & 22 \end{bmatrix}$$

   Write down the output of the following Python commands:

   i. `print(A[:,2])` (1 mark)

   Ans. [26 13 13 35 33] ..... [1 mark]

   ii. `print(A[1:3,-2])` (1.5 marks)

   Ans. [43 28] ..... [1.5 marks]

   iii. `print(A[:,1:7:2].max(axis=0))` (1.5 marks)

   Ans. [44 38 42] ..... [1.5 marks]

   iv. `print(A[:,:4]-np.arange(1,5))` (2 marks)

   Ans.

$$\begin{bmatrix} 24 & 33 & 23 & 32 \\ 17 & 32 & 10 & 22 \\ 37 & 9 & 10 & 9 \\ 38 & 42 & 32 & 25 \\ 27 & 35 & 30 & 34 \end{bmatrix}$$

   ..... [2 marks]

   v. `print(np.inner(A[:,1],A[:,2]))` (1 mark)

   Ans.

$$\begin{bmatrix} 35 & 34 & 11 & 44 & 37 \end{bmatrix} \cdot \begin{bmatrix} 26 & 13 & 13 & 35 & 33 \end{bmatrix} = 4256$$

   ..... [1 mark]

(b) An element in a matrix is said to be out of bound if it is less than zero or greater than 10. Write down a Python command to **count** the elements in a matrix `A` which are out of bound. (1 mark)

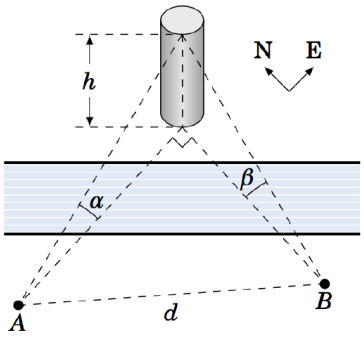
Ans. `((A<0)|(A>10)).sum()` ..... [1 mark]

2. CO3: Write program scripts for mathematical software

(a) Write down the output of the command `print(type("1"))`. (1 mark)

Ans. `<class 'str'>` ..... [1 mark]

(b) A tower on one side of a river is directly east and north of points A and B, respectively, on the other side of the river. The top of the tower has angles of elevation α and β from A and B, respectively, both units in degree as in the picture below.



Let d be the distance between A and B. Assuming that both sides of the river are at the same elevation, the height h of the tower is

$$h = \frac{d}{\sqrt{\cot^2\alpha + \cot^2\beta}}.$$

Implement the formula as a **Python function** `height(d,alph,beta)` in a Python script and write down the output the following command using calculator (accurate to 6 decimal places):

```
print(f"height = {height(6, 31, 22)}")
```

(3 marks)

Ans. The Python function is implemented as follows:

```
1  def height(d, alph, beta):                          # [0.3 mark]
2      from math import sqrt, tan, radians             # [0.3 mark]
3      _alph, _beta = radians(alph), radians(beta)     # [0.3 mark]
4      h = d/sqrt(1/tan(_alph)**2 + 1/tan(_beta)**2)   # [0.9 mark]
5      return h                                        # [0.2 mark]
```

By using scientific calculator, it is possible to derive the print output to be 'height = 2.011670'. ..... [1 mark]

(c) Given a linear system $M\boldsymbol{x} = \boldsymbol{b}$ where the entries of the $n \times n$ matrix $M = (m_{ij})$ are given by

$$m_{ij} = \begin{cases} a + (i-1)d, & i = j = 1, \cdots, n, \\ 1, & i = 1, \cdots, n-1,\ j = i+1, \\ 1, & i = j+1,\ j = 1, \cdots, n-1, \\ 0, & \text{otherwise} \end{cases} \qquad (*)$$

and a is a constant **larger than or equal to** 2 (otherwise, a None is returned) and d is a positive value. Write

(I) a Python script to implement the Python function `trid(a,d,n)` which takes in parameters a, d and n and returns a matrix M defined by the equation $(*)$. For example, `trid(0,0.1,6)` returns None and `trid(2,0.1,6)` generates

$$\texttt{M} = \begin{bmatrix} 2. & 1. & 0. & 0. & 0. & 0. \\ 1. & 2.1 & 1. & 0. & 0. & 0. \\ 0. & 1. & 2.2 & 1. & 0. & 0. \\ 0. & 0. & 1. & 2.3 & 1. & 0. \\ 0. & 0. & 0. & 1. & 2.4 & 1. \\ 0. & 0. & 0. & 0. & 1. & 2.5 \end{bmatrix};$$

(II) a Python command to solve for $\boldsymbol{x}$ if `a`, `d`, `n` and $\boldsymbol{b}$ are given. (3 marks)

Ans. (I)

```
1  def trid(a,d,n:int):                                    # [0.4 mark]
2      import numpy as np
3      if a < 2: return None                               # [0.2 mark]
4      M = np.diag(np.linspace(a,a+(n-1)*d,n))             # [1   mark]
5      for i in range(n-1):                                # [0.5 mark]
6          M[i+1,i] = M[i,i+1] = 1                         # [0.5 mark]
7      return M                                            # [0.1 mark]
```

(II) If `a`, `d`, `n` and $\boldsymbol{b}$ are given,

```
from scipy import linalg                                   # [0.1 mark]
x = linalg.solve(trid(a,d,n),b)                            # [0.2 mark]
```