Outline

Practical 2a

Due to its commercial value, SPM past year exam questions for mathmatics are difficult to find on the Internet. You are encouraged to use what you learned in Topic 1 to explore how to solve SPM mathematics. In this part, we will try to explore how to solve a few linear algebra problems from UTAR foundation.

Given $D = \begin{bmatrix} -5 & 3 & 4 \\ -1 & 0 & -1 \\ 3 & -3 & -3 \end{bmatrix}$ and $F = \begin{bmatrix} -1 & -1 & -1 \\ -2 & 1 & -3 \\ 1 & -2 & 1 \end{bmatrix}$.

Estimate Time: 10 minutes

Practical 2a (cont)

1. Write down the Python commands to store D and F. [Make sure you import the right module to do your job.]
2. Write down the Python command to give you the matrix product DF.
3. Write down the Python command to given you D^{-1} and F^{-1}.
4. Solve the following system of linear equations:

$$\begin{aligned} x + y + z &= -2 \\ 2x - y + 3z &= -20 \\ -x + 2y - z &= 14 \end{aligned}$$

Practical 2b

Given $A = \begin{bmatrix} -2 & 3 & 1 \\ -3 & -1 & 2 \\ 5 & 3 & 2 \end{bmatrix}$ and B is the adjoint of matrix A, i.e. $AB = (\det A)I$. Write down the Python commands to find B and the inverse of matrix A.

Estimate Time: 5 minutes

Practical 2c

Given that $P = \begin{bmatrix} -1 & 0 \\ 1 & 3 \end{bmatrix}$, $Q = \begin{bmatrix} 3 & -2 \\ 1 & -1 \end{bmatrix}$ and $R = \begin{bmatrix} 0 & 1 \\ 3 & -2 \end{bmatrix}$

1. Write down the Python commands to find P^2 and Q^{-1}.
2. Write down the Python command to find matrix L if $P^2LQ = R$.

Estimate Time: 5 minutes

Practical 2d

Mr. Cat makes three types of mini cupcakes X, Y and Z. The ingredients for each type of the cupcake are as follows:

Type of	Ingredient		
Cupcake	Flour	Sugar	Butter
X	10 g	10 g	5 g
Y	7 g	5 g	4 g
Z	4 g	2 g	3 g

The amounts of ingredients available are 660 g of flour, 500 g of sugar and 395 g of butter.

Practical 2d (cont)

1. Obtain a matrix equation to represent the information above in Python.
2. Use Python to determine the number of each type of cupcakes to be made.

Estimate Time: 10 minutes

Practical 2e: Working with “Arrays” from University Subjects

Given some 1-D data, the statistical summary is as follows:

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.000	9.375	11.950	11.337	13.825	17.100

Implement the statistical summary using Python using the functions from Numpy library.

Estimate Time: 15 minutes

Outline

Practical 2f: Theory of Interest

Key idea: Cashflows, interest.

Example 1: If every month you can save 1000 at the end of the month from Jan to Aug. Suppose you will get an increment of 200 from Sep onward and you will put all the increments into your saving. With an annual interest of 2.5%, what is the future value at the end of the year?

Future value

$$
\begin{aligned}
=&1000\left(1+\frac{0.025}{12}\right)^{11}+\cdots+1000\left(1+\frac{0.025}{12}\right)^{4}+ \\
&1200\left(1+\frac{0.025}{12}\right)^{3}+\cdots+1200\left(1+\frac{0.025}{12}\right)^{0}
\end{aligned}
$$

Estimate Time: 15 minutes

Practical 2f: TOI (cont)

Example 2: Suppose an investment has the following projected cashflow (assuming the cash are invested and returned in the beginning of the years):

Year	2020	2021	2022	2023	2024	2025
Investment	-10000	-5000	-1000	-1000	-1000	-1000
Return	0	0	0	0	0	0
Year	2026	2027	2028	2029	2030	
Investment	-1000	-1000	-1000	-1000	-1000	
Return	+8000	+9000	+10000	+11000	+12000	

If the average annual interest rate is estimated to be 3%, is the investment worthy?

Tactic 1: Find present value (using Excel & Python)

Tactic 2: Find IRR (Internal Rate of Return)

Estimate Time: 30 minutes

Outline

Working with Image Arrays

An image array can be 2-D or 3-D depending on whether it is “grey-scale” or “coloured”. In the case of coloured image, red, green, blue are required an this means that a $m \times n \times 3$-array is required. Scipy plans to remove its image IO utilities and is promoting the use of `imageio` or `matplotlib` module. By using `matplotlib.pyplot` (which uses `pillow`, a PIL for Python 3), it is possible to use `imread` to read typical images such as Jpeg, Png and Bmp and use `imshow` and `imsave` to view and save the image.

Working with Image Arrays (cont)

The module `scipy.ndimage` provides many image array processing functions for *measuring*, *filtering*, *interpolating* and *morphing* a given image. To test and add more functions, newer Scipy package includes two images `ascent` and `face`.

In this part, we will first explore the image array manipulation of the two standard images.

```
>>> from scipy import misc
>>> ascent = misc.ascent()
>>> face = misc.face()
>>> import matplotlib.pyplot as plt
>>> plt.imshow(ascent, cmap=plt.cm.gray)
>>> plt.show()
```

Working with Image Arrays (cont)

1. How do you know that the type of the image `ascent` and its dimension using Python?
2. How "large" is the image `ascent` (by pixels?)
3. Write down the commands to find minimum, maximum and average values of the image `ascent`.
4. Explain what does the following commands do?

```
face2 = face[:-200,200:-50]
plt.imshow(face2)
plt.show()
```

Estimate Time: 30 minutes

Working with Image Arrays (cont)

1. Write down the Python commands to generate the following image on the left from `face`:

Try to write your commands so that it is easy for you to import your favourite image and generate something like the right figure.

More Work with Image Arrays

For a colour image,
(a) Show the original image and the red, green and blue components.
(b) show it in a 3x2 plot with the original image, histogram, the image with default colour map and

- Convert the colour image to grayscale images:
  - AverageGray = $\frac{1}{3}$(R + G + B)
  - Grayscale = 0.3 R + 0.59 G + 0.11 B
  - Luminosity = 0.2126 R + 0.7152 G + 0.0722 B
- Rotation can be performed on a vector centred at 0 using the rotation matrix $\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$.

  Try to rotate the image using this matrix (if time permits).

Estimate: 1 hour

Practical : Working with Solving PDE

In this part, the practical class tries to explore how to code mathematical formulas from the lecture.
The Numerical Method "CTCS Explicit Scheme":

1. GENERATE A GRID: Let $u_{i,j} = u(ih, jk) = u(x_i, t_j)$, $i = 0, \cdots, M$, $j = 0, \cdots, N$.
2. NUMERICAL DIFFERENCE APPROXIMATION:

$$u_{i,j+1} = ru_{i+1,j} + 2(1-r)u_{i,j} + ru_{i-1,j} - u_{i,j-1}, \; r = \left(\frac{\alpha k}{h}\right)^2. \tag{1}$$

Practical : 1-D Wave Equation (cont)

3. The homogeneous boundary conditions (BC):
   $u_{0,j} = u_{M,j} = 0, \quad j = 1, \cdots, N.$
4. The initial conditions (IC):
   $$u_{i,0} = u(x_i, 0) = f(x_i), \quad i = 0, 1, \cdots, M,$$
   $$u_{i,1} = (1-r)f(x_i) + \frac{r}{2}(f(x_{i+1}) + f(x_{i-1})) + kg(x_i).$$
5. SOLVE THE ITERATION (1) FOR A GIVEN INITIAL CONDITION.

Ref: https://wiki.seg.org/wiki/Solving_the_wave_equation_in_1D

Estimate: 1 hour