Outline

Using Python as a Calculator

- `from math import *`
- Q: How do you know that functions have been loaded from the library `math`?
- Q: How to get help on a particular function, e.g. sin?

Solving high school problems by (i) **express them in Python** (ii) run the Python command and get the "numerical" answer:

- $\cos 15^\circ \cdot |\cos 225^\circ| - \sin 315^\circ \cdot |\cos 105^\circ|$
- In a triangle $\triangle ABC$, $\cos A = \frac{2}{5}\sqrt{5}$, $\cos B = \frac{3}{10}\sqrt{10}$, find C (in degree).

Time: 10 mins

Outline

Getting familiar with Numbers

Use Python to find out the answers for the following questions:

- Which of the number $\frac{2^6}{2^6-1}$ and $(1-\frac{1}{2^6})^{-1}$ is larger?
- Estimate the square root of $2+3\frac{\sqrt{6}-1}{(\sqrt{6}+1)^3}$
- Solve $5^x = 31$ for the numeric approximation of x to 8 decimal places.
- Calculate $\frac{3+4i}{3-4i}$ using Python. Do you know how to find it using hand calculation?

Time: 10 mins

Elementary Functions

- Are $\sin 50^\circ$ and $\sin 50$ the same? Discover their numeric values using Python
- Is $\sin^2 50^\circ + \cos^2 50^\circ = 1$?
- Try $\tan 0^\circ$, $\tan 45^\circ$ and $\tan 90^\circ$.
- Find $e^{2\pi\sqrt{-1}}$ using complex arithmetic.

Time: 10 mins

Defining functions

In UECM3034 Numerical Methods (for AM & FM), there are algorithms (e.g. newton, fsolve, root from scipy.optimize) which can estimate the solution of the following equation

$$f(x) = 0.$$

If we want to solve the nonlinear equation

$$x = \frac{1}{2}(e^{x/2} + x)$$

using Python, we need to define an appropriate function after rearranging the given equation.

Time: 10 mins

Outline

Formatting Numbers

We need to print format numbers when we are dealing with large numbers or small numbers or printing them in a table.

Let us run the following commands to get a feel of formatting numbers. Time: 5 mins

```
from math import pi
num1 = pi**50
num2 = pi**-50
num3 = pi
print("\nUsing default formatting in Python ...")
print("num1 = {n1}".format(n1=num1))
print("num2 = {n2}".format(n2=num2))
print("num3 = {n3}".format(n3=num3))
print("\nFix number of decimal places to 6 decimal places ...")
print("num1 = {n1:.6f}".format(n1=num1))
print("num2 = {n2:.6f}".format(n2=num2))
print("num3 = {n3:.6f}".format(n3=num3))
print("\nA mix of decimal format or engineering format (6 effective figur
print("num1 = {n1:6g}".format(n1=num1))
print("num2 = {n2:6g}".format(n2=num2))
```

Handling Strings

- Python provide a way to create a string with multiple characters: E.g. ”=”*20
- Formating a string: ”abc”.upper(), ”ABC”.lower(), ”a:10s”.format(a=”a string”), etc.
- Printing strings: print(”one string”, ”two string”, ”three string”, end=””)

Time: 5 mins (10 mins break + Q&A)

Outline

Creating 1-D Array

There are two basic types 1-D array:

- Arithmetic sequence: a, $a + d$, ..., $a + (n - 1)d$. For example,
  - 1, 3, 5, 7, ..., 1111
  - 3567, 3562, 3557, ..., 12, 7
- Geometric sequence: a, ar, ar^2, ..., ar^{n-1}
  - 1, 1/2, 1/4, 1/8, ..., 1/1048576
  - 10.0, 7.071068, 5.0, 3.535534, 2.5, 1.767767, 1.25, 0.883883, 0.625, 0.441942, 0.3125, 0.220971, 0.15625, 0.110485, 0.078125, 0.055243, 0.039062, 0.027621, 0.019531, 0.013811

Write down the Python command to generate them.

Time: 15 mins

Comparing Arrays

Although the equality '==' is mentioned in the lecture slide, the comparison between two arrays are not metioned. The command for comparing two arrays is `np.array_equal(A,B)`. Try to determine the equality of the objects below.

- np.zeros(1) vs 0
- np.zeros((1,1)) vs np.array([0])
- np.zeros((1,1)) vs np.array([[0]])
- np.ones((10,10)).ravel() vs np.ones(100)
- np.r_[1:5].reshape((2,2)) vs np.array([[1,2],[3,4]])
- np.r_[5:1] vs []
- np.r_[5:1:-1] vs np.array([5,4,3,2])
- Feel free to suggest more ...

Time: 15 mins

Special Matrices

In Science and Engineering, it is important to recognise special matrices because we not only need to generate them for calculations but also to use special methods to speed up the solution process.
Identify the command to generate the following matrices:

$$A = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 10 & 10 & 10 \\ 10 & 10 & 10 \\ 10 & 10 & 10 \\ 10 & 10 & 10 \end{bmatrix}$$

Time: 5 mins

Special Matrices (cont)

Identify the command to generate the following tri-diagonal matrix:

$$\begin{bmatrix} 3 & -2 & 0 & 0 & 0 & 0 & 0 \\ -2 & 4 & -2 & 0 & 0 & 0 & 0 \\ 0 & -2 & 3 & -2 & 0 & 0 & 0 \\ 0 & 0 & -2 & 4 & -2 & 0 & 0 \\ 0 & 0 & 0 & -2 & 3 & -2 & 0 \\ 0 & 0 & 0 & 0 & -2 & 4 & -2 \\ 0 & 0 & 0 & 0 & 0 & -2 & 3 \end{bmatrix}$$

I think there are two ways to generate this special matrix without using loop (to be introduced by Dr Goh):

- array indexing
- elementwise operations

Time: 5 mins

Special Matrices (cont)

Identify the command to generate the following block matrix in two commands without keying in the values:

$$\begin{bmatrix} 1 & 0 & 8 & 7 & 6 & 5 \\ 0 & 1 & 4 & 3 & 2 & 1 \\ 8 & 4 & 100 & 100 & 100 & 100 \\ 7 & 3 & 100 & 100 & 100 & 100 \\ 6 & 2 & 100 & 100 & 100 & 100 \\ 5 & 1 & 100 & 100 & 100 & 100 \end{bmatrix}$$

Time: 5 mins

Special Matrices (cont)

If the matrix is in arithmetic progress such as

$$\begin{bmatrix} 99 & 92 & 85 & 78 & 71 \\ 64 & 57 & 50 & 43 & 36 \\ 29 & 22 & 15 & 8 & 1 \\ -6 & -13 & -20 & -27 & -34 \end{bmatrix}$$

Write down two different single-line commands which gives us the above matrix.

Time: 5 mins (10 mins break + Q&A)

Outline

Special Matrix Indexing

Solving block matrix problems

$$A\mathbf{x} = \mathbf{b}.$$

where

```
A = np.array([
   [2, -1,  0, 0, 2, 3, 1, 3],
   [-1, 2, -1, 0, 5, 6, 4, 3],
   [0, -1, 2, -1, 0, 5, 3, 4],
   [0,  0, -1, 2, 7, 3, 5, 4],
   [2,  5,  0, 7, 0, 0, 0, 0],
   [3,  6,  5, 3, 0, 0, 0, 0],
   [1,  4,  3, 5, 0, 0, 0, 0],
   [3,  3,  4, 4, 0, 0, 0, 0]])
b = np.array([5, 4, 7, 8, 2, 3, 9, 7])
```

Special Matrix Indexing (cont)

When a matrix is very large (in real-world situation, a matrix can be more than 10000×10000), it is necessary to 'identify' patterns and one visualisation tool is

```
import matplotlib.pylab as plt
plt.spy(A)
plt.show()  # This is needed for Python prompt
```

We can see a block structure in the matrix A.

We can then use 'indexing' to extract the blocks to reduce the dimension of the linear algebra problem to

$$A_{11}\mathbf{x}_1 + A_{12}\mathbf{x}_2 = \mathbf{b}_1$$
$$A_{21}\mathbf{x}_1 = \mathbf{b}_2$$

Special Matrix Indexing (cont)

Try writing down the commands to extract the matrices A_{11}, A_{12} A_{21}, $\mathbf{b}_1$ and $\mathbf{b}_2$ and find the solutions of $\mathbf{x}_1$ and $\mathbf{x}_2$.

Compare the results if we solve the problem $A\mathbf{x} = \mathbf{b}$ directly.

Time: 20 mins

Array Indexing (cont)

In Calculus and Numerical Methods, we need to create the following 1-D array

$$a, a + h, a + 2h, ..., b$$

where $h = \frac{b-a}{n}$.
Command: `xarr = np.linspace(a, b, n+1)`
Note that if we cut an interval to n pieces, there will be $n + 1$ boundary points: E.g. $[0, 1]$ cut into 10 pieces, we have 11 points:

$$0,\ 0.1,\ 0.2,\ 0.3,\ 0.4,\ 0.5,\ 0.6,\ 0.7,\ 0.8,\ 0.9,\ 1.$$

Array Indexing (cont)

We learn that

$$\int_0^{\pi} (\sin x + 0.1 \sin 24x) dx = 2$$

from Calculus and we are told that the '2' indicates the 'area' of the sine function over the interval 0 to π.

To verify this, we will do some drawing, i.e. cutting $[0, \pi]$ into many (e.g. 20) pieces and estimate the trapezoidal area to check if the area approximates 2.

```
xpts = np.linspace(0, np.pi, 20+1)
ypts = np.sin(xpts) + 0.1*np.sin(24*xpts)
# If you don't understand what is going on, print(ypts)
# and compare with the mathematical formula
```

Array Indexing (cont)

Comparing the 'smooth' curve with the 'points' from the 'trapezoid'.

```
x = np.linspace(0, np.pi, 1000+1)  # More points => smoother
y = np.sin(x) + 0.1*np.sin(24*x)
plt.plot(x,y, xpts,ypts,'-*')
plt.show()   # It is not the trapezoid we want
```

How can we draw a trapezoid? The ponts we have are

$$(x_0, y_0), (x_1, y_1), (x_2, y_2), \ldots$$

But we need

$$(x_0, 0), (x_0, y_0), (x_1, y_1), (x_1, 0),$$
$$(x_1, 0), (x_1, y_1), (x_2, y_2), (x_2, 0),$$
$$\ldots$$

Array Indexing (cont)

Can you think of how to get the points in that order?
Lecturer's guess:

```
N = xpts.shape[0]
trapx = np.vstack((xpts[0:(N-1)],xpts[0:(N-1)],xpts[1:N],xpts[1:N]))
trapy = np.vstack((np.zeros(N-1),ypts[0:(N-1)],ypts[1:N],np.zeros(N-1)))
plt.plot(x,y,trapx.T.ravel(),trapy.T.ravel(),'-*')
```

What is the total area of all the trapezoids?
Using SPM formula "Area of Trapezoids = (base1+base2)/2*height", we have

$$T = 0.5*(y_0+y_1)h+0.5*(y_1+y_2)h+...+0.5*(y_{n-1}+y_n)h$$

where $n = 20$, $h = (\pi - 0)/n$ (we cut $[0, \pi]$ into $n = 20$ pieces)

Creating 1-D Array (cont)

Do you know how to write down the command for getting the area???

The answer is 1.995886

You can try getting it using

- Array arithmetic: $0.5h(y_0 + 2y_1 + \cdots + 2y_{n-1} + y_n)$
- np.trapz

Time: 30 mins (10 mins break + Q&A)

Outline

Array Arithmetic

Use Numpy array arithmetic to solve the following SPM questions:

- Write 10001001_2 in base 10
- Express 1056_8 as a number in base 10
- Express 324_5 as a number in base 10

Hint: Recall

$$\begin{aligned} 10001001_2 &= 2^7 + 2^3 + 1 \\ &= \text{np.array([1,0,0,0,1,0,0,1])@...} \end{aligned}$$

Time: 10 mins

Array Arithmetic (cont)

Use Numpy array arithmetic to solve the following SPM question.
Complete the table below for the equation
$y = x^3 - 6x + 7$.

x	-3.5	-3	-2	-1.5	1	0	1	2	3	3.5
y	-14.9		11	12.6	12	7	2		16	28.9

Write down the Python commands to generate the table above and fill in the blanks.

Time: 10 mins

Array Arithmetic (cont)

Creating a Table of Mathematical Functions

- Print a table with x, $\sin x$, $\sin 2x$ and $\cos x$. The variable x ranges from $-\pi$ to π and increases by 0.01.
- Save the table with x, $\sin x$, $\sin 2x$ and $\cos x$ to a text file.
- View the data using a spreadsheet program

Time: 10 mins

Array Arithmetic (cont)

The difference of a sequence $a = [a_0, a_1, a_2, a_3, ...]$ is

$$\Delta a = [a_1 - a_0, a_2 - a_1, a_3 - a_2, ...].$$

Generate a sequence y as follows:

$$y = [f(0), f(0.1), f(0.2), ..., f(6.3)]$$

where f is the cosine function.

- Find Δy.
- Construct a table with $x = 0, 0.1, ..., 6.3$, y, $\Delta y/0.1$ and the derivative $f'(x)$.

Time: 20 mins (10 mins break + Q&A)

Outline

Scripting

Write a Python script which can perform the following:

- Show the information about machine architecture (32/64 bit)
- Show the information about the operating system you are using
- Show the version of the Python you are using
- Show that the purpose of the program is to solve a quadratic equation $ax^2 + bx + c = 0$
- Prompt the user for the coefficients for a, b and c (assuming they are all real numbers). Hint: use input() and convert to float using float()
- Show the (complex value) answer of the quadratic equation using the formula $x1 = \frac{-b+\sqrt{b^2-4ac}}{2a}$ and $x2 = \frac{-b-\sqrt{b^2-4ac}}{2a}$.

Time: 15 mins

Final Exam Oct 2020, Q2(a)

The area of a triangle ABC can be calculated by the formula

$$|\triangle ABC| = \sqrt{s(s-a)(s-b)(s-c)}, \quad s = \frac{a+b+c}{2}$$

when the lengths of the three sides, $a = BC$, $b = AC$, $c = AB$ of the triangle ABC are given. The sine rules and cosine rules of the triangle ABC are given below

$$\frac{a}{\sin A} = \frac{b}{\sin B} = \frac{c}{\sin C}$$

$$a^2 = b^2 + c^2 - 2bc\cos A,$$

$$b^2 = a^2 + c^2 - 2ac\cos B,$$

$$c^2 = a^2 + b^2 - 2ab\cos C.$$

Final Exam Oct 2020, Q2(a) cont

Sample Answer

```
b = 100.0
c = 75.0
A = 35
from math import *
radA = radians(A)
a = sqrt(b**2+c**2-2*b*c*cos(radians(A)))
print("Length a = {a:8.4f} cm".format(a=a))
s = (a+b+c)/2
Area = sqrt(s*(s-a)*(s-b)*(s-c))
print("The area of the triangle with a={a:8.4f},".format(a=a), end="")
print(" b={b:8.4f}, c={c:8.4f} is {S:8.4f}".format(b=b,c=c,S=Area))

# Sine formula returns two values.  The smaller answer is Wrong!!!!!
#B = degrees(asin(sin(radA)*b/a))
B = degrees(acos((a**2+c**2-b**2)/(2*a*c)))
C = degrees(asin(sin(radA)*c/a))
print("Angle B = {B:8.4f} degree".format(B=B))
print("Angle C = {C:8.4f} degree".format(C=C))
```

Final Exam Oct 2020, Q2(a) cont

Marking Guidelines:

- Appropriate imports and definition of variables . . . . . [0.5 mark]
- Appropriate use of formula for calculating the answers. . . . . [1 mark]
- Include the output of the program script [0.5 mark]

```
Length a =  57.7730 cm
The area of the triangle with a= 57.7730, b=100.0000,
c= 75.0000 is 2150.9116
Angle B =  83.1254 degree
Angle C =  48.1254 degree
```

Time: 15 mins

Programming Fundamentals Final Exam May 2021

Given the following data (of integers) representing the exam marks:

```
X = np.array([75,79,88,96,90,80,76,76,79,76,
              75,91,92,82,84,76,97,93,95,82,
              78,81,81,82,77,75,85,63,71,73,
              73,72,68,74,69,67,73,68,62,61,
              68,70,66,61,62,64,62,71,69,66,
              69,61,74,68,60,62,71,73,66,72,
              70,67,69,65,72,61,60,63,61,61,
              66,69,58,56,56,52,55,55,50,55,
              56,54,59,57,54,52,51,58,59,50,
  59,51,55,53,54,54,54,25,25,19,10,42,22,31])
```

Programming Fundamentals Final Exam May 2021 (cont)

And also the following information: Marks

- in the range $[90, 100]$ gets grade A+;
- in the range $[80, 90)$ gets grade A;
- in the range $[70, 80)$ gets grade B+;
- in the range $[60, 70)$ gets grade B;
- in the range $[55, 60)$ gets grade C+;
- in the range $[50, 55)$ gets grade C

Programming Final May 2021 (cont)

You are required to generate the following text report:

```
The final marks:
 75B+     79B+     88A      96A+     90A+     80A      76B+     76B+
 79B+     76B+     75B+     91A+     92A+     82A      84A      76B+
 97A+     93A+     95A+     82A      78B+     81A      81A      82A
 77B+     75B+     85A      63B      71B+     73B+     73B+     72B+
 68B      74B+     69B      67B      73B+     68B      62B      61B
 68B      70B+     66B      61B      62B      64B      62B      71B+
 69B      66B      69B      61B      74B+     68B      60B      62B
 71B+     73B+     66B      72B+     70B+     67B      69B      65B
 72B+     61B      60B      63B      61B      61B      66B      69B
 58C+     56C+     56C+     52C      55C+     55C+     50C      55C+
 56C+     54C      59C+     57C+     54C      52C      51C      58C+
 59C+     50C      59C+     51C      55C+     53C      54C      54C
 54C      25F      25F      19F      10F      42F      22F      31F

Total number of students: 104
```

Programming Final May 2021 (cont)

and also the following text report:

```
Academic Performance Report
-----------------------------------

Grade | No. of Student | Percentage | Histogram (Percentage)
-------------------------------------------------------------------------
 A+   |          7     |    6.73 %  | *******
 A    |          9     |    8.65 %  | *********
 B+   |         25     |   24.04 %  | *************************
 B    |         31     |   29.81 %  | ********************************
 C+   |         13     |   12.50 %  | *************
 C    |         12     |   11.54 %  | ************
 F    |          7     |    6.73 %  | *******
-------------------------------------------------------------------------
Total |       104      | 100.00 %
```

Time: 20 mins (10 Q&A)
Class ends at 6pm.