

Problem Sheet 0

Problem 0.1 Introduction to Python

This problem is designed as an introduction to using Python to solve problems. Please see the **Python Cheat Sheet** on the course page for advice.

(0.1a) Find the eigenvalues of the matrices

$$A = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 2 & 0 \\ 1 & 1 & 2 \\ 1 & 0 & 2 \end{pmatrix}.$$

Solution:

Eigenvalues of A : 0.586, 2, 3.414.

Eigenvalues of B : 3.270, $0.365 \pm 0.692i$.

Listing 0.1: eig.py

```
1 import numpy as np
2
3 A = np.array([[ 2, 1, 0], [ 1, 2, 1], [ 0, 1, 2]])
4 B = np.array([[ 1, 2, 0], [ 1, 1, 2], [ 1, 0, 2]])
5 wa, va = np.linalg.eig(A)
6 wb, vb = np.linalg.eig(B)
7
8 print("Eigenvalues of A: " + str(wa[0]) + ", " + str(wa[1]) +
9       ", " + str(wa[2]))
9 print("Eigenvalues of B: " + str(wb[0]) + ", " + str(wb[1]) +
10       ", " + str(wb[2]))
```

(0.1b) Plot, on the same axes, $y_1(x) = x^2$ and $y_2(x) = \cos(x)$ for $-2 \leq x \leq 2$.

Solution:

Listing 0.2: plotting.py

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 t0, t1 = -2, 2 # start and end
```

```

5 t = np.linspace(t0, t1, 100)          # time mesh
6 y1 = t**2                             # define y1
7 y2 = np.cos(t)                        # define y2
8 plt.plot(t, y1, label="y_1")
9 plt.plot(t, y2, label="y_2")
10 plt.legend(loc="upper left")
11 plt.show()

```

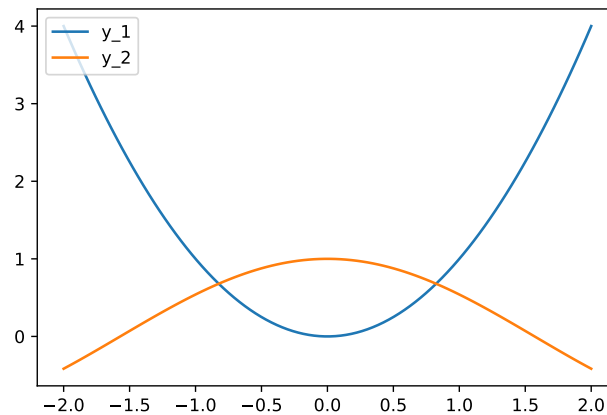


Figure 0.1: Plot of y_1 and y_2

(0.1c) Write a function `Add` that has two input variables `a` and `b` and returns `a+b`. Test your function by computing `9 + 7`.

Solution:

Listing 0.3: `adding.py`

```

1 def Add(a, b):
2     return a + b
3
4 a = Add(9, 7)
5 print(a)

```

(0.1d) Use a `for` loop to calculate

$$\sum_{n=1}^{100} n.$$

Can you write your code so that it uses your `Add` function from (0.1c)?

Solution: Answer: 5050.

Listing 0.4: `sumofintegers.py`

```

1 from adding import Add
2

```

```

3 N = 100          # number of integers to sum
4 x = 0
5 for n in range(1,N+1):
6     x = Add(x, n)
7
8 print(x)

```

(0.1e) Write a function that takes an array, *e.g.* `a=[1,3,4,4,2,8,5,6,3,5,2]`, and prints all the elements that are less than four.

Can you edit your code so that it prints each element only once?

Solution:

Listing 0.5: arraychecker.py

```

1 import numpy as np
2
3 def fourcheck(a):
4     for n in range(0,len(a)):
5         if a[n] < 4:
6             print(a[n])
7
8 # An example, to test the code
9 fourcheck(np.array([1,3,4,4,2,8,5,6,3,5,2]))
10
11 # Now, edit the code to only print each entry once
12 def fourcheck2(a):
13     x = []          # create array to store output
14     for n in range(0,len(a)):
15         if a[n] < 4 and np.logical_not(np.isin(a[n],x)):
16             x.append(a[n])
17     print(x)
18
19 # An example, to test the code
20 fourcheck2(np.array([1,3,4,4,2,8,5,6,3,5,2]))

```

To be solved during the class on 24 February 2022.

Last modified on February 17, 2022