

## レポート用紙

講義名 : 数値解析 1	年月日 : 2024 年 6 月 10 日(月)
学籍番号 : 99999999	氏名 : 幸谷 智紀

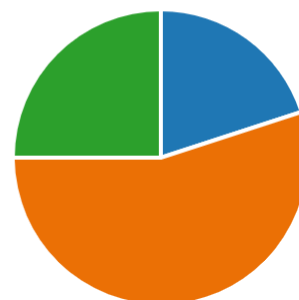
本日の課題 P. 問題 4.5, 問題 4.6, 問題 4.7

感想

## 2. 課題の難易度はどうでしたか？

[詳細](#)

● 難しかった	4
● 少し難しかった	11
● ちょうど良かった	5
● 簡単だった	0
● とても簡単だった	0



### 課題 1 P.30 問題 4.5

```
# first_complex_vector.py: 最初の複素ベクトル演算
import numpy as np # NumPy

# ベクトル
vec_a = np.array([complex(1, 1), 2 + 2 * 1j, 3 + 3 * 1j])
vec_b = np.array([complex(-3, -3), complex(-2, -2), complex(-1, -1)])

print('vec_a = ', vec_a)
print('vec_b = ', vec_b)

# ベクトル演算ができる
vec_c = np.sqrt(2) * vec_a - np.sqrt(3) * vec_b
print('vec_c = ', vec_c)
```

実行結果 :

```
vec_a = [1.+1.j 2.+2.j 3.+3.j]
vec_b = [-3.-3.j -2.-2.j -1.-1.j]
vec_c = [6.61036599+6.61036599j 6.29252874+6.29252874j 5.97469149+5.97469149j]
```

### 課題 2 P.32 問題 4.5

## レポート用紙

```
# first_complex_matrix.py: 最初の複素行列演算
import numpy as np # NumPy

# 複素行列: 3 + 3i -> 3 + 3j, 3 + 3 * 1j or complex(3, 3)
mat_a = np.array([
    [1 + 1j, 2 + 2 * 1j, 3 + 3 * 1j],
    [2 + 2j, 3 + 3 * 1j, 4 + 4 * 1j],
    [3 + 3j, 4 + 4 * 1j, 5 + 5 * 1j]
])

mat_b = np.array([
    [-3 - 3j, -2 - 2j, -1 - 1j],
    [-2 - 2j, -1 - 1j, 0],
    [-1 - 1j, 0, 1 + 1j]
])

print('mat_a = %n', mat_a)
print('mat_b = %n', mat_b)

# 行列演算ができる
mat_c = (2**(1/2) + 1j) * mat_a - (np.sqrt(3) * 1j) * mat_b
print('mat_c = %n', mat_c)
```

実行結果:

```
mat_a =
[[1.+1.j 2.+2.j 3.+3.j]
 [2.+2.j 3.+3.j 4.+4.j]
 [3.+3.j 4.+4.j 5.+5.j]]
mat_b =
[[-3.-3.j -2.-2.j -1.-1.j]
 [-2.-2.j -1.-1.j 0.+0.j]
 [-1.-1.j 0.+0.j 1.+1.j]]
mat_c =
[[-4.78193886 +7.61036599j -2.63567449 +8.29252874j
 -0.48941012 +8.97469149j]
 [-2.63567449 +8.29252874j -0.48941012 +8.97469149j
 1.65685425 +9.65685425j]
 [-0.48941012 +8.97469149j 1.65685425 +9.65685425j
 3.80311862+10.339017j  ]]
```

課題3 問題4.7

# 問題4.7

## レポート用紙

```
import numpy as np # NumPy

# 問題 4.7 (1)
vec_v = np.array([3 + 2j, -4 - 5j])
print('||vec_v||_2          = ', np.linalg.norm(vec_v))
print('sqrt(vec_v, vec_v) = ', np.sqrt(np.conj(vec_v).dot(vec_v)))

# 問題 4.7 (2)
vec_w = np.array([-2 + 1j, -2 - 7j])
print('||vec_v + vec_w||_2      = ', np.linalg.norm(vec_v + vec_w))
print('||vec_v||_2 + ||vec_w||_2 = ', np.linalg.norm(vec_v) +
      np.linalg.norm(vec_w))
if np.linalg.norm(vec_v + vec_w) <= (np.linalg.norm(vec_v) +
    np.linalg.norm(vec_w)):
    print('||vec_v + vec_w||_2 <= ||vec_v||_2 + ||vec_w||_2 が成立')
else:
    print('||vec_v + vec_w||_2 > ||vec_v||_2 + ||vec_w||_2 が成立(あり得ない)')

mat_a = np.array([
    [-1 + 2j, -3 - 3j],
    [-5 - 4j, -2 - 7j]
])
mat_b = np.array([
    [-7j, -5j],
    [2, 3]
])
print('||mat_a + mat_b||_2      = ', np.linalg.norm(mat_a + mat_b))
print('||mat_a||_2 + ||mat_b||_2 = ', np.linalg.norm(mat_a) +
      np.linalg.norm(mat_b))
if np.linalg.norm(mat_a + mat_b) <= (np.linalg.norm(mat_a) +
    np.linalg.norm(mat_b)):
    print('||mat_a + mat_b||_2 <= ||mat_a||_2 + ||mat_b||_2 が成立')
else:
    print('||mat_a + mat_b||_2 <= ||mat_a||_2 + ||mat_b||_2 が成立(あり得ない)')
```

実行結果：

## レポート用紙

```
||vec_v||_2          = 7.3484692283495345
sqrt(vec_v, vec_v) = (7.3484692283495345+0j)
||vec_v + vec_w||_2  = 13.784048752090222
||vec_v||_2 + ||vec_w||_2 = 14.964242334213443
||vec_v + vec_w||_2 <= ||vec_v||_2 + ||vec_w||_2が 成立
||mat_a + mat_b||_2  = 13.19090595827292
||mat_a||_2 + ||mat_b||_2 = 20.144032879480783
||mat_a + mat_b||_2 <= ||mat_a||_2 + ||mat_b||_2が 成立
```

以上