

# Linear Algebra 01

## Table of contents

<b>Series 3</b>	<b>1</b>
Exercise 3.13 . . . . .	1
a. . . . .	1
b. . . . .	3
c. . . . .	3
d. . . . .	3
e. . . . .	4
f. . . . .	4
<b>Series 5</b>	<b>5</b>
Exercise 5.2 . . . . .	5
a . . . . .	5

## Series 3

### Exercise 3.13

```
import numpy as np
```

a.

```
M = np.array([[1, 7], [4, 2]])
M
```

```
array([[1, 7],
       [4, 2]])
```

```
N = np.diag([6, 5])
N
```

```
array([[6, 0],
       [0, 5]])
```

```
P = np.full(shape=[2, 2], fill_value=-np.pi)
P
```

```
array([[-3.14159265, -3.14159265],
       [-3.14159265, -3.14159265]])
```

```
Q = np.e * np.eye(1) + np.sqrt(2) * np.eye(2)
Q
```

```
array([[4.13249539, 2.71828183],
       [2.71828183, 4.13249539]])
```

```
R = np.block([[M, N], [P, Q]])
R
```

```
array([[ 1.          ,  7.          ,  6.          ,  0.          ],
       [ 4.          ,  2.          ,  0.          ,  5.          ],
       [-3.14159265, -3.14159265,  4.13249539,  2.71828183],
       [-3.14159265, -3.14159265,  2.71828183,  4.13249539]])
```

```
Null = np.zeros([2,2])
Null
```

```
S = np.block([[M+N, Null], [Null, P/np.pi]])
S
```

```
array([[ 7.,  7.,  0.,  0.],
       [ 4.,  7.,  0.,  0.],
       [ 0.,  0., -1., -1.],
       [ 0.,  0., -1., -1.]])
```

**b.**

```
np.shape(R)
```

(4, 4)

**c.**

```
RS = R + S  
RS
```

```
array([[ 8.          , 14.          ,  6.          ,  0.          ],  
       [ 8.          ,  9.          ,  0.          ,  5.          ],  
       [-3.14159265, -3.14159265,  3.13249539,  1.71828183],  
       [-3.14159265, -3.14159265,  1.71828183,  3.13249539]])
```

```
R_t = np.transpose(R)  
R_t
```

```
array([[ 1.          ,  4.          , -3.14159265, -3.14159265],  
       [ 7.          ,  2.          , -3.14159265, -3.14159265],  
       [ 6.          ,  0.          ,  4.13249539,  2.71828183],  
       [ 0.          ,  5.          ,  2.71828183,  4.13249539]])
```

```
MN = M @ N  
MN
```

```
array([[ 6, 35],  
       [24, 10]])
```

```
NM = N @ M  
NM
```

```
array([[ 6, 42],  
       [20, 10]])
```

**d.**

```
MM = M + M
MM
```

```
array([[ 2, 14],
       [ 8,  4]])
```

```
NN = np.power(N, N)
NN
```

```
array([[46656,    1],
       [    1, 3125]])
```

**e.**

```
S_2 = S[:, 1:4]
S_2
```

```
array([[ 7.,  0.,  0.],
       [ 7.,  0.,  0.],
       [ 0., -1., -1.],
       [ 0., -1., -1.]])
```

```
R_13 = R[:, [0, 2]]
R_13
```

```
array([[ 1.          ,  6.          ],
       [ 4.          ,  0.          ],
       [-3.14159265,  4.13249539],
       [-3.14159265,  2.71828183]])
```

**f.**

```
R[:, [1, 3]] = R[:, [3, 1]]
R
```

```
array([[ 1.          ,  0.          ,  6.          ,  7.          ],
       [ 4.          ,  5.          ,  0.          ,  2.          ],
       [-3.14159265,  2.71828183,  4.13249539, -3.14159265],
       [-3.14159265,  4.13249539,  2.71828183, -3.14159265]])
```

```
R[:, 0] = 2 * R[:, 0]
R
```

```
array([[ 2.          ,  0.          ,  6.          ,  7.          ],
       [ 8.          ,  5.          ,  0.          ,  2.          ],
       [-6.28318531,  2.71828183,  4.13249539, -3.14159265],
       [-6.28318531,  4.13249539,  2.71828183, -3.14159265]])
```

```
R[2, :] = R[0, :] + R[1, :]
R
```

```
array([[ 2.          ,  0.          ,  6.          ,  7.          ],
       [ 8.          ,  5.          ,  0.          ,  2.          ],
       [10.          ,  5.          ,  6.          ,  9.          ],
       [-6.28318531,  4.13249539,  2.71828183, -3.14159265]])
```

## Series 5

### Exercise 5.2

**a**

```
import numpy as np

A = np.array(
    [
        [24, 68, 0, -36],
        [0, 34, 82, -79],
        [5, 76, -33, 0],
        [-2, 0, -63, 65]
    ]
)

b = np.array(
```

```
[
    [16],
    [-2],
    [58],
    [69]
]
)

res = np.linalg.solve(A, b)
print(res)
```

```
[[1.]
 [2.]
 [3.]
 [4.]]
```