HW1 Report

September 4, 2023

1 HW0 Report

1.1 Task 1

[2]:
```
! conda info
```

```

     active environment : None
       user config file : /home/zealer/.condarc
 populated config files :
          conda version : 23.7.3
    conda-build version : not installed
         python version : 3.11.4.final.0
       virtual packages : __archspec=1=x86_64
                          __cuda=12.2=0
                          __glibc=2.35=0
                          __linux=5.15.90.1=0
                          __unix=0=0
       base environment : /home/zealer/miniconda3  (writable)
      conda av data dir : /home/zealer/miniconda3/etc/conda
  conda av metadata url : None
           channel URLs : https://repo.anaconda.com/pkgs/main/linux-64
                          https://repo.anaconda.com/pkgs/main/noarch
                          https://repo.anaconda.com/pkgs/r/linux-64
                          https://repo.anaconda.com/pkgs/r/noarch
          package cache : /home/zealer/miniconda3/pkgs
                          /home/zealer/.conda/pkgs
       envs directories : /home/zealer/miniconda3/envs
                          /home/zealer/.conda/envs
               platform : linux-64
             user-agent : conda/23.7.3 requests/2.29.0 CPython/3.11.4
Linux/5.15.90.1-microsoft-standard-WSL2 ubuntu/22.04.1 glibc/2.35
                UID:GID : 1000:1000
             netrc file : None
           offline mode : False
```

1.2 Task 2

[3]:
```python
import numpy as np
import scipy.linalg
```

[4]:
```python
a = np.random.randint(1, 5, (5, 5))
```

[5]:
```python
np.ndim(a)
```

[5]:
```
2
```

[6]:
```python
a.size
```

[6]:
```
25
```

[7]:
```python
a.shape
```

[7]:
```
(5, 5)
```

[8]:
```python
a.shape[1]
```

[8]:
```
5
```

[9]:
```python
np.array([[1.,2.,3.], [4.,5.,6.]])
```

[9]:
```
array([[1., 2., 3.],
       [4., 5., 6.]])
```

[10]:
```python
b = np.random.randint(1, 5, (5, 5))
c = np.random.randint(1, 5, (5, 5))
d = np.random.randint(1, 5, (5, 5))
np.block([[a,b], [c,d]])
```

[10]:
```
array([[4, 2, 3, 1, 3, 3, 3, 3, 3, 2],
       [3, 1, 2, 1, 1, 3, 1, 4, 1, 2],
       [4, 3, 2, 4, 1, 2, 2, 2, 2, 4],
       [4, 1, 3, 3, 3, 1, 1, 2, 4, 1],
       [1, 2, 1, 3, 1, 1, 1, 3, 4, 3],
       [3, 3, 1, 3, 4, 3, 4, 2, 2, 2],
       [2, 2, 4, 4, 4, 4, 2, 4, 1, 3],
       [2, 1, 3, 4, 3, 2, 1, 3, 4, 3],
       [2, 4, 2, 2, 2, 2, 2, 3, 1, 2],
       [3, 2, 2, 1, 4, 2, 4, 2, 4, 1]])
```

[11]:
```python
a[-1]
```

[11]:
```
array([1, 2, 1, 3, 1])
```

[12]:
```python
a[1,4]
```

[12]:
```
1
```

[13]:
```python
a[1]
```

[13]:
```
array([3, 1, 2, 1, 1])
```

[14]:

```
print(a[0:5])
print(a[:5])
print(a[0:5,:])
```

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

[15]:

```
a[-5:]
```

[15]:

```
array([[4, 2, 3, 1, 3],
       [3, 1, 2, 1, 1],
       [4, 3, 2, 4, 1],
       [4, 1, 3, 3, 3],
       [1, 2, 1, 3, 1]])
```

[16]:

```
a[0:3,4:9]
```

[16]:

```
array([[3],
       [1],
       [1]])
```

[17]:

```
a[np.ix_([1,3],[0,2])]
```

[17]:

```
array([[3, 2],
       [4, 3]])
```

[18]:

```
a[ 2:5:2,:]
```

[18]:

```
array([[4, 3, 2, 4, 1],
       [1, 2, 1, 3, 1]])
```

[19]:

```
a[ ::2,:]
```

[19]:

```
array([[4, 2, 3, 1, 3],
       [4, 3, 2, 4, 1],
       [1, 2, 1, 3, 1]])
```

[20]:

```
a[ ::-1,:]
```

[20]:
```
array([[1, 2, 1, 3, 1],
       [4, 1, 3, 3, 3],
       [4, 3, 2, 4, 1],
       [3, 1, 2, 1, 1],
       [4, 2, 3, 1, 3]])
```

[21]:
```
a[:len(a),0]
```

[21]:
```
array([4, 3, 4, 4, 1])
```

[22]:
```
a[np.r_[:len(a),0]]
```

[22]:
```
array([[4, 2, 3, 1, 3],
       [3, 1, 2, 1, 1],
       [4, 3, 2, 4, 1],
       [4, 1, 3, 3, 3],
       [1, 2, 1, 3, 1],
       [4, 2, 3, 1, 3]])
```

[23]:
```
a.transpose()
```

[23]:
```
array([[4, 3, 4, 4, 1],
       [2, 1, 3, 1, 2],
       [3, 2, 2, 3, 1],
       [1, 1, 4, 3, 3],
       [3, 1, 1, 3, 1]])
```

[24]:
```
a.conj().transpose()
```

[24]:
```
array([[4, 3, 4, 4, 1],
       [2, 1, 3, 1, 2],
       [3, 2, 2, 3, 1],
       [1, 1, 4, 3, 3],
       [3, 1, 1, 3, 1]])
```

[25]:
```
a @ b
```

[25]:
```
array([[28, 24, 37, 36, 34],
       [18, 16, 22, 22, 20],
       [30, 24, 39, 39, 29],
       [27, 25, 37, 43, 34],
       [15, 11, 22, 23, 16]])
```

[26]:
```
a * b
```

[26]:
```
array([[12,  6,  9,  3,  6],
       [ 9,  1,  8,  1,  2],
       [ 8,  6,  4,  8,  4],
       [ 4,  1,  6, 12,  3],
       [ 1,  2,  3, 12,  3]])
```

[27]:
```
a / b
```

[27]:
```
array([[1.33333333, 0.66666667, 1.        , 0.33333333, 1.5       ],
       [1.        , 1.        , 0.5       , 1.        , 0.5       ],
       [2.        , 1.5       , 1.        , 2.        , 0.25      ],
       [4.        , 1.        , 1.5       , 0.75      , 3.        ],
       [1.        , 2.        , 0.33333333, 0.75      , 0.33333333]])
```

[28]:
```
a ** 3
```

[28]:
```
array([[64,  8, 27,  1, 27],
       [27,  1,  8,  1,  1],
       [64, 27,  8, 64,  1],
       [64,  1, 27, 27, 27],
       [ 1,  8,  1, 27,  1]])
```

[29]:
```
a > 0.5
```

[29]:
```
array([[ True,  True,  True,  True,  True],
       [ True,  True,  True,  True,  True],
       [ True,  True,  True,  True,  True],
       [ True,  True,  True,  True,  True],
       [ True,  True,  True,  True,  True]])
```

[30]:
```
np.nonzero(a>0.5)
```

[30]:
```
(array([0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 4, 4,
        4, 4, 4]),
 array([0, 1, 2, 3, 4, 0, 1, 2, 3, 4, 0, 1, 2, 3, 4, 0, 1, 2, 3, 4, 0, 1,
        2, 3, 4]))
```

[31]:
```
v = np.random.randint(5, size=(2, 4))
a[:,np.nonzero(v>0.5)[0]]
```

[31]:
```
array([[4, 4, 4, 2, 2],
       [3, 3, 3, 1, 1],
       [4, 4, 4, 3, 3],
       [4, 4, 4, 1, 1],
       [1, 1, 1, 2, 2]])
```

[109]:
```
a[a<0.5]=0
print(a)
```

```
[[3 3 3 3 3]
 [3 3 3 3 3]
 [3 3 3 3 3]
 [3 3 3 3 3]
 [3 3 3 3 3]]
```

[110]:
```
a * (a>0.5)
```

[110]:
```
array([[3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3]])
```

[111]:

```
a[:] = 3
print(a)
```

```
[[3 3 3 3 3]
 [3 3 3 3 3]
 [3 3 3 3 3]
 [3 3 3 3 3]
 [3 3 3 3 3]]
```

[112]:

```
y = a.copy()
print(y)
```

```
[[3 3 3 3 3]
 [3 3 3 3 3]
 [3 3 3 3 3]
 [3 3 3 3 3]
 [3 3 3 3 3]]
```

[113]:

```
y = a[1,:].copy()
print(y)
```

```
[3 3 3 3 3]
```

[114]:

```
y = a.flatten()
print(y)
```

```
[3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3]
```

[40]:

```
np.arange(1.,11.)
```

[40]:

```
array([ 1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.])
```

[41]:

```
np.arange(10.)
```

[41]:

```
array([0., 1., 2., 3., 4., 5., 6., 7., 8., 9.])
```

[42]:

```
np.arange(1.,11.)[:]
```

[42]:

```
array([ 1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.])
```

[43]:

```
np.zeros((3,4))
```

[43]:

```
array([[0., 0., 0., 0.],
       [0., 0., 0., 0.],
       [0., 0., 0., 0.]])
```

[44]:

```
np.ones((3,4))
```

```
[44]: array([[1., 1., 1., 1.],
             [1., 1., 1., 1.],
             [1., 1., 1., 1.]])
```

```
[45]: np.eye(3)
```

```
[45]: array([[1., 0., 0.],
             [0., 1., 0.],
             [0., 0., 1.]])
```

```
[46]: np.diag(a)
```

```
[46]: array([3, 3, 3, 3, 3])
```

```
[117]: np.diag(v, 0)
```

```
[117]: array([0, 1])
```

```
[48]: np.random.rand(3,4)
```

```
[48]: array([[0.79781805, 0.31183134, 0.96870775, 0.81317329],
             [0.82896719, 0.18729088, 0.38960244, 0.819809  ],
             [0.60032037, 0.36841622, 0.84258099, 0.17148099]])
```

```
[49]: np.linspace(1,3,4)
```

```
[49]: array([1.        , 1.66666667, 2.33333333, 3.        ])
```

```
[50]: np.mgrid[0:9.,0:6.]
```

```
[50]: array([[[0., 0., 0., 0., 0., 0.],
              [1., 1., 1., 1., 1., 1.],
              [2., 2., 2., 2., 2., 2.],
              [3., 3., 3., 3., 3., 3.],
              [4., 4., 4., 4., 4., 4.],
              [5., 5., 5., 5., 5., 5.],
              [6., 6., 6., 6., 6., 6.],
              [7., 7., 7., 7., 7., 7.],
              [8., 8., 8., 8., 8., 8.]],

             [[0., 1., 2., 3., 4., 5.],
              [0., 1., 2., 3., 4., 5.],
              [0., 1., 2., 3., 4., 5.],
              [0., 1., 2., 3., 4., 5.],
              [0., 1., 2., 3., 4., 5.],
              [0., 1., 2., 3., 4., 5.],
              [0., 1., 2., 3., 4., 5.],
              [0., 1., 2., 3., 4., 5.],
              [0., 1., 2., 3., 4., 5.]]])
```

```
[51]: np.ogrid[0:9.,0:6.]
```

```
[51]: [array([[0.],
              [1.],
        [2.],
        [3.],
        [4.],
        [5.],
        [6.],
        [7.],
        [8.]]),
 array([[0., 1., 2., 3., 4., 5.]])]
```

[52]:
```
np.meshgrid([1,2,4],[2,4,5])
```

[52]:
```
[array([[1, 2, 4],
        [1, 2, 4],
        [1, 2, 4]]),
 array([[2, 2, 2],
        [4, 4, 4],
        [5, 5, 5]])]
```

[120]:
```
np.tile(a, (10, 10))
```

[120]:
```
array([[3, 3, 3, ..., 3, 3, 3],
       [3, 3, 3, ..., 3, 3, 3],
       [3, 3, 3, ..., 3, 3, 3],
       ...,
       [3, 3, 3, ..., 3, 3, 3],
       [3, 3, 3, ..., 3, 3, 3],
       [3, 3, 3, ..., 3, 3, 3]])
```

[54]:
```
np.concatenate((a,b),1)
np.hstack((a,b))
np.column_stack((a,b))
```

[54]:
```
array([[3, 3, 3, 3, 3, 3, 3, 3, 3, 2],
       [3, 3, 3, 3, 3, 3, 1, 4, 1, 2],
       [3, 3, 3, 3, 3, 2, 2, 2, 2, 4],
       [3, 3, 3, 3, 3, 1, 1, 2, 4, 1],
       [3, 3, 3, 3, 3, 1, 1, 3, 4, 3]])
```

[55]:
```
a.max()
```

[55]:
```
3
```

[56]:
```
a.max(0)
```

[56]:
```
array([3, 3, 3, 3, 3])
```

[57]:
```
np.maximum(a, b)
```

[57]:
```
array([[3, 3, 3, 3, 3],
       [3, 3, 4, 3, 3],
       [3, 3, 3, 3, 4],
       [3, 3, 3, 4, 3],
       [3, 3, 3, 4, 3]])
```

[79]:
```
np.linalg.norm(v)
```

[79]:
```
4.898979485566356
```

[59]:
```
np.logical_and(a,b)
```

[59]:
```
array([[ True,  True,  True,  True,  True],
       [ True,  True,  True,  True,  True],
       [ True,  True,  True,  True,  True],
       [ True,  True,  True,  True,  True],
       [ True,  True,  True,  True,  True]])
```

[121]:
```
np.logical_or(a,b)
```

[121]:
```
array([[ True,  True,  True,  True,  True],
       [ True,  True,  True,  True,  True],
       [ True,  True,  True,  True,  True],
       [ True,  True,  True,  True,  True],
       [ True,  True,  True,  True,  True]])
```

[60]:
```
a & b
```

[60]:
```
array([[3, 3, 3, 3, 2],
       [3, 1, 0, 1, 2],
       [2, 2, 2, 2, 0],
       [1, 1, 2, 0, 1],
       [1, 1, 3, 0, 3]])
```

[61]:
```
a | b
```

[61]:
```
array([[3, 3, 3, 3, 3],
       [3, 3, 7, 3, 3],
       [3, 3, 3, 3, 7],
       [3, 3, 3, 7, 3],
       [3, 3, 3, 7, 3]])
```

[62]:
```
np.linalg.inv(a[1:2, 1:2])
```

[62]:
```
array([[0.33333333]])
```

[63]:
```
np.linalg.pinv(a)
```

[63]:
```
array([[0.01333333, 0.01333333, 0.01333333, 0.01333333, 0.01333333],
       [0.01333333, 0.01333333, 0.01333333, 0.01333333, 0.01333333],
       [0.01333333, 0.01333333, 0.01333333, 0.01333333, 0.01333333],
       [0.01333333, 0.01333333, 0.01333333, 0.01333333, 0.01333333],
       [0.01333333, 0.01333333, 0.01333333, 0.01333333, 0.01333333]])
```

[64]:
```
np.linalg.matrix_rank(a)
```

[64]:
```
1
```

[83]:
```
np.linalg.lstsq(a,a, rcond=None)
```

[83]:

```
(array([[0.2, 0.2, 0.2, 0.2, 0.2],
        [0.2, 0.2, 0.2, 0.2, 0.2],
        [0.2, 0.2, 0.2, 0.2, 0.2],
        [0.2, 0.2, 0.2, 0.2, 0.2],
        [0.2, 0.2, 0.2, 0.2, 0.2]]),
 array([], dtype=float64),
 1,
 array([1.50000000e+01, 1.58882186e-15, 0.00000000e+00, 0.00000000e+00,
        0.00000000e+00]))
```

[108]:

```
U, S, Vh = np.linalg.svd(np.random.randint(1, 9, (10, 10, 10)))
V = Vh.T
```

[96]:

```
np.linalg.cholesky(np.random.randint(1, 2, (1, 1))).T
```

[96]:

```
array([[1.]])
```

[97]:

```
D,V = np.linalg.eig(a)
```

[98]:

```
D,V = scipy.linalg.eig(a,b)
```

[99]:

```
Q,R = scipy.linalg.qr(a)
```

[100]:

```
LU,P = scipy.linalg.lu_factor(b)
```

[101]:

```
np.fft.fft(a)
```

[101]:

```
array([[15.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j],
       [15.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j],
       [15.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j],
       [15.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j],
       [15.+0.j,  0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j]])
```

[102]:

```
np.fft.ifft(a)
```

[102]:

```
array([[3.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j],
       [3.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j],
       [3.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j],
       [3.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j],
       [3.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j]])
```

[103]:

```
np.sort(a)
```

[103]:

```
array([[3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3]])
```

[122]:

```
np.sort(a, axis=1)
```

[122]:

```
array([[3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3]])
```

[104]:
```
I = np.argsort(a[:,1])
```

[105]:
```
np.linalg.lstsq(a,b, rcond=None)
```

[105]:
```
(array([[0.13333333, 0.10666667, 0.18666667, 0.18666667, 0.16      ],
        [0.13333333, 0.10666667, 0.18666667, 0.18666667, 0.16      ],
        [0.13333333, 0.10666667, 0.18666667, 0.18666667, 0.16      ],
        [0.13333333, 0.10666667, 0.18666667, 0.18666667, 0.16      ],
        [0.13333333, 0.10666667, 0.18666667, 0.18666667, 0.16      ]]),
 array([], dtype=float64),
 1,
 array([1.50000000e+01, 1.58882186e-15, 0.00000000e+00, 0.00000000e+00,
        0.00000000e+00]))
```

[106]:
```
np.unique(a)
```

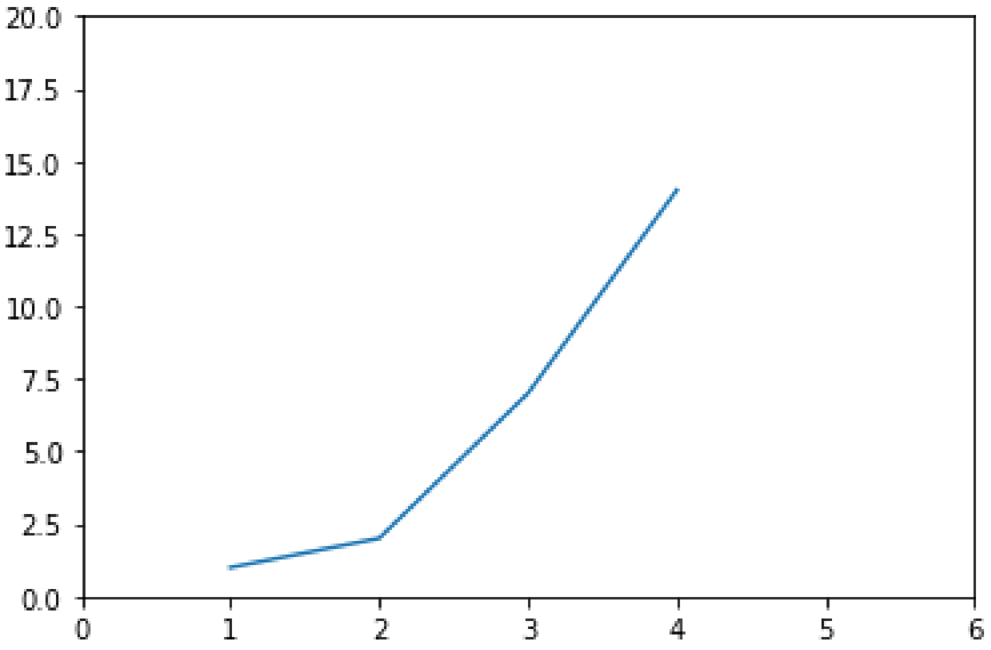
[106]:
```
array([3])
```

[107]:
```
a.squeeze()
```

[107]:
```
array([[3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3]])
```

1.3 Task 3

[124]:
```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4], [1,2,7,14])
plt.axis([0, 6, 0, 20])
plt.show()
```



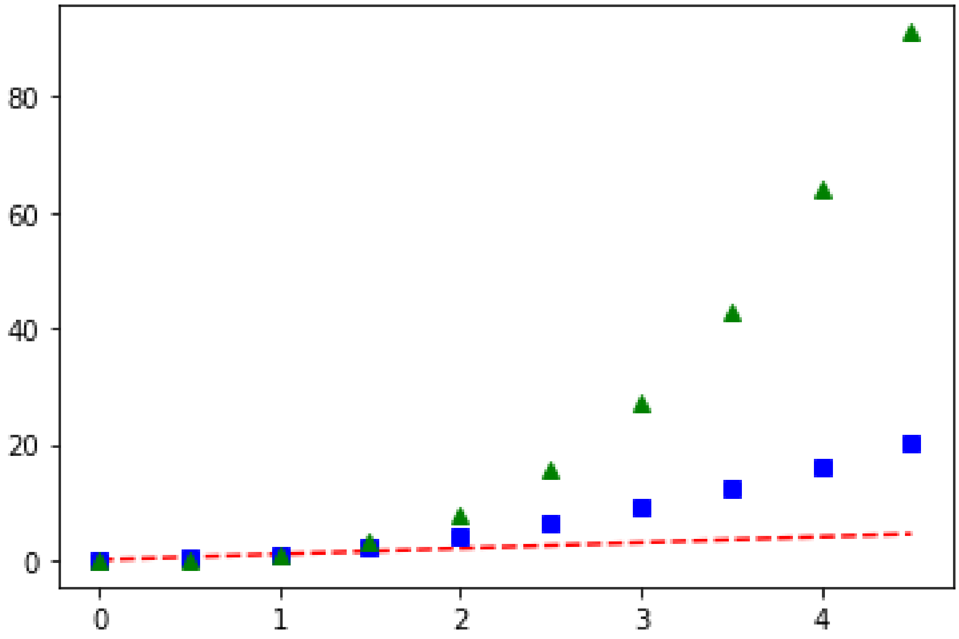
1.4 Task 4

[128]:

```
import numpy as np

t = np.arange(0., 5., 0.5)

plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3, 'g^')
plt.show()
```



1.5 Task 5

Github Acccount name: xiaozealer

1.6 Task 6

https://github.com/xiaozealer/576