TASK2

In [1]:

```
import numpy as np
import scipy.linalg as lina

a = np.array([1,2,3,4,5])
b = np.array([[1.,2.,3.],[4.,5.,6.]])
```

In [2]:

```
np.ndim(a)
```

Out[2]:

```
1
```

In [3]:

```
a.size
```

Out[3]:

```
5
```

In [4]:

```
a.shape
```

Out[4]:

```
(5,)
```

In [5]:

```
a.shape[0]
```

Out[5]:

```
5
```

In [6]:

```
b.shape
```

Out[6]:

```
(2, 3)
```

In [7]:

```
b.shape[0]
```

Out[7]:

```
2
```

In [8]:

```
b.shape[1]
```

Out[8]:

```
3
```

In [9]:

```
c = np.array([1.,2.,3.])
d = np.array([2.,3.,4.])
e = np.array([3.,4.,5.])
f = np.array([4.,5.,6.])
np.block([[c,d],[e,f]])
```

Out[9]:

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

In [10]:

```
c[-1]
```

Out[10]:

```
3.0
```

In [11]:

```
c[1:2]
```

```
Out[11]: array([2.])
```

```
In [12]:  g = np.array([[1.,2.,3.],[4.,5.,6.]])
          g[1,:]

Out[12]: array([4., 5., 6.])
```

```
In [13]:  g[np.ix_([0],[0])]

Out[13]: array([[1.]])
```

```
In [14]:  g[::2,:]

Out[14]: array([[1., 2., 3.]])
```

```
In [15]:  g[::-1,:]

Out[15]: array([[4., 5., 6.],
                [1., 2., 3.]])
```

```
In [16]:  g[np.r_[:len(g),0]]

Out[16]: array([[1., 2., 3.],
                [4., 5., 6.],
                [1., 2., 3.]])
```

```
In [17]:  g.transpose()

Out[17]: array([[1., 4.],
                [2., 5.],
                [3., 6.]])
```

```
In [18]:  g.conj().transpose()

Out[18]: array([[1., 4.],
                [2., 5.],
                [3., 6.]])
```

```
In [19]:  g @ c

Out[19]: array([14., 32.])
```

```
In [20]:  g * c

Out[20]: array([[ 1.,  4.,  9.],
                [ 4., 10., 18.]])
```

```
In [21]:  g / c

Out[21]: array([[1. , 1. , 1. ],
                [4. , 2.5, 2. ]])
```

```
In [22]:  g ** 3
```

Out[22]:

```
array([[  1.,   8.,  27.],
       [ 64., 125., 216.]])
```

In [23]:

```
(g > 4)
```

Out[23]:

```
array([[False, False, False],
       [False,  True,  True]])
```

In [24]:

```
np.nonzero(g > 4)
```

Out[24]:

```
(array([1, 1], dtype=int64), array([1, 2], dtype=int64))
```

In [25]:

```
g[g<0.5] = 0
```

In [26]:

```
g*(g>0.5)
```

Out[26]:

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

In [27]:

```
g[:] =3
```

In [28]:

```
g
```

Out[28]:

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

In [29]:

```
h = g.copy()
```

In [30]:

```
g,h
```

Out[30]:

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

In [31]:

```
h = g[1,:].copy()
h
```

Out[31]:

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

In [32]:

```
h = g.flatten()
h
```

Out[32]:

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

In [33]:

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

Out[33]:

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

In [34]:

```
np.r_[1.,11.]
```

Out[34]:
```
array([ 1., 11.])
```

In [35]:
```
np.r_[:9:10j]
```

Out[35]:
```
array([0., 1., 2., 3., 4., 5., 6., 7., 8., 9.])
```

In [36]:
```
np.arange(1.,11.)[:, np.newaxis]
```

Out[36]:
```
array([[ 1.],
       [ 2.],
       [ 3.],
       [ 4.],
       [ 5.],
       [ 6.],
       [ 7.],
       [ 8.],
       [ 9.],
       [10.]])
```

In [37]:
```
np.zeros((3,4))
```

Out[37]:
```
array([[0., 0., 0., 0.],
       [0., 0., 0., 0.],
       [0., 0., 0., 0.]])
```

In [38]:
```
np.zeros((3,4,5))
```

Out[38]:
```
array([[[0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.]],

       [[0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.]],

       [[0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.]]])
```

In [39]:
```
np.ones((3,4))
```

Out[39]:
```
array([[1., 1., 1., 1.],
       [1., 1., 1., 1.],
       [1., 1., 1., 1.]])
```

In [40]:
```
np.eye(3)
```

Out[40]:
```
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])
```

In [41]:
```
np.diag(g)
```

Out[41]:
```
array([3., 3.])
```

In [42]:

```
np.diag(g,0)
```

Out[42]:

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

In [43]:

```
np.random.rand(3,4)
```

Out[43]:

```
array([[0.3046904 , 0.42363178, 0.44782142, 0.24737927],
       [0.44185316, 0.71664589, 0.63792223, 0.95376616],
       [0.80750063, 0.78858148, 0.43814691, 0.72184037]])
```

In [44]:

```
np.random.random_sample((3,4,5))
```

Out[44]:

```
array([[[0.28145637, 0.7982989 , 0.04958842, 0.92593   , 0.44494309],
        [0.56843217, 0.7446242 , 0.12918298, 0.2116079 , 0.36624408],
        [0.33633955, 0.73674309, 0.051678  , 0.86616078, 0.39255255],
        [0.93787273, 0.96424178, 0.72468754, 0.86594152, 0.54570867]],

       [[0.7093951 , 0.23598997, 0.29766608, 0.99025535, 0.97826562],
        [0.36978306, 0.7424667 , 0.36365427, 0.61264278, 0.03591981],
        [0.26275241, 0.16050864, 0.37053614, 0.8685359 , 0.33639181],
        [0.84963327, 0.61998175, 0.64923596, 0.35117235, 0.98023683]],

       [[0.41805584, 0.42656075, 0.82987279, 0.81636332, 0.23411082],
        [0.801149  , 0.11770049, 0.52521844, 0.17292715, 0.97238982],
        [0.56109557, 0.3610911 , 0.42416395, 0.00435305, 0.92093353],
        [0.60846586, 0.34162096, 0.67806109, 0.96781278, 0.50820537]]])
```

In [45]:

```
np.linspace(1,3,4)
```

Out[45]:

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

In [46]:

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

Out[46]:

```
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.]]])
```

In [47]:

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

Out[47]:

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

In [48]:

```
np.ix_(np.r_[0:9.],np.r_[0:6.])
```

Out[48]:

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

In [49]:

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

Out[49]:

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

In [50]:

```
np.tile(g,(3,4))
```

Out[50]:

```
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.],
       [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.]])
```

In [51]:

```
np.concatenate((c,d),0)
```

Out[51]:

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

In [52]:

```
np.vstack((c,d))
```

Out[52]:

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

In [53]:

```
c.max()
```

Out[53]:

```
3.0
```

In [54]:

```
c.max(0)
```

Out[54]:

```
3.0
```

In [55]:

```
g.max(1)
```

Out[55]:

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

In [56]:

```
np.maximum(c,d)
```

Out[56]:

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

In [57]:

```
np.sqrt(c@c)
```

Out[57]:

```
3.7416573867739413
```

In [58]:

```
np.linalg.norm(c)
```

Out[58]:

```
3.7416573867739413
```

In [59]:

```
np.logical_and(c,d)
```

Out[59]:

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

In [60]:

```
np.logical_or(c,d)
```

Out[60]:

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

In [61]:

```
i = np.array([[1,2,3],[4,5,6],[7,10,9]])
lina.eig(i)
```

Out[61]:

```
(array([16.82540654+0.j, -0.56655283+0.j, -1.2588537 +0.j]),
 array([[-0.22187905, -0.86720234, -0.7307057 ],
        [-0.49983984,  0.47988984, -0.16747068],
        [-0.83721552,  0.13291287,  0.66183288]]))
```

In [62]:

```
lina.inv(i)
```

Out[62]:

```
array([[-1.25      ,  1.        , -0.25      ],
       [ 0.5       , -1.        ,  0.5       ],
       [ 0.41666667,  0.33333333, -0.25      ]])
```

In [63]:

```
lina.pinv(i)
```

Out[63]:

```
array([[-1.25      ,  1.        , -0.25      ],
       [ 0.5       , -1.        ,  0.5       ],
       [ 0.41666667,  0.33333333, -0.25      ]])
```

In [64]:

```
lina.solve(i,c)
```

Out[64]:

```
array([ 1.26882631e-16, -0.00000000e+00,  3.33333333e-01])
```

In [65]:

```
lina.svd(i)
```

Out[65]:

```
(array([[-0.20157473, -0.72595537, -0.65753816],
        [-0.48945851, -0.50683053,  0.70961481],
        [-0.8484091 ,  0.46487806, -0.25316081]]),
 array([17.86008107,  1.32709403,  0.50628624]),
 array([[-0.45342865, -0.63462942, -0.62581783],
        [ 0.3774178 ,  0.4993747 , -0.77985942],
        [ 0.80743932, -0.58980539,  0.01308965]]))
```

In [66]:

```
lina.qr(i)
```

Out[66]:

```
(array([[-0.12309149,  0.69631062, -0.70710678],
        [-0.49236596, -0.66149509, -0.56568542],
        [-0.86164044,  0.27852425,  0.42426407]]),
 array([[ -8.1240384 , -11.32441717, -11.07823419],
        [  0.        ,   0.87038828,   0.62667956],
        [  0.        ,   0.        ,  -1.69705627]]))
```

In [67]:

```
lina.lu(i)
```

Out[67]:

```
(array([[0., 0., 1.],
        [0., 1., 0.],
        [1., 0., 0.]]),
 array([[ 1.        ,  0.        ,  0.        ],
        [ 0.57142857,  1.        ,  0.        ],
        [ 0.14285714, -0.8       ,  1.        ]]),
 array([[ 7.        , 10.        ,  9.        ],
        [ 0.        , -0.71428571,  0.85714286],
        [ 0.        ,  0.        ,  2.4       ]]))
```

In [68]:

```
np.fft.fft(i)
```

Out[68]:

```
array([[ 6. +0.j       , -1.5+0.8660254j, -1.5-0.8660254j],
       [15. +0.j       , -1.5+0.8660254j, -1.5-0.8660254j],
       [26. +0.j       , -2.5-0.8660254j, -2.5+0.8660254j]])
```

In [69]:

```
np.fft.ifft(i)
```

Out[69]:

```
array([[ 2.        +0.j        , -0.5       -0.28867513j,
        -0.5       +0.28867513j],
       [ 5.        +0.j        , -0.5       -0.28867513j,
        -0.5       +0.28867513j],
       [ 8.66666667+0.j        , -0.83333333+0.28867513j,
        -0.83333333-0.28867513j]])
```

In [70]:

```
np.sort(i)
```

Out[70]:

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

In [71]:

```
j = np.array([[1,2,3],[4,5,6],[7,8,9]])
lina.lstsq(i,j)
```

Out[71]:

```
(array([[ 1.00000000e+00,  5.00000000e-01,  2.24258837e-16],
        [ 1.05692096e-15,  4.99600361e-16, -4.44089210e-16],
        [ 1.22191171e-16,  5.00000000e-01,  1.00000000e+00]]),
 array([], dtype=float64),
 3,
 array([17.86008107,  1.32709403,  0.50628624]))
```

In [72]:

```
np.unique(i)
```

Out[72]:

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

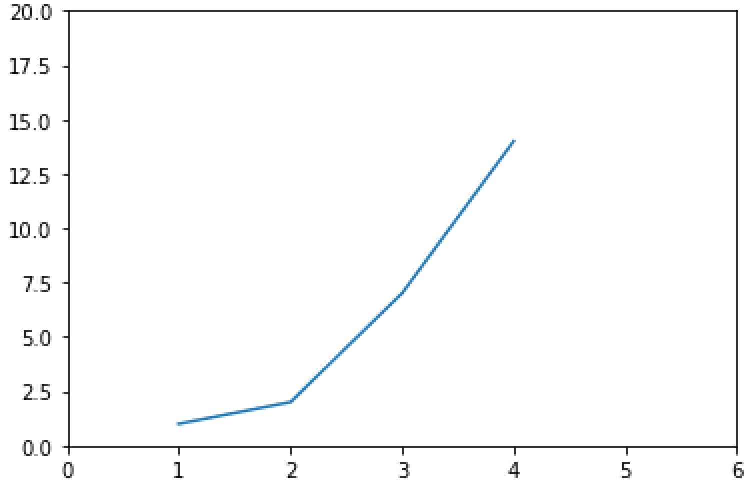
In [73]:

```
i.squeeze()
```

```
array([[ 1,  2,  3],
Out[73]:        [ 4,  5,  6],
                [ 7, 10,  9]])
```

TASK3

```
In [74]: import matplotlib.pyplot as plt
         plt.plot([1,2,3,4], [1,2,7,14])
         plt.axis([0, 6, 0, 20])
         plt.show()
```

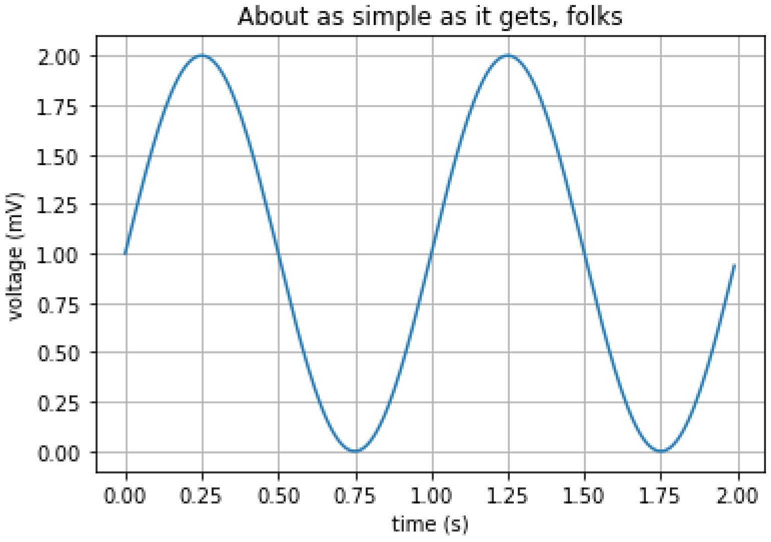


```
In [75]: t = np.arange(0.0, 2.0, 0.01)
         s = 1 + np.sin(2 * np.pi * t)

         fig, ax = plt.subplots()
         ax.plot(t, s)

         ax.set(xlabel='time (s)', ylabel='voltage (mV)',
                title='About as simple as it gets, folks')
         ax.grid()

         fig.savefig("test.png")
         plt.show()
```



```
In [ ]:
```