Report of Assignment 0

Zihao Zhao
(zz53)

September 3, 2018

0.1 Task 1

```
(base) C:\Users\Zihao>conda info

 active environment : base
active env location : C:\Users\Zihao\Anaconda3
       shell level : 1
   user config file : C:\Users\Zihao\.condarc
 populated config files : C:\Users\Zihao\.condarc
      conda version : 4.5.11
conda-build version : 3.13.0
     python version : 3.6.6.final.0
   base environment : C:\Users\Zihao\Anaconda3  (writable)
       channel URLs : https://repo.anaconda.com/pkgs/main/win-64
                      https://repo.anaconda.com/pkgs/main/noarch
                      https://repo.anaconda.com/pkgs/free/win-64
                      https://repo.anaconda.com/pkgs/free/noarch
                      https://repo.anaconda.com/pkgs/r/win-64
                      https://repo.anaconda.com/pkgs/r/noarch
                      https://repo.anaconda.com/pkgs/pro/win-64
                      https://repo.anaconda.com/pkgs/pro/noarch
                      https://repo.anaconda.com/pkgs/msys2/win-64
                      https://repo.anaconda.com/pkgs/msys2/noarch
      package cache : C:\Users\Zihao\Anaconda3\pkgs
                      C:\Users\Zihao\AppData\Local\conda\conda\pkgs
   envs directories : C:\Users\Zihao\Anaconda3\envs
                      C:\Users\Zihao\AppData\Local\conda\conda\envs
                      C:\Users\Zihao\.conda\envs
           platform : win-64
         user-agent : conda/4.5.11 requests/2.19.1 CPython/3.6.6 Windows/10 Windows/10.0.17134
      administrator : False
         netrc file : None
       offline mode : False
```

0.2 Task 2

```
In [1]: import numpy as np
        import scipy.linalg
        import scipy.signal
        import scipy.sparse.linalg
```

```
In [2]: lst = [[1,2,3],[1,3,4]]
        print(lst)
```

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

```
In [3]: a = np.array(lst)
        print(a)
```

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

```
In [4]: a.ndim
```

```
Out[4]: 2
```

```
In [5]: a.size
```

```
Out[5]: 6
```

```
In [6]: a.shape
```

```
Out[6]: (2, 3)
```

```
In [7]: a.shape[1]
```

```
Out[7]: 3
```

```
In [8]: np.array([[1.,2.,3.], [4.,5.,6.]])
```

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

```
In [9]: a = np.array([1,4,2])
        b = np.array([3,4])
        c = np.array([4,2])
        d = np.array([3,3,3])
        e = np.block([[a,b], [c,d]])
        print(e)
```

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

```
In [10]: a[-1]

Out[10]: 2

In [11]: e[1,3]

Out[11]: 3

In [12]: e[1]

Out[12]: array([4, 2, 3, 3, 3])

In [13]: f = np.array([[1,2],[3,4],[5,6]])
         print(f)

[[1 2]
 [3 4]
 [5 6]]


In [14]: f[0:2]

Out[14]: array([[1, 2],
                [3, 4]])

In [15]: f[-2:]

Out[15]: array([[3, 4],
                [5, 6]])

In [16]: f[0:2,0:1]

Out[16]: array([[1],
                [3]])

In [17]: f[np.ix_([0,2],[1])]

Out[17]: array([[2],
                [6]])

In [18]: f[0:3:2,:]

Out[18]: array([[1, 2],
                [5, 6]])

In [19]: f[::2]

Out[19]: array([[1, 2],
                [5, 6]])

In [20]: f[::-2]
```

```
Out[20]: array([[5, 6],
                [1, 2]])

In [21]: f[np.r_[:len(f),0]]

Out[21]: array([[1, 2],
                [3, 4],
                [5, 6],
                [1, 2]])

In [22]: f.T

Out[22]: array([[1, 3, 5],
                [2, 4, 6]])

In [23]: a.conj().T

Out[23]: array([1, 4, 2])

In [24]: a = np.array([[1,1],[3,3]])
         b = np.array([[2,2],[4,4]])

In [25]: a @ b

Out[25]: array([[ 6,  6],
                [18, 18]])

In [26]: a * b

Out[26]: array([[ 2,  2],
                [12, 12]])

In [27]: a / b

Out[27]: array([[0.5 , 0.5 ],
                [0.75, 0.75]])

In [28]: a**3

Out[28]: array([[ 1,  1],
                [27, 27]], dtype=int32)

In [29]: a > 0.5

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

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

Out[30]: (array([0, 0, 1, 1], dtype=int64), array([0, 1, 0, 1], dtype=int64))
```

```
In [31]: v = a[1,1]
         a[:,np.nonzero(v>0.5)[0]]

Out[31]: array([[1],
                [3]])

In [32]: v = a[:,1]
         a[:,v.T>0.5]

Out[32]: array([[1, 1],
                [3, 3]])

In [33]: a[a<0.5] = 0
         print(a)

[[1 1]
 [3 3]]


In [34]: a * (a>0.5)

Out[34]: array([[1, 1],
                [3, 3]])

In [35]: a[:] = 3
         print(a)

[[3 3]
 [3 3]]


In [36]: y = a.copy()
         print(y)

[[3 3]
 [3 3]]


In [37]: y = a[1,:].copy()
         print(y)

[3 3]


In [38]: y = a.flatten()
         print(y)

[3 3 3 3]


In [39]: np.r_[1.:11.]
```

```
Out[39]: array([ 1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.])
```

```
In [40]: np.r_[:10.]
```

```
Out[40]: array([0., 1., 2., 3., 4., 5., 6., 7., 8., 9.])
```

```
In [41]: np.arange(1.,11.)[:, np.newaxis]
```

```
Out[41]: array([[ 1.],
                [ 2.],
                [ 3.],
                [ 4.],
                [ 5.],
                [ 6.],
                [ 7.],
                [ 8.],
                [ 9.],
                [10.]])
```

```
In [42]: np.zeros((3,4,5))
```

```
Out[42]: 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 [43]: np.ones((3,4))
```

```
Out[43]: array([[1., 1., 1., 1.],
                [1., 1., 1., 1.],
                [1., 1., 1., 1.]])
```

```
In [44]: np.eye(3)
```

```
Out[44]: array([[1., 0., 0.],
                [0., 1., 0.],
                [0., 0., 1.]])
```

```
In [45]: a = np.array([[2,3,5],[3,6,8],[3,5,7]])
         np.diag(a)
```

```
Out[45]: array([2, 6, 7])

In [46]: np.diag(a,0)

Out[46]: array([2, 6, 7])

In [47]: np.random.rand(3,4)

Out[47]: array([[0.45416245, 0.86106424, 0.5201348 , 0.57611571],
                [0.25987878, 0.14829017, 0.6542522 , 0.71671522],
                [0.25351193, 0.53502168, 0.48579698, 0.16829893]])

In [48]: np.linspace(1,3,4)

Out[48]: array([1.        , 1.66666667, 2.33333333, 3.        ])

In [49]: np.mgrid[0:9.,0:6.]

Out[49]: 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 [50]: np.ogrid[0:9.,0:6.]

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

```
In [51]: np.meshgrid([1,2,4],[2,4,5])

Out[51]: [array([[1, 2, 4],
                 [1, 2, 4],
                 [1, 2, 4]]), array([[2, 2, 2],
                 [4, 4, 4],
                 [5, 5, 5]])]

In [52]: np.ix_([1,2,4],[2,4,5])

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

In [53]: np.tile(a,(3,4))

Out[53]: array([[2, 3, 5, 2, 3, 5, 2, 3, 5, 2, 3, 5],
                [3, 6, 8, 3, 6, 8, 3, 6, 8, 3, 6, 8],
                [3, 5, 7, 3, 5, 7, 3, 5, 7, 3, 5, 7],
                [2, 3, 5, 2, 3, 5, 2, 3, 5, 2, 3, 5],
                [3, 6, 8, 3, 6, 8, 3, 6, 8, 3, 6, 8],
                [3, 5, 7, 3, 5, 7, 3, 5, 7, 3, 5, 7],
                [2, 3, 5, 2, 3, 5, 2, 3, 5, 2, 3, 5],
                [3, 6, 8, 3, 6, 8, 3, 6, 8, 3, 6, 8],
                [3, 5, 7, 3, 5, 7, 3, 5, 7, 3, 5, 7]])

In [54]: a = np.arange(1.,11.)[:, np.newaxis]
         b = np.arange(1.,11.)[:, np.newaxis]
         np.c_[a,b]

Out[54]: array([[  1.,   1.],
                [  2.,   2.],
                [  3.,   3.],
                [  4.,   4.],
                [  5.,   5.],
                [  6.,   6.],
                [  7.,   7.],
                [  8.,   8.],
                [  9.,   9.],
                [ 10.,  10.]])

In [55]: np.r_[a,b]

Out[55]: array([[  1.],
                [  2.],
                [  3.],
                [  4.],
                [  5.],
                [  6.],
               [ 7.],
               [ 8.],
               [ 9.],
               [10.],
               [ 1.],
               [ 2.],
               [ 3.],
               [ 4.],
               [ 5.],
               [ 6.],
               [ 7.],
               [ 8.],
               [ 9.],
               [10.]])
```

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

Out[56]: 10.0

In [57]: a.max(1)

Out[57]: array([ 1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.])

In [58]: np.maximum(a,b)

Out[58]: array([[ 1.],
               [ 2.],
               [ 3.],
               [ 4.],
               [ 5.],
               [ 6.],
               [ 7.],
               [ 8.],
               [ 9.],
               [10.]])

In [59]: np.linalg.norm(a)

Out[59]: 19.621416870348583

In [60]: a = np.array([[1,2],[3,4]])
         b = np.array([[3,4],[4,5]])
         np.logical_and(a,b)

Out[60]: array([[ True,  True],
               [ True,  True]])

In [61]: np.logical_or(a,b)

Out[61]: array([[ True,  True],
               [ True,  True]])
```

```
In [62]: a & b

Out[62]: array([[1, 0],
                [0, 4]], dtype=int32)

In [63]: a | b

Out[63]: array([[3, 6],
                [7, 5]], dtype=int32)

In [64]: np.linalg.inv(a)

Out[64]: array([[-2. ,  1. ],
                [ 1.5, -0.5]])

In [65]: np.linalg.pinv(a)

Out[65]: array([[-2. ,  1. ],
                [ 1.5, -0.5]])

In [66]: np.linalg.matrix_rank(a)

Out[66]: 2

In [67]: np.linalg.solve(a,b)

Out[67]: array([[-2. , -3. ],
                [ 2.5,  3.5]])

In [68]: np.linalg.solve(a.T,b.T)

Out[68]: array([[ 0. , -0.5],
                [ 1. ,  1.5]])

In [69]: U, S, Vh = np.linalg.svd(a)
         print(U)
         print(S)
         print(Vh)
         V = Vh.T
         print(V)

[[-0.40455358 -0.9145143 ]
 [-0.9145143   0.40455358]]
[5.4649857  0.36596619]
[[-0.57604844 -0.81741556]
 [ 0.81741556 -0.57604844]]
[[-0.57604844  0.81741556]
 [-0.81741556 -0.57604844]]
```

```
In [70]: a = np.eye(3)
         np.linalg.cholesky(a).T

Out[70]: array([[1., 0., 0.],
                [0., 1., 0.],
                [0., 0., 1.]])

In [71]: D,V = np.linalg.eig(a)
         print(D)
         print(V)

[1. 1. 1.]
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]


In [72]: Q,R = scipy.linalg.qr(a)
         print(Q)
         print(R)

[[ 1.  0.  0.]
 [-0.  1.  0.]
 [-0. -0.  1.]]
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]


In [73]: L,U = scipy.linalg.lu_factor(a)
         print(L)
         print(U)

[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
[0 1 2]


In [74]: scipy.sparse.linalg.cg

Out[74]: <function scipy.sparse.linalg.isolve.iterative.cg(A, b, x0=None, tol=1e-05, maxiter=Non

In [75]: scipy.fft(a)

Out[75]: array([[ 1. +0.j       ,  1. -0.j       ,  1. +0.j       ],
                [ 1. +0.j       , -0.5-0.8660254j, -0.5+0.8660254j],
                [ 1. +0.j       , -0.5+0.8660254j, -0.5-0.8660254j]])

In [76]: scipy.ifft(a)
```

```
Out[76]: array([[ 0.33333333-0.j        ,  0.33333333+0.j        ,
                  0.33333333-0.j        ],
                [ 0.33333333-0.j        , -0.16666667+0.28867513j,
                 -0.16666667-0.28867513j],
                [ 0.33333333-0.j        , -0.16666667-0.28867513j,
                 -0.16666667+0.28867513j]])
```

```
In [77]: np.sort(a)
```

```
Out[77]: array([[0., 0., 1.],
                [0., 0., 1.],
                [0., 0., 1.]])
```

```
In [78]: I = np.argsort(a[:,1])
         b = a[I,:]
         print(I)
         print(b)
```

```
[0 2 1]
[[1. 0. 0.]
 [0. 0. 1.]
 [0. 1. 0.]]
```

```
In [79]: X = np.array([[143, 145], [146, 147]])
         y = np.array([[88, 85], [88, 91]])
         np.linalg.lstsq(X,y, rcond=None)
```

```
Out[79]: (array([[-1.18120805,  4.69798658],
                 [ 1.77181208, -4.04697987]]),
          array([], dtype=float64),
          2,
          array([290.51460712,   0.51288299]))
```

```
In [80]: x = np.linspace(0, 5, 10, endpoint=False)
         y = np.cos(-x**4/3.0)
         scipy.signal.resample(y, 3)
```

```
Out[80]: array([0.50570367, 0.67677839, 0.06021344])
```

```
In [81]: a = np.array([[1,2,3],[3,4,5]])
         np.unique(a)
```

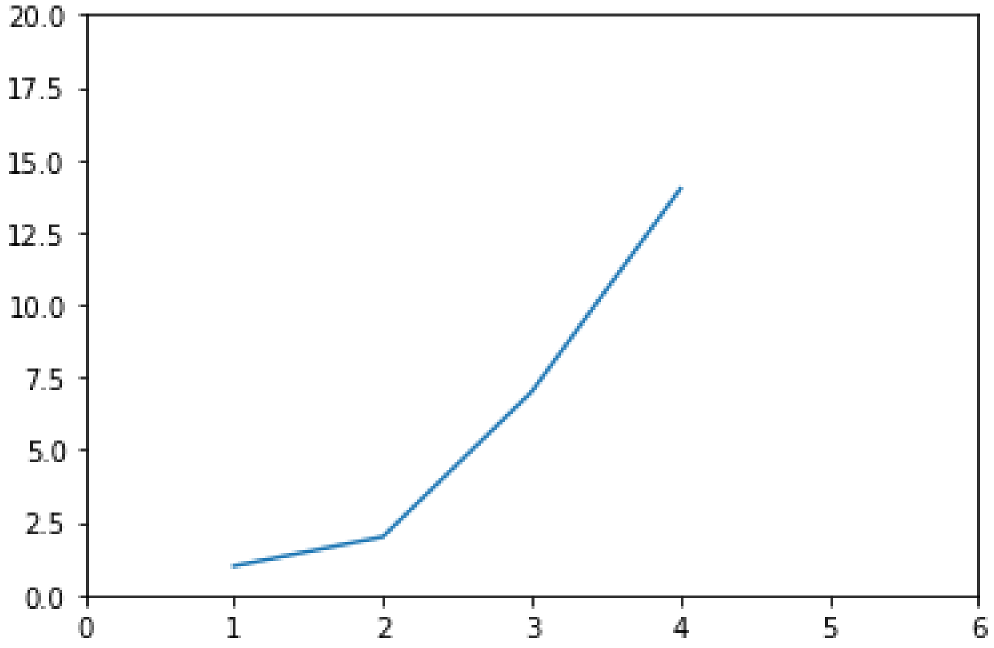
```
Out[81]: array([1, 2, 3, 4, 5])
```

```
In [82]: a.squeeze()
```

```
Out[82]: array([[1, 2, 3],
                [3, 4, 5]])
```

0.3 Task 3

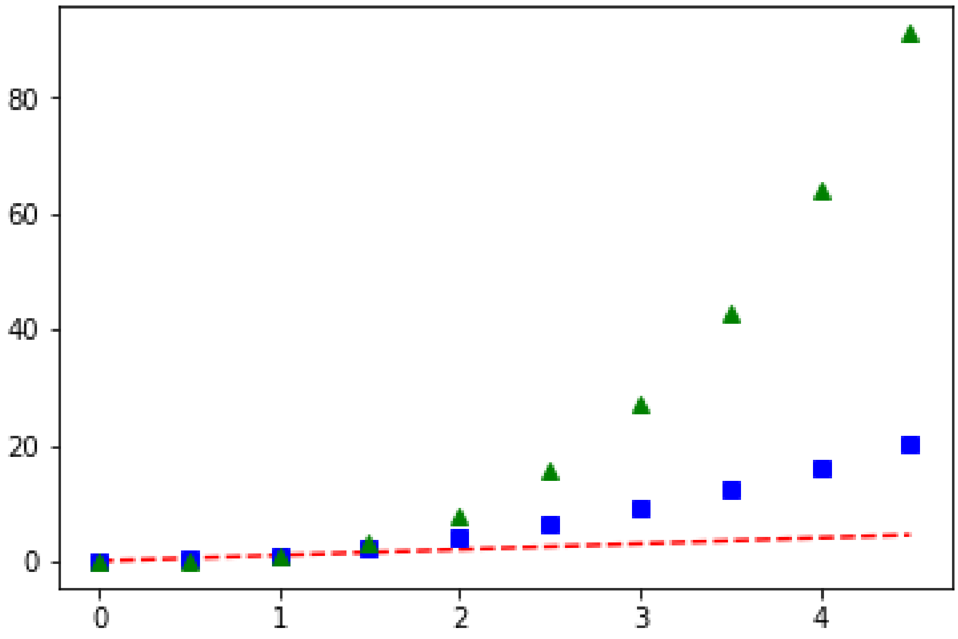
```
In [85]: import matplotlib.pyplot as plt
         plt.plot([1,2,3,4], [1,2,7,14])
         plt.axis([0, 6, 0, 20])
         plt.show()
```



0.4 Task 4

```
In [84]: import numpy as np
         import matplotlib.pyplot as plt

         t = np.arange(0., 5., 0.5)
         plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3, 'g^')
         plt.show()
```



0.5 Task 5

https://github.com/zihao623

0.6 Task 6

https://github.com/zihao623/ELEC576