

# ELEC 576 / COMP 576 – Fall 2022

## Assignment 0

Due: September 13, 2023 11 a.m. via Canvas

### Task1 :

```
(base) caosicheng@caosichengdeMacBook-Air ~ % conda info

active environment : base
active env location : /Users/caosicheng/anaconda3
shell level : 1
user config file : /Users/caosicheng/.condarc
populated config files : /Users/caosicheng/.condarc
conda version : 23.7.2
conda-build version : 3.26.0
python version : 3.11.4.final.0
virtual packages : __archspec=1=arm64
                  __osx=12.5=0
                  __unix=0=0
base environment : /Users/caosicheng/anaconda3 (writable)
conda av data dir : /Users/caosicheng/anaconda3/etc/conda
conda av metadata url : None
channel URLs : https://repo.anaconda.com/pkgs/main/osx-arm64
               https://repo.anaconda.com/pkgs/main/noarch
               https://repo.anaconda.com/pkgs/r/osx-arm64
               https://repo.anaconda.com/pkgs/r/noarch
package cache : /Users/caosicheng/anaconda3/pkgs
                /Users/caosicheng/.conda/pkgs
envs directories : /Users/caosicheng/anaconda3/envs
                  /Users/caosicheng/.conda/envs
platform : osx-arm64
user-agent : conda/23.7.2 requests/2.31.0 CPython/3.11.4 Darwin/21.6.0 OSX/12.5
UID:GID : 501:20
netrc file : /Users/caosicheng/.netrc
offline mode : False
```

## Task2:

```
(base) caosicheng@caosichengdeMacBook-Air ~ % ipython
Python 3.11.4 (main, Jul 5 2023, 08:54:11) [Clang 14.0.6 ]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.12.0 -- An enhanced Interactive Python. Type '?' for help.
```

```
In [1]: import numpy as np
...: from scipy import io, integrate, linalg, signal
...: from scipy.sparse.linalg import cg, eigs
```

```
In [2]: a = np.arange(8).reshape(2,4)
```

```
In [3]: a
Out[3]:
array([[0, 1, 2, 3],
       [4, 5, 6, 7]])
```

```
In [4]: np.ndim(a)
Out[4]: 2
```

```
In [5]: np.size(a)
Out[5]: 8
```

```
In [6]: np.shape(a)
Out[6]: (2, 4)
```

```
In [7]: a.shape[2-1]
Out[7]: 4
```

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

```
In [9]: b = np.array([1, 2, 3])
...: c = np.array([4, 5, 6])
...: np.block([b,c, 10])
Out[9]: array([ 1,  2,  3,  4,  5,  6, 10])
```

```
In [10]: a[-1]
Out[10]: array([4, 5, 6, 7])
```

```
In [11]: a[1,1]
Out[11]: 5
```

```
In [12]: a[0:1,1:1]
Out[12]: array([], shape=(1, 0), dtype=int64)
```

```
In [13]: a[1:]
Out[13]: array([[4, 5, 6, 7]])
```

```
[In [27]: d = np.arange(16).reshape(4,4)
```

```
[In [28]: d
```

```
Out[28]:
```

```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11],
       [12, 13, 14, 15]])
```

```
[In [29]: d[-4:]
```

```
Out[29]:
```

```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11],
       [12, 13, 14, 15]])
```

```
[In [30]: d[np.ix_([1, 3], [0, 2])]
```

```
Out[30]:
```

```
array([[ 4,  6],
       [12, 14]])
```

```
[In [31]: d[:,::2,:]
```

```
Out[31]:
```

```
array([[ 0,  2,  3],
       [ 8, 10, 11]])
```

```
[In [32]: d[np.r_[0:len(d),0]]
```

```
Out[32]:
```

```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11],
       [12, 13, 14, 15],
       [ 0,  1,  2,  3]])
```

```
[In [33]: d.transpose()
```

```
Out[33]:
```

```
array([[ 0,  4,  8, 12],
       [ 1,  5,  9, 13],
       [ 2,  6, 10, 14],
       [ 3,  7, 11, 15]])
```

```
[In [34]: d.conj().transpose()
```

```
Out[34]:
```

```
array([[ 0,  4,  8, 12],
       [ 1,  5,  9, 13],
       [ 2,  6, 10, 14],
       [ 3,  7, 11, 15]])
```



```

In [49]: a @ d
Out[49]:
array([[ 56,  62,  68,  74],
       [152, 174, 196, 218]])

In [50]: a**3
Out[50]:
array([[ 0,  1,  8, 27],
       [ 64, 125, 216, 343]])

In [51]: (a>0.5)
Out[51]:
array([[False,  True,  True,  True],
       [ True,  True,  True,  True]])

In [52]: np.nonzero(a>0.5)
Out[52]: (array([0, 0, 0, 1, 1, 1, 1]), array([1, 2, 3, 0, 1, 2, 3]))

In [53]: a[np.r_[0:len(a),0]]
Out[53]:
array([[0, 1, 2, 3],
       [4, 5, 6, 7],
       [0, 1, 2, 3]])

In [54]: e = a.copy()

In [55]: e
Out[55]:
array([[0, 1, 2, 3],
       [4, 5, 6, 7]])

In [56]: e = a.flatten()

In [57]: e
Out[57]: array([0, 1, 2, 3, 4, 5, 6, 7])

In [58]: np.arange(1., 11.)
Out[58]: array([ 1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.])

In [59]: np.arange(1.,11.)[:, np.newaxis]
Out[59]:
array([[ 1.],
       [ 2.],
       [ 3.],
       [ 4.],
       [ 5.],
       [ 6.],
       [ 7.],
       [ 8.],
       [ 9.],
       [10.]])

In [60]: np.zeros((3, 4, 5))
Out[60]:
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 [82]: np.eye(3)
Out[82]:
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])

In [83]: rng = np.random.default_rng(seed=42)
...: arr2 = rng.random((3, 3))
...: arr2
Out[83]:
array([[0.77395605, 0.43887844, 0.85859792],
       [0.69736803, 0.09417735, 0.97562235],
       [0.7611397 , 0.78606431, 0.12811363]])

[In [84]: np.linspace(1,3,4)
Out[84]: array([1.          , 1.66666667, 2.33333333, 3.          ])

[In [85]: np.mgrid[0:9,0:6.]
Out[85]:
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 [86]: a.max(0)
Out[86]: array([4, 5, 6, 7])

[In [87]: a.max(1)
Out[87]: array([3, 7])

[In [88]: np.linalg.norm(d)
Out[88]: 35.21363372331802

```

```
[In [92]: cg
Out[92]: <function scipy.sparse.linalg._isolve.iterative.cg(A, b, x0=None, tol=1e-05, maxiter=None, M=None, callback=None, atol=None)>

[In [93]: np.fft.fft(a)
Out[93]:
array([[ 6.+0.j, -2.+2.j, -2.+0.j, -2.-2.j],
       [22.+0.j, -2.+2.j, -2.+0.j, -2.-2.j]])

[In [94]: np.sort(a)
Out[94]:
array([[0, 1, 2, 3],
       [4, 5, 6, 7]])

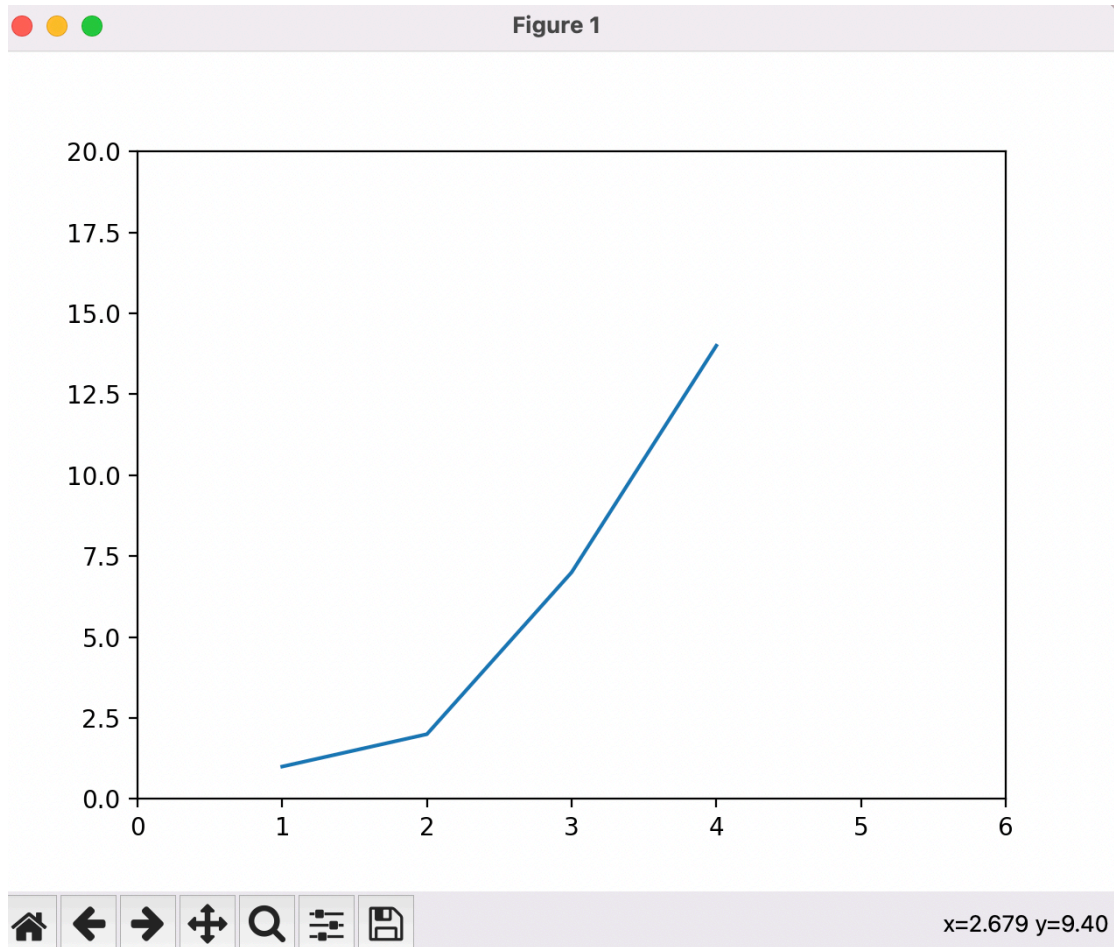
[In [95]: ^I
...: np.unique(a)
Out[95]: array([0, 1, 2, 3, 4, 5, 6, 7])

[In [96]:
...: a.squeeze()
Out[96]:
array([[0, 1, 2, 3],
       [4, 5, 6, 7]])
```

### Task3:

```
In [97]: import matplotlib.pyplot as plt
...: plt.plot([1,2,3,4], [1,2,7,14])
...: plt.axis([0, 6, 0, 20])
...: plt.show()
```

Installed osx event loop hook.





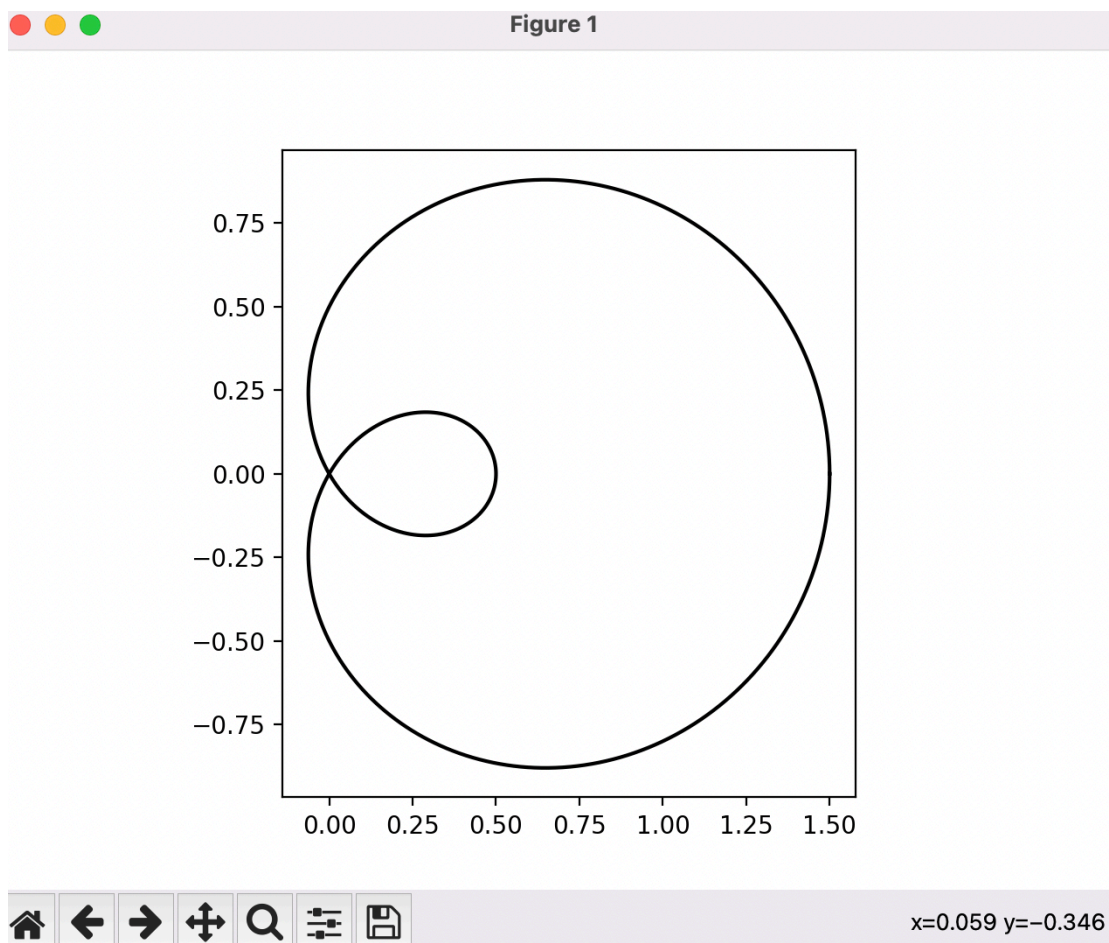
## Task4:

```
In [3]: import numpy as np
...:
...: import matplotlib.pyplot as plt
...: from matplotlib.path import Path
...: from matplotlib.patches import PathPatch
...:
...: N = 400
...: t = np.linspace(0, 2 * np.pi, N)
...: r = 0.5 + np.cos(t)
...: x, y = r * np.cos(t), r * np.sin(t)
...:
...: fig, ax = plt.subplots()
...: ax.plot(x, y, "k")
...: ax.set(aspect=1)
```

Installed osx event loop hook.

Out[3]: [None]

```
[In [4]: plt.show()
```





**Task5 :** <https://github.com/MajorTom000> (GitHub account)

**Task6:** <https://github.com/MajorTom000/576Machine-Learning/upload/main>