

COMP 576 Assignment 1 Report

Ziyi Zhao, zz89@rice.edu

Tue Sep 12 2023

Task 1: Conda Info Output

```
$ conda info

active environment : None
shell level        : 0
user config file    : /Users/tropping/.condarc
populated config files : /Users/tropping/.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=13.4.1=0
                      __unix=0=0
base environment     : /opt/homebrew/anaconda3 (writable)
conda av data dir     : /opt/homebrew/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        : /opt/homebrew/anaconda3/pkgs
                      /Users/tropping/.conda/pkgs
envs directories      : /opt/homebrew/anaconda3/envs
                      /Users/tropping/.conda/envs
platform             : osx-arm64
user-agent           : conda/23.7.2 requests/2.31.0 CPython/3.11.4
Darwin/22.5.0 OSX/13.4.1
UID:GID              : 501:20
netrc file           : None
offline mode         : False
```

Task 2: IPython Command Execution

```
$ ipython
[20:20:39]
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
...: import scipy.linalg
...:
```

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

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

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

```
-----
AttributeError                                Traceback (most recent call last)
Cell In[4], line 1
----> 1 a.dim
```

```
AttributeError: 'numpy.ndarray' object has no attribute 'dim'
```

```
In [5]: a.ndim
Out[5]: 2
```

```
In [6]: a.size
Out[6]: 6
```

```
In [7]: a.shape
Out[7]: (2, 3)
```

```
In [8]: a.shape[1]
Out[8]: 3
```

```
In [9]: a.shape[2]
```

```
-----
IndexError                                Traceback (most recent call last)
Cell In[9], line 1
----> 1 a.shape[2]
```

```
IndexError: tuple index out of range
```

```
In [10]: a.shape[0]
```

```
Out[10]: 2
```

```
In [11]: b = a
```

```
In [12]: c = a
```

```
In [13]: d = a
```

```
In [14]: np.block([[a, b], [c, d]])
```

```
Out[14]:
```

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

```
In [15]: e = np.block([[a, b], [c, d]])
```

```
In [16]: e
```

```
Out[16]:
```

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

```
In [17]: e.shape
```

```
Out[17]: (4, 6)
```

```
In [18]: e[0][0]
```

```
Out[18]: 1.0
```

```
In [19]: e[0][1]
```

```
Out[19]: 2.0
```

```
In [20]: e[0]
```

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

```
In [21]: e[0, :]
```

```
Out[21]: array([1., 2., 3., 1., 2., 3.])
```

```
In [22]: e[0:1][2:3]
```

```
Out[22]: array([], shape=(0, 6), dtype=float64)
```

```
In [23]: e[0:1, 2:3]
```

```
Out[23]: array([[3.]])
```

```
In [24]: e[0:2, 2:3]
```

```
Out[24]:
```

```
array([[3.],  
       [6.]])
```

```
In [25]: e[0:2, 2:4]
```

```
Out[25]:
```

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

```
In [26]: e[0:2][2:4]
```

```
Out[26]: array([], shape=(0, 6), dtype=float64)
```

```
In [27]: e[0:2]
```

```
Out[27]:
```

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

```
In [28]: e[0,0]
```

```
Out[28]: 1.0
```

```
In [29]: e[0,1]
```

```
Out[29]: 2.0
```

```
In [30]: e[0][1]
```

```
Out[30]: 2.0
```

```
In [31]: ^I
```

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

```
Out[31]:
```

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

```
In [32]: a.T
```

```
Out[32]:
```

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

```
In [33]: a
```

```
Out[33]:
```

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

```
In [34]: a.conj()
```

```
Out[34]:
```

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

```
In [35]: a.conj().T
```

```
Out[35]:
```

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

```
In [36]: a @ b
```

```
-----  
ValueError
```

```
Traceback (most recent call last)
```

```
Cell In[36], line 1
```

```
----> 1 a @ b
```

```
ValueError: matmul: Input operand 1 has a mismatch in its core dimension 0,  
with gufunc signature (n?,k),(k,m?)->(n?,m?) (size 2 is different from 3)
```

```
In [37]: a @ b.T
```

```
Out[37]:
```

```
array([[14., 32.],  
       [32., 77.]])
```

```
In [38]: a * b
```

```
Out[38]:
```

```
array([[ 1.,  4.,  9.],  
       [16., 25., 36.]])
```

```
In [39]: a * b.T
```

```
-----  
ValueError
```

```
Traceback (most recent call last)
```

```
Cell In[39], line 1
```

```
----> 1 a * b.T
```

```
ValueError: operands could not be broadcast together with shapes (2,3) (3,2)
```

```
In [40]: a/b
```

```
Out[40]:
```

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

```
In [41]: a**3
```

```
Out[41]:
```

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

```
In [42]: a > 0.5
```

```
Out[42]:
```

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

```
In [43]: (a > 0.5)
```

```
Out[43]:
```

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

```
In [44]: np.nonzero(a > 0.5)
```

```
Out[44]: (array([0, 0, 0, 1, 1, 1]), array([0, 1, 2, 0, 1, 2]))
```

```
In [45]: a[:,np.nonzero(v > 0.5)[0]]
```

```
NameError
```

```
Traceback (most recent call last)
```

```
Cell In[45], line 1
```

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

```
NameError: name 'v' is not defined
```

```
In [46]: a[a < 0.5]=0
```

```
In [47]: a[a > 3]=233
```

```
In [48]: a
```

```
Out[48]:
```

```
array([[ 1.,  2.,  3.],  
       [233., 233., 233.]])
```

```
In [49]: b
```

```
Out[49]:
```

```
array([[ 1.,  2.,  3.],  
       [233., 233., 233.]])
```

```
In [50]: b = a.copy()
```

```
In [51]: b
```

```
Out[51]:
```

```
array([[ 1.,  2.,  3.],  
       [233., 233., 233.]])
```

```
In [52]: b[0]=3
```

```
In [53]: b
```

```
Out[53]:
```

```
array([[ 3.,  3.,  3.],  
       [233., 233., 233.]])
```

```
In [54]: a
```

```
Out[54]:
```

```
array([[ 1.,  2.,  3.],  
       [233., 233., 233.]])
```

```
In [55]: a.flatten()
Out[55]: array([ 1.,  2.,  3., 233., 233., 233.])
```

```
In [56]: np.r_[1:10:10j]
Out[56]: array([ 1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.])
```

```
In [57]: help(np.r_)
```

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

```
In [59]: np.arange(1.,11.)
Cell In[59], line 1
      np.arange(1.,11.)
                ^
SyntaxError: unmatched '']'
```

```
In [60]: np.arange(1.,11.)
Out[60]: array([ 1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.])
```

```
In [61]: np.r_(1.,11.)
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[61], line 1
----> 1 np.r_(1.,11.)
```

```
TypeError: 'RClass' object is not callable
```

```
In [62]: np.r_[1.,11.]
Out[62]: array([ 1., 11.])
```

```
In [63]: np.r_[1.,11.]
Out[63]: array([ 1., 11.])
```

```
In [64]: np.r_[1.:11.]
Out[64]: array([ 1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.])
```

```
In [65]: np.r_[1.:11.]
Out[65]: array([ 1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.])
```

```
In [66]: np.r_[1.:11.][:]
Out[66]: array([ 1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.])
```

```
In [67]: np.r_[1.:11.][:,]
Out[67]: array([ 1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.])
```

```
In [68]: np.r_[1.:11.][:,np.newaxis]
```

```
Out[68]:
array([[ 1.],
       [ 2.],
       [ 3.],
       [ 4.],
       [ 5.],
       [ 6.],
       [ 7.],
       [ 8.],
       [ 9.],
       [10.]])
```

```
In [69]: np.r_[1.:11.][:,np.newaxis,np.newaxis]
```

```
Out[69]:
array([[[ 1.]],
       [[ 2.]],
       [[ 3.]],
       [[ 4.]],
       [[ 5.]],
       [[ 6.]],
       [[ 7.]],
       [[ 8.]],
       [[ 9.]],
       [[10.]])])
```



```
In [70]: np.eye(3,3,3)
```

```
Out[70]:
```

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

```
In [71]: np.eye(3,3)
```

```
Out[71]:
```

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

```
In [72]: np.diag(a)
```

```
Out[72]: array([ 1., 233.])
```

```
In [73]: np.diag(a,0)
```

```
Out[73]: array([ 1., 233.])
```

```
In [74]: from numpy.random import default_rng
```

```
...: rng = default_rng(42)
```

```
...: rng.random(3, 4)
```

```
-----  
TypeError
```

```
Traceback (most recent call last)
```

```
Cell In[74], line 3
```

```
1 from numpy.random import default_rng
```

```
2 rng = default_rng(42)
```

```
----> 3 rng.random(3, 4)
```

```
File _generator.pyx:296, in numpy.random._generator.Generator.random()
```

```
TypeError: Cannot interpret '4' as a data type
```

```
In [75]: rng
```

```
Out[75]: Generator(PCG64) at 0x11CB8B840
```

```
In [76]: rng.random(4,5)
```

```
-----  
TypeError
```

```
Traceback (most recent call last)
```

```
Cell In[76], line 1
```

```
----> 1 rng.random(4,5)
```

```
File _generator.pyx:296, in numpy.random._generator.Generator.random()
```

```
TypeError: Cannot interpret '5' as a data type
```

```
In [77]: rng.random(4)
```

```
Out[77]: array([0.77395605, 0.43887844, 0.85859792, 0.69736803])
```

```
In [78]: rng.random((4,5))
```

```
Out[78]:
```

```
array([[0.09417735, 0.97562235, 0.7611397 , 0.78606431, 0.12811363],
       [0.45038594, 0.37079802, 0.92676499, 0.64386512, 0.82276161],
       [0.4434142 , 0.22723872, 0.55458479, 0.06381726, 0.82763117],
       [0.6316644 , 0.75808774, 0.35452597, 0.97069802, 0.89312112]])
```

```
In [79]: np.mgrid[0:9.,0:6.]
```

```
Out[79]:
```

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

```
NameError
```

```
-----  
Traceback (most recent call last)
```

```
Cell In[80], line 1
```

```
----> 1 ogrid[0:9.,0:6.]
```

```
NameError: name 'ogrid' is not defined
```

```
In [81]: np.ix_(np.r_[0:9.],np.r_[0:6.]
```

```
...:
```

```
...:
```

```
...: )
```

```
Out[81]:
```

```
(array([[0.],
       [1.],
       [2.],
       [3.],
       [4.],
       [5.]
```

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

```
In [82]: np.ix_([1,2,4],[2,4,5])
```

```
Out[82]:
```

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

```
In [83]: np.tile(a, (0, 2))
```

```
Out[83]: array([], shape=(0, 6), dtype=float64)
```

```
In [84]: np.concatenate((a,b),1)
```

```
Out[84]:
```

```
array([[ 1.,  2.,  3.,  3.,  3.,  3.],
       [233., 233., 233., 233., 233., 233.]])
```

```
In [85]: np.vstack((a,b))
```

```
Out[85]:
```

```
array([[ 1.,  2.,  3.],
       [233., 233., 233.],
       [ 3.,  3.,  3.],
       [233., 233., 233.]])
```

```
In [86]: a.max()
```

```
Out[86]: 233.0
```

```
In [87]: ^I
```

```
...: a.max(0)
```

```
Out[87]: array([233., 233., 233.]])
```

```
In [88]: a.max(1)
```

```
Out[88]: array([ 3., 233.]])
```

```
In [89]: ^I
```

```
...: np.maximum(a, b)
```

```
Out[89]:
```

```
array([[ 3.,  3.,  3.],
       [233., 233., 233.]])
```

```
In [90]: np.linalg.norm(a)
```

```
Out[90]: 403.58518307787267
```

```
In [91]: ^I
```

```
...: logical_and(a,b)
```

```
-----
NameError                                Traceback (most recent call last)
Cell In[91], line 1
----> 1 logical_and(a,b)
```

NameError: name 'logical_and' is not defined

```
In [92]: ^I
...: np.logical_and(a,b)
Out[92]:
array([[ True,  True,  True],
       [ True,  True,  True]])
```

```
In [93]: ^I
...: np.logical_or(a,b)
Out[93]:
array([[ True,  True,  True],
       [ True,  True,  True]])
```

```
In [94]: ^I
...: a & b
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[94], line 1
----> 1 a & b
```

TypeError: ufunc 'bitwise_and' not supported for the input types, and the inputs could not be safely coerced to any supported types according to the casting rule ''safe''

```
In [95]: ^I
...: a | b
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[95], line 1
----> 1 a | b
```

TypeError: ufunc 'bitwise_or' not supported for the input types, and the inputs could not be safely coerced to any supported types according to the casting rule ''safe''

```
In [96]: ^I
...: linalg.inv(a)
```

```
-----
NameError                                Traceback (most recent call last)
Cell In[96], line 1
----> 1 linalg.inv(a)
```

```
NameError: name 'linalg' is not defined
```

```
In [97]: ^I
...: nplinalg.inv(a)
```

```
-----
NameError                                Traceback (most recent call last)
Cell In[97], line 1
----> 1 nplinalg.inv(a)
```

```
NameError: name 'nplinalg' is not defined
```

```
In [98]: ^I
...: np.linalg.inv(a)
```

```
-----
LinAlgError                              Traceback (most recent call last)
Cell In[98], line 1
----> 1 np.linalg.inv(a)
```

```
File <__array_function__ internals>:200, in inv(*args, **kwargs)
```

```
File /opt/homebrew/anaconda3/lib/python3.11/site-
packages/numpy/linalg/linalg.py:533, in inv(a)
```

```
    531 a, wrap = _makearray(a)
    532 _assert_stacked_2d(a)
--> 533 _assert_stacked_square(a)
    534 t, result_t = _commonType(a)
    536 signature = 'D->D' if isComplexType(t) else 'd->d'
```

```
File /opt/homebrew/anaconda3/lib/python3.11/site-
packages/numpy/linalg/linalg.py:190, in _assert_stacked_square(*arrays)
```

```
    188 m, n = a.shape[-2:]
    189 if m != n:
--> 190     raise LinAlgError('Last 2 dimensions of the array must be square')
```

```
LinAlgError: Last 2 dimensions of the array must be square
```

```
In [99]: np.^I
...: linalg.pinv(a)
Cell In[99], line 1
np.
```

```
      ^
SyntaxError: invalid syntax
```

```
In [100]: np.linalg.pinv(a)
Out[100]:
array([[ -5.00000000e-01,  5.72246066e-03],
       [ -5.41302892e-17,  1.43061516e-03],
```

```
[ 5.00000000e-01, -2.86123033e-03]])
```

```
In [101]: np.linalg.matrix_rank(a)
```

```
Out[101]: 2
```

```
In [102]: linalg.solve(a, b)
```

```
-----
NameError                                Traceback (most recent call last)
Cell In[102], line 1
----> 1 linalg.solve(a, b)
```

```
NameError: name 'linalg' is not defined
```

```
In [103]: np.linalg.solve(a, b)
```

```
-----
LinAlgError                              Traceback (most recent call last)
Cell In[103], line 1
----> 1 np.linalg.solve(a, b)
```

```
File <__array_function__ internals>:200, in solve(*args, **kwargs)
```

```
File /opt/homebrew/anaconda3/lib/python3.11/site-
packages/numpy/linalg/linalg.py:373, in solve(a, b)
```

```
    371 a, _ = _makearray(a)
    372 _assert_stacked_2d(a)
--> 373 _assert_stacked_square(a)
    374 b, wrap = _makearray(b)
    375 t, result_t = _commonType(a, b)
```

```
File /opt/homebrew/anaconda3/lib/python3.11/site-
packages/numpy/linalg/linalg.py:190, in _assert_stacked_square(*arrays)
```

```
    188 m, n = a.shape[-2:]
    189 if m != n:
--> 190     raise LinAlgError('Last 2 dimensions of the array must be square')
```

```
LinAlgError: Last 2 dimensions of the array must be square
```

```
In [104]: ^I
```

```
...: U, S, Vh = linalg.svd(a); V = Vh.T
```

```
-----
NameError                                Traceback (most recent call last)
Cell In[104], line 1
----> 1 U, S, Vh = linalg.svd(a); V = Vh.T
```

```
NameError: name 'linalg' is not defined
```

```
In [105]: ^I
```

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

```
-----
NameError                                Traceback (most recent call last)
Cell In[105], line 1
----> 1 D,V = linalg.eig(a)
```

NameError: name 'linalg' is not defined

In [106]: from numpy import linalg

```
In [107]: ^I
...: D,V = linalg.eig(a)
```

```
-----
LinAlgError                                Traceback (most recent call last)
Cell In[107], line 1
----> 1 D,V = linalg.eig(a)
```

File <__array_function__ internals>:200, in eig(*args, **kwargs)

File /opt/homebrew/anaconda3/lib/python3.11/site-packages/numpy/linalg/linalg.py:1297, in eig(a)

```
1295 a, wrap = _makearray(a)
1296 _assert_stacked_2d(a)
-> 1297 _assert_stacked_square(a)
1298 _assert_finite(a)
1299 t, result_t = _commonType(a)
```

File /opt/homebrew/anaconda3/lib/python3.11/site-packages/numpy/linalg/linalg.py:190, in _assert_stacked_square(*arrays)

```
188 m, n = a.shape[-2:]
189 if m != n:
--> 190     raise LinAlgError('Last 2 dimensions of the array must be square')
```

LinAlgError: Last 2 dimensions of the array must be square

```
In [108]: ^I
...: D,V = linalg.eig(a, b)
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[108], line 1
----> 1 D,V = linalg.eig(a, b)
```

File <__array_function__ internals>:198, in eig(*args, **kwargs)

TypeError: eig() takes 1 positional argument but 2 were given

```
In [109]: ^I
...: D,V = eigs(a, k=3)
```

```
NameError                                Traceback (most recent call last)
Cell In[109], line 1
----> 1 D,V = eigs(a, k=3)
```

NameError: name 'eigs' is not defined

```
In [110]: ^I
...: D,V = linalg.eigs(a, k=3)
```

```
-----
AttributeError                            Traceback (most recent call last)
Cell In[110], line 1
----> 1 D,V = linalg.eigs(a, k=3)
```

AttributeError: module 'numpy.linalg' has no attribute 'eigs'

```
In [111]: ^I
...: Q,R = linalg.qr(a)
```

```
In [112]: Q
Out[112]:
array([[ -0.00429181, -0.99999079],
       [-0.99999079,  0.00429181]])
```

```
In [113]: R
Out[113]:
array([[ -233.00214591, -233.00643772, -233.01072952],
       [  0.          , -0.99999079, -1.99998158]])
```

```
In [114]: ^I
...: cg
```

```
-----
NameError                                Traceback (most recent call last)
Cell In[114], line 1
----> 1 cg
```

NameError: name 'cg' is not defined

```
In [115]: np.cg
```

```
-----
AttributeError                            Traceback (most recent call last)
Cell In[115], line 1
----> 1 np.cg
```

```
File /opt/homebrew/anaconda3/lib/python3.11/site-
packages/numpy/__init__.py:320, in __getattr__(attr)
    317     from .testing import Tester
    318     return Tester
--> 320 raise AttributeError("module {!r} has no attribute "
```



```
321                                     "{!r}".format(__name__, attr))
```

AttributeError: module 'numpy' has no attribute 'cg'

```
In [116]: np.fft.fft(a)
```

```
Out[116]:
```

```
array([[ 6. +0.j          , -1.5+0.8660254j, -1.5-0.8660254j],
       [699. +0.j         ,  0. +0.j         ,  0. +0.j         ]])
```

```
In [117]: ^I
```

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

```
Out[117]:
```

```
array([[ 2. +0.j          , -0.5-0.28867513j, -0.5+0.28867513j],
       [233. +0.j         ,  0. +0.j         ,  0. +0.j         ]])
```

```
In [118]: ^I
```

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

```
Out[118]:
```

```
array([[ 1.,  2.,  3.],
       [233., 233., 233.]])
```

```
In [119]: sort(a, 2)
```

```
...:
```

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

```
-----
NameError                                Traceback (most recent call last)
```

```
Cell In[119], line 1
```

```
----> 1 sort(a, 2)
```

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

NameError: name 'sort' is not defined

```
In [120]: I = np.argsort(a[:, 0]); b = a[I,:]
```

```
In [121]: ^I
```

```
...: x = linalg.lstsq(Z, y)
```

```
-----
NameError                                Traceback (most recent call last)
```

```
Cell In[121], line 1
```

```
----> 1 x = linalg.lstsq(Z, y)
```

NameError: name 'Z' is not defined

```
In [122]: ^I
```

```
...: np.unique(a)
```

```
Out[122]: array([ 1.,  2.,  3., 233.])
```

```
In [123]: ^I
```

```
...: a.squeeze()
Out[123]:
array([[ 1.,  2.,  3.],
       [233., 233., 233.]])
```

```
In [124]: ^I
...: x = linalg.lstsq(Z, y)
```

```
-----
NameError                                Traceback (most recent call last)
Cell In[124], line 1
----> 1 x = linalg.lstsq(Z, y)
```

NameError: name 'Z' is not defined

```
In [125]: signal.resample(x, np.ceil(len(x)/q))
```

```
-----
NameError                                Traceback (most recent call last)
Cell In[125], line 1
----> 1 signal.resample(x, np.ceil(len(x)/q))
```

NameError: name 'signal' is not defined

```
In [126]: np.signal.resample(x, np.ceil(len(x)/q))
```

```
-----
AttributeError                            Traceback (most recent call last)
Cell In[126], line 1
----> 1 np.signal.resample(x, np.ceil(len(x)/q))
```

```
File /opt/homebrew/anaconda3/lib/python3.11/site-
packages/numpy/__init__.py:320, in __getattr__(attr)
    317     from .testing import Tester
    318     return Tester
--> 320 raise AttributeError("module {!r} has no attribute "
    321                        "{!r}".format(__name__, attr))
```

AttributeError: module 'numpy' has no attribute 'signal'

```
In [127]: ^I
...: a.squeeze()
```

```
Out[127]:
array([[ 1.,  2.,  3.],
       [233., 233., 233.]])
```

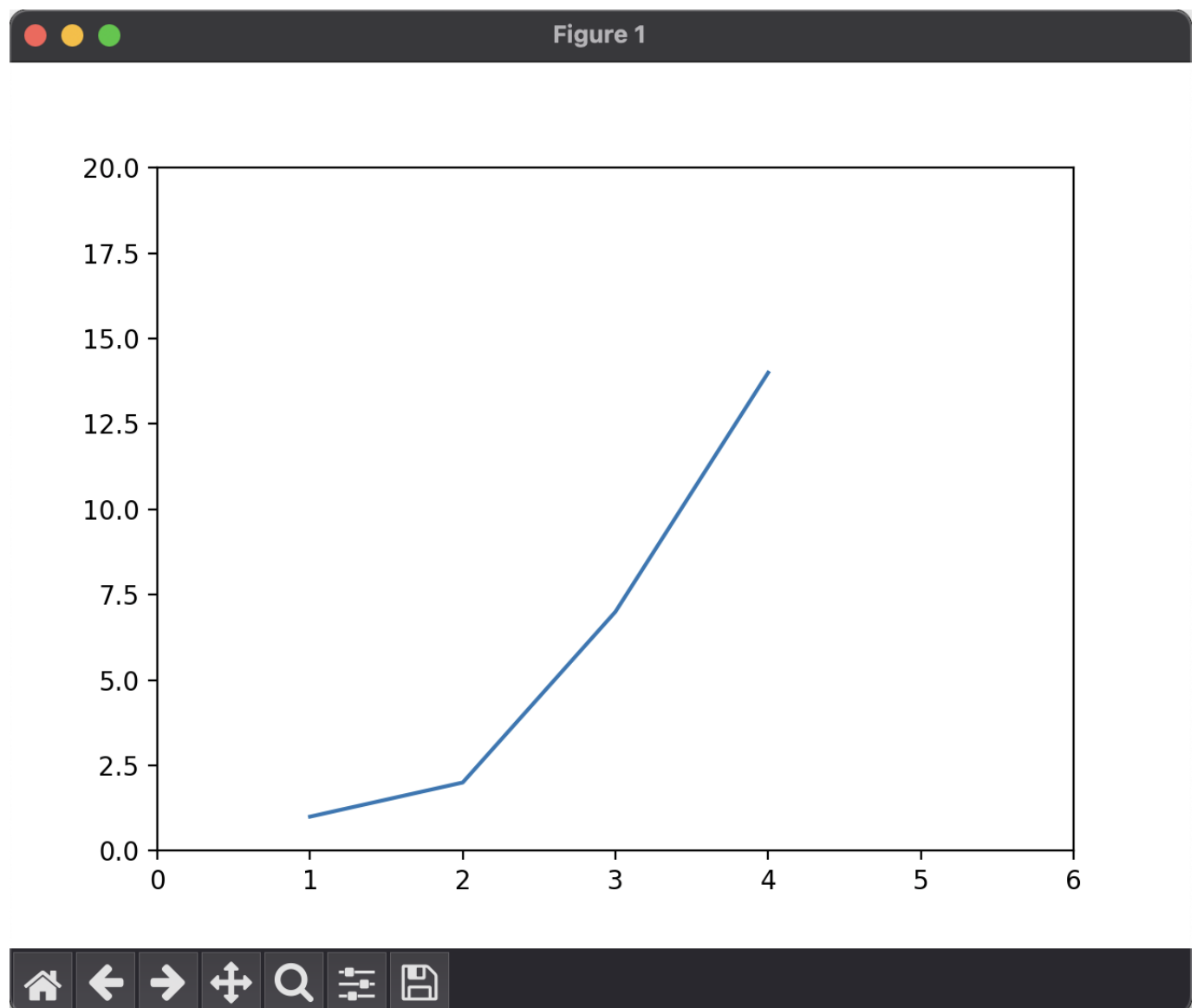
```
In [128]: a.T
```

```
Out[128]:
array([[ 1., 233.],
       [ 2., 233.],
       [ 3., 233.]])
```

```
In [129]: np.fft.ifft(a)
Out[129]:
array([[ 2. +0.j          , -0.5-0.28867513j, -0.5+0.28867513j],
       [233. +0.j        ,  0. +0.j        ,  0. +0.j        ]])
```

Task 3: Plotting

Plotted figure:



Task 4: Plotting

```
In [11]: import numpy as np
```

```
In [12]: x = np.r_[-10.:10.:0.5]
```

```
In [13]: x
```

```
Out[13]:
```

```
array([-10. , -9.5, -9. , -8.5, -8. , -7.5, -7. , -6.5, -6. ,  
       -5.5, -5. , -4.5, -4. , -3.5, -3. , -2.5, -2. , -1.5,  
       -1. , -0.5,  0. ,  0.5,  1. ,  1.5,  2. ,  2.5,  3. ,  
        3.5,  4. ,  4.5,  5. ,  5.5,  6. ,  6.5,  7. ,  7.5,  
        8. ,  8.5,  9. ,  9.5])
```

```
In [14]: y=np.sin(x)
```

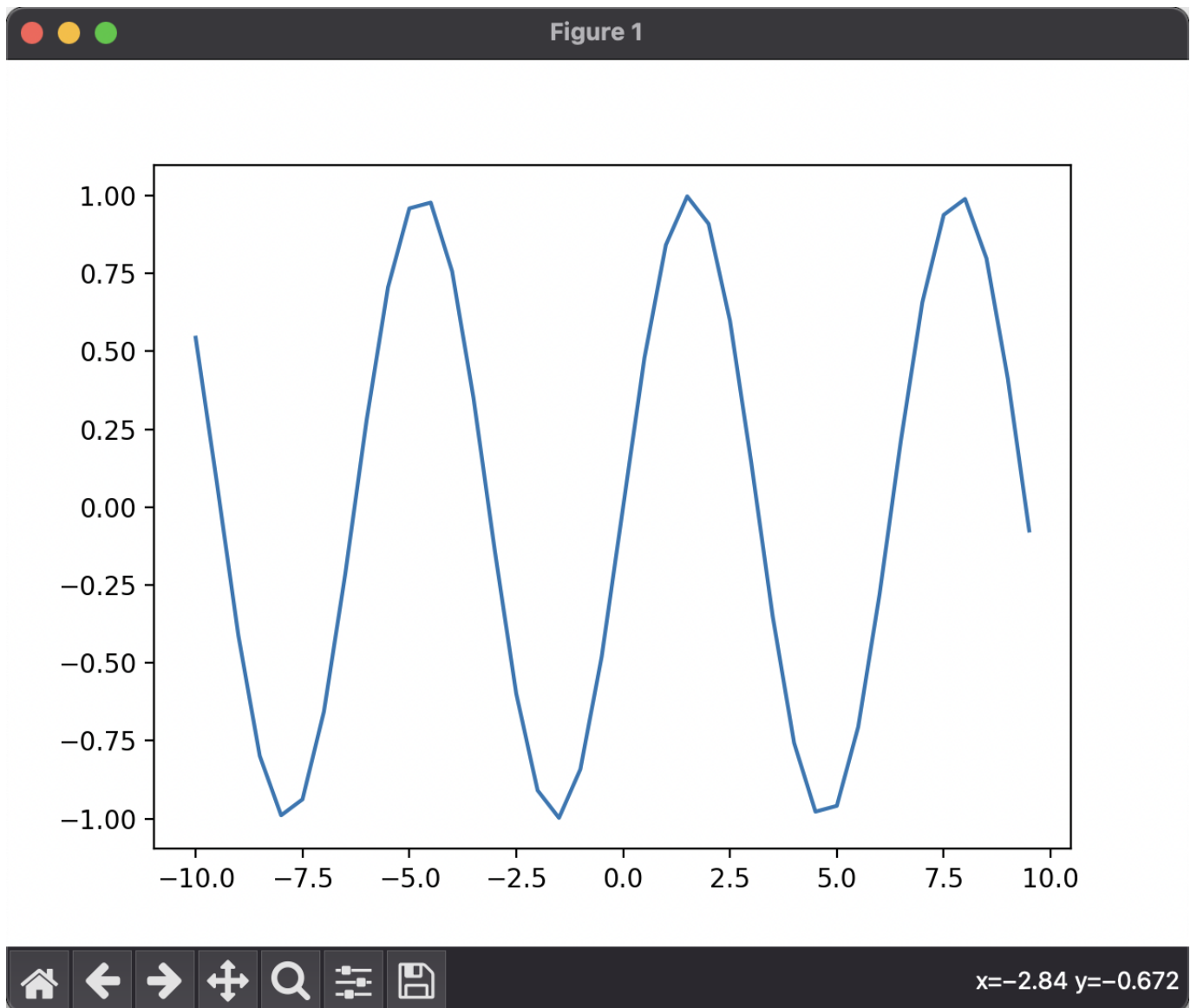
```
In [15]: plt.plot(x,y)
```

```
Out[15]: [<matplotlib.lines.Line2D at 0x16b2a1250>]
```

```
In [16]: plt.plot(x,-y, 'co')
```

```
Out[16]: [<matplotlib.lines.Line2D at 0x16b410850>]
```

```
In [17]: plt.show()
```



Task 5: VCS Account

My Github Account: Tr0py <https://github.com/tr0py/>

Task 6: GitHub Project

My Github Project: [Tr0py/COMP576-hw](#) · GitHub