

t2

September 11, 2019

```
[1]: # Task 2
import numpy as np
import scipy.linalg
import scipy.sparse.linalg
import scipy.signal
a = np.random.rand(25,25)
b = np.random.rand(25,25)
c = np.random.rand(25,25)
d = np.random.rand(25,25)
v = np.random.rand(1,25)
x = np.random.rand(25,25)
atemp = a

a.ndim
a.size
a.shape
a.shape[0]
np.array([[1.,2.,3.],[4.,5.,6.]])
np.block([[a,b],[c,d]])
a[-1]
a[1,4]
a[1]
a[0:5]
a[-5:]
a[0:3][:,4:9]
a[np.ix_([1,3,4],[0,2])]
a[2:32:2,: ]
a[:,2,: ]
a[:, :-1, :]
a[np.r_[len(a),0]]
a.T
a.conj().T
a@b
a*b
a/b
a**3
a>0.5
```

```

np.nonzero(a>0.5)
a[:,np.nonzero(v>0.5)[0]]
atemp[a<0.5] = 0
a*(a>0.5)
atemp[:] = 3
y = x.copy()
y = x[1,:].copy()
y = x.flatten()
np.arange(1.,11.)
np.arange(10.)
np.arange(1.,11.)[:, np.newaxis]
np.zeros((3,4))
np.zeros((3,4,5))
np.ones((3,4))
np.eye(3)
np.diag(a)
np.diag(a,0)
np.random.rand(3,4)
np.linspace(1,3,4)
np.mgrid[0:9., 0:6.]
np.ogrid[0:9., 0:6.]
np.meshgrid([1,2,4],[2,4,5])
np.tile(a, (2,3))
np.concatenate((a,b),1)
np.concatenate((a,b))
a.max()
a.max(0)
np.maximum(a,b)
np.linalg.norm(v)
np.logical_and(a,b)
np.logical_or(a,b)
np.linalg.inv(np.diag(np.diag(a),0))
np.linalg.pinv(a)
np.linalg.matrix_rank(a)
np.linalg.solve(np.diag(np.diag(a)),b)
U,S,Vh = np.linalg.svd(a)
np.linalg.cholesky(np.eye(25)) # don't want to worry about PSD
D,V =np.linalg.eig(a)
D,V = scipy.linalg.eig(a,b)
Q,R = scipy.linalg.qr(a)
scipy.sparse.linalg.cg
np.fft.fft(a)
np.fft.ifft(a)
a.sort()
I = np.argsort(a[:,3])
np.linalg.lstsq(a,x)
scipy.signal.resample(x, int(len(x)/5))

```

```
np.unique(a)
a.squeeze()
```

```
/home/matt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:83:
FutureWarning: `rcond` parameter will change to the default of machine precision
times ``max(M, N)`` where M and N are the input matrix dimensions.
To use the future default and silence this warning we advise to pass
`rcond=None`, to keep using the old, explicitly pass `rcond=-1`.
```

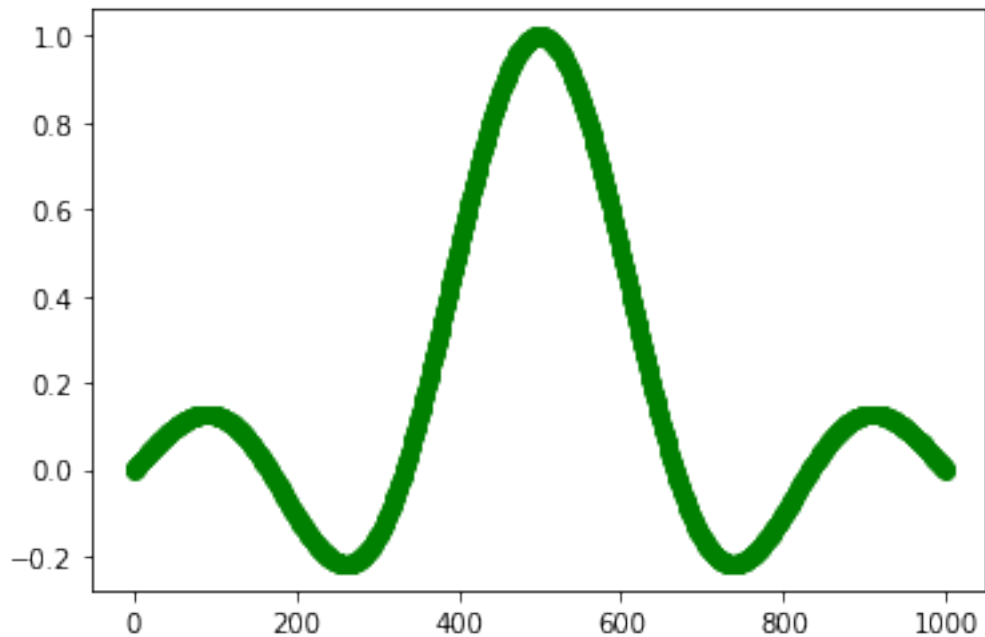
[illegible]

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

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

<Figure size 640x480 with 1 Axes>

```
[3]: # Task 4
plt.plot(np.sinc(np.linspace(-3,3,1000)), 'g')
plt.show()
```



[: