Matrix Condition Number

January 30, 2019

```
In [4]: import numpy as np
        import math
        import matplotlib.pyplot as plt
```

```
In [42]: x = np.array([[1., 2.5, 3.],[1., 3.5, 3.]])
```

```
In [43]: x.shape
```

```
Out[43]: (2, 3)
```

```
In [44]: y = np.array([3., 4., 9.])
```

```
In [45]: y.shape
```

```
Out[45]: (3,)
```

```
In [35]: # don't do this
         z = []
         for i in range(3):
             z.append(x[i] + y[i])
         np.array(z)
```

```
Out[35]: array([ 4.  ,  6.5, 12. ])
```

```
In [46]: # do this instead
         x * y
```

```
Out[46]: array([[ 3., 10., 27.],
                [ 3., 14., 27.]])
```

```
In [32]: # dot product
         np.dot(x, y)
```

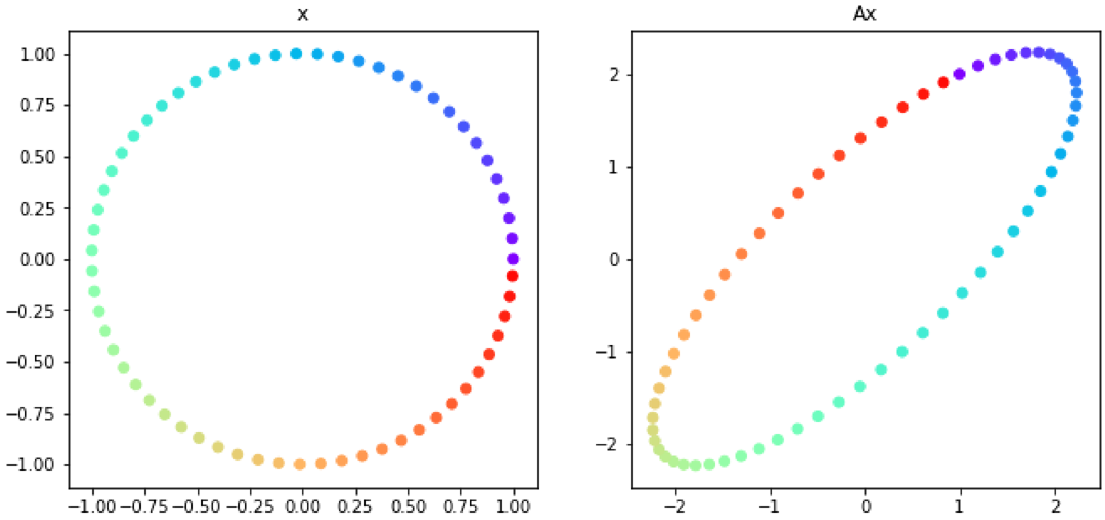
```
Out[32]: 40.0
```

```
In [6]: def plot_effect(A):
            assert A.shape == (2, 2)
            # create points evenly spaced on a sphere
            rad = np.arange(0, 2*math.pi, 0.1)
            xs = np.cos(rad)
```

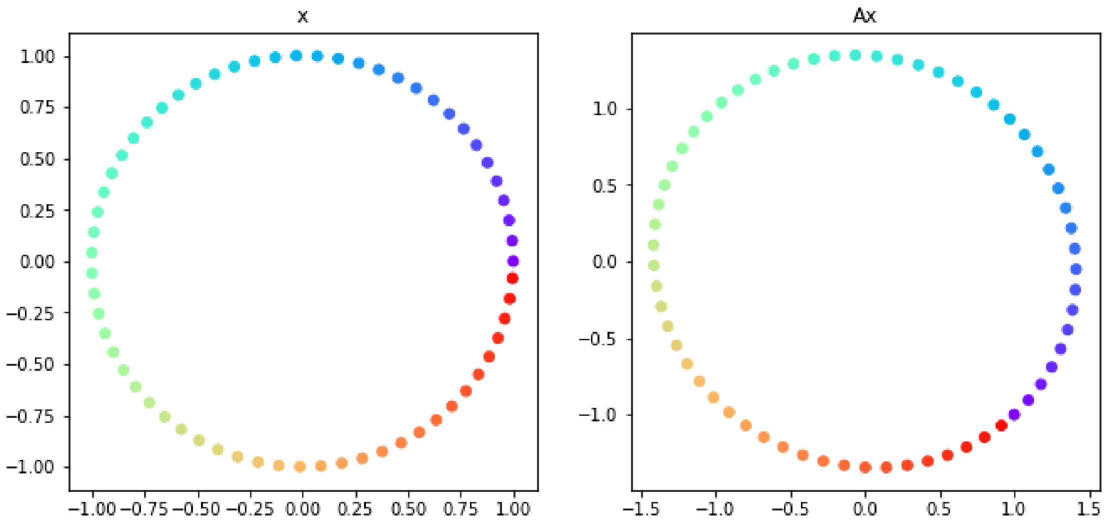
```
    ys = np.sin(rad)
    # apply "A" to the points
    Ax = np.matmul(A, np.stack([xs, ys]))

    # plot the figure
    plt.figure(figsize=(11,5))
    plt.subplot(1, 2, 1)
    plt.scatter(xs, ys, color=plt.cm.rainbow(rad/2/math.pi))
    plt.title("x")
    plt.subplot(1, 2, 2)
    plt.scatter(Ax[0,:], Ax[1,:], color=plt.cm.rainbow(rad/2/math.pi))
    plt.title("Ax")
```

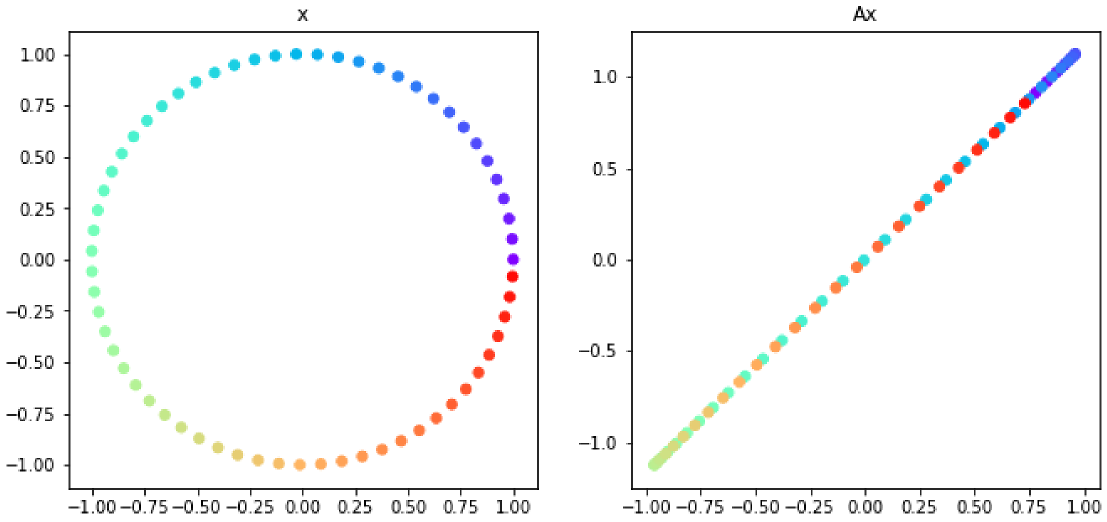
```
In [10]: plot_effect(np.array([[1., 2.],
                               [2., 1.]]))
```



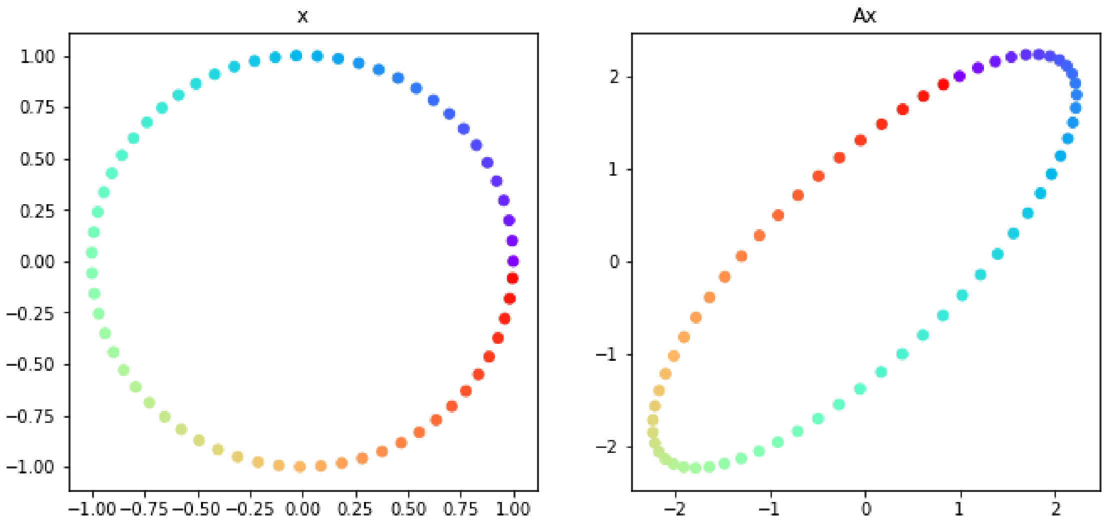
```
In [18]: plot_effect(np.array([[1., 1.],
                               [-1., 0.9]]))
```



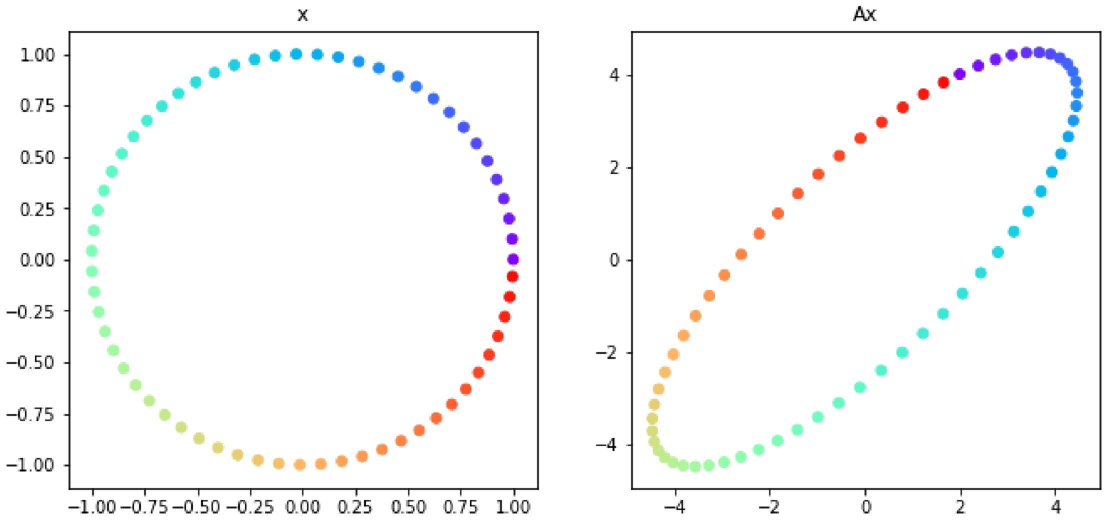
```
In [47]: plot_effect(np.array([[0.780, 0.563],
                               [0.913, 0.659]]))
```



```
In [48]: plot_effect(np.array([[1., 2.],
                               [2., 1.]]))
```



```
In [49]: plot_effect(np.array([[2., 4.],
                                [4., 2.]]))
```


x
Ax