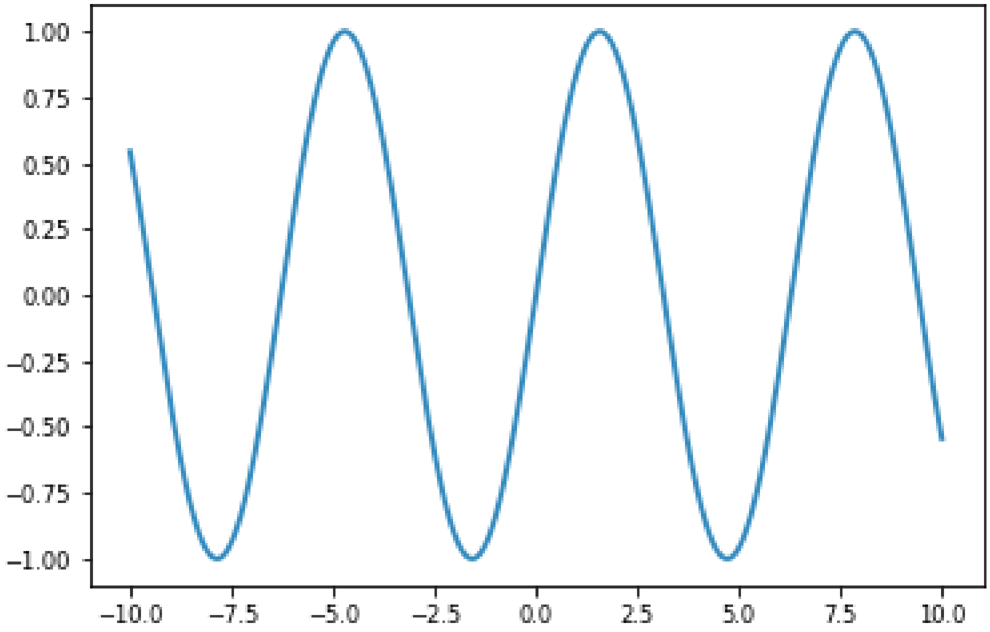
My-test

August 25, 2017

$e^{i\pi} + 1 = 0$

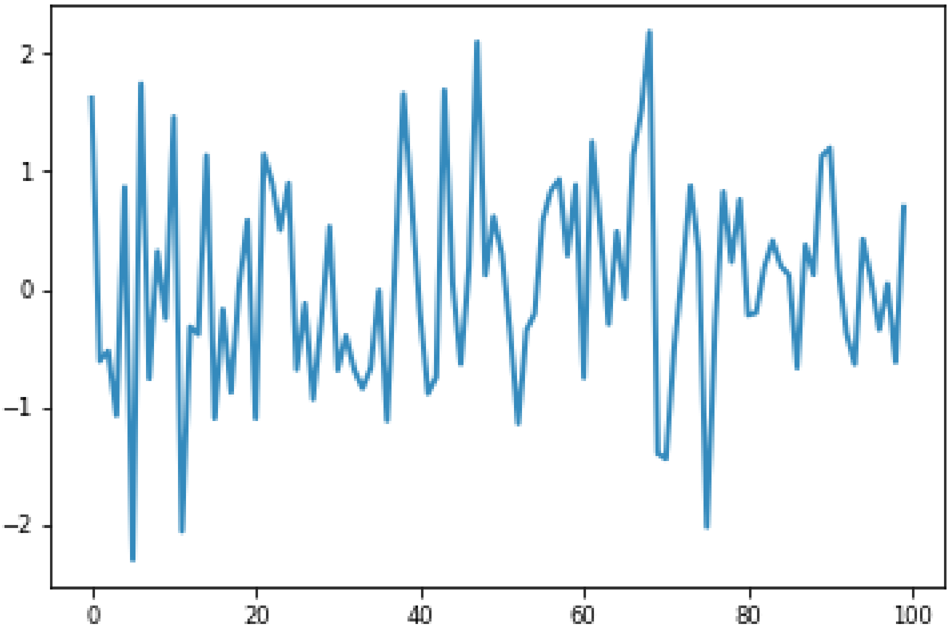
```
In [7]: import numpy as np
        import matplotlib.pyplot as plt
        x=np.linspace(-10,10,1000)
        y=np.sin(x)
        plt.plot(x,y)
        plt.show()
        %matplotlib inline
        import pylab as pl
        pl.seed(1)
        data = pl.randn(100)
        pl.plot(data)
```

```
d:\program files\python\lib\site-packages\matplotlib\__init__.py:913: UserWarning: axes.color_c
  warnings.warn(self.msg_depr % (key, alt_key))
d:\program files\python\lib\site-packages\matplotlib\font_manager.py:1297: UserWarning: findfor
  (prop.get_family(), self.defaultFamily[fontext]))
```



```
Out[7]: [<matplotlib.lines.Line2D at 0x24b0caecac8>]
```

```
d:\program files\python\lib\site-packages\matplotlib\font_manager.py:1297: UserWarning: findfor
  (prop.get_family(), self.defaultFamily[fontext]))
```



```
In [8]: a=[1,2,3]
        %timeit a[1]=10
```

```
119 ns ś 4.7 ns per loop (mean ś std. dev. of 7 runs, 10000000 loops each)
```

```
In [9]: %%timeit
        a=[]
        for i in range(10):
            a.append(i)
```

```
2.59 ţs ś 111 ns per loop (mean ś std. dev. of 7 runs, 100000 loops each)
```

```
In [10]: %%time
         a=[]
         for i in range(10000):
             a.append(i)
```

```
Wall time: 4 ms
```

```
In [11]: %%capture time_results
         import random
         for n in [1000,5000,10000,50000]:
             print("n={0}".format(n))
             alist=list(range(n))
             %time random.shuffle(alist)
         print(time_results.stderr,time_results.stdout)
```

```
In [12]: %%capture time_results
         import random
         timeit_results=[]
         for n in [1000,5000,10000,50000]:
             alist=[random.random() for i in range(n)]
             res = %timeit -o sorted(alist)
             timeit_results.append((n,res))
         #print(time_results.stderr,time_results.stdout)
```

```
In [13]: %%prun
         def fib(n):
             if n<2:
                 return 1
             else:
                 return fib(n-1)+fib(n-2)
         def fib_fast(n,a=1,b=1):
             if n==1:
                 return b
             else:
                 return fib_fast(n-1,b,a+b)
         %time fib(20)
         %time fib_fast(20)
```

```
Wall time: 17.1 ms
Wall time: 0 ns
```

```
In [15]: import math
         def sinc(x):
             return math.sin(x)/x
         [sinc(x) for x in range(1,5)]
```

```
Out[15]: [0.8414709848078965,
          0.45464871341284085,
          0.0470400026866224,
          -0.18920062382698205]
```

```
In [16]: %debug
```

```
> <ipython-input-14-dc428f8d2f58>(3)sinc()
      1 import math
      2 def sinc(x):
----> 3     return math.sin(x)/x
      4 [sinc(x) for x in range(5)]

ipdb> p x
0
ipdb> q
```

```
In [17]: from IPython import display
         for i in range(2,4):
             display.display(display.Latex("$x^{i}+y^{i}$".format(i=i)))
         display.Latex("$x^4+y^4$")
```

$x^2 + y^2$

$x^3 + y^3$

Out[17]:

$x^4 + y^4$

```
In [18]: logurl="https://www.python.org/static/community_logos/python-logo-master-v3-TM.png"
         display.Image(url=logurl,embed=True)
```

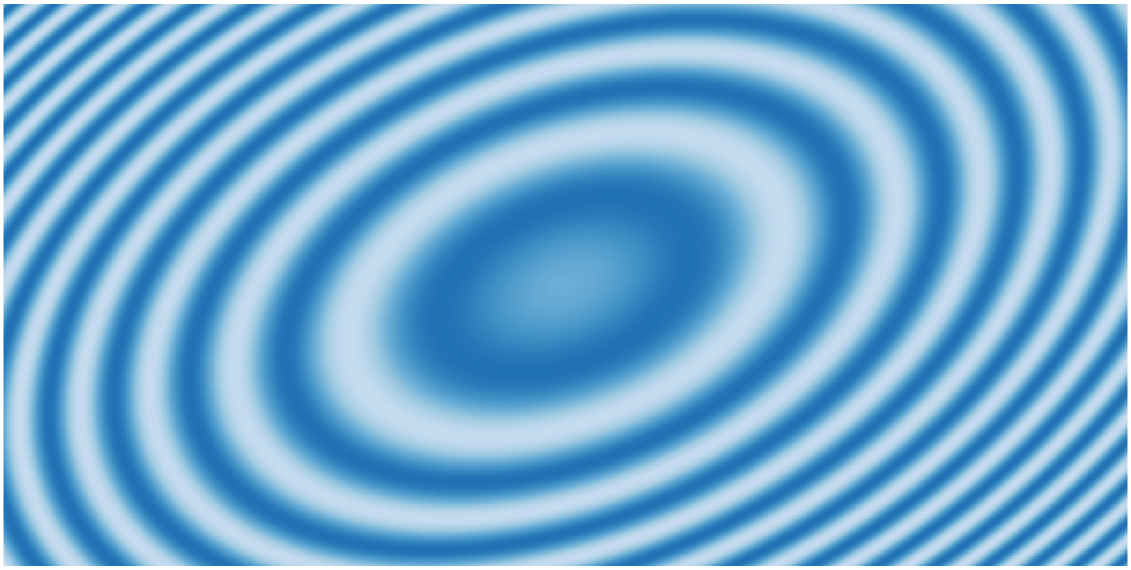
Out[18]:

```
In [19]: def as_png(img,**kw):
             import io
             from matplotlib import image
             from IPython import display
             buf=io.BytesIO()
             image.imsave(buf,img,**kw)
             return buf.getvalue()
```

```
In [20]: import numpy as np
         from matplotlib import image
         y,x=np.mgrid[-3:3:300j,-6:6:600j]
         z=np.sin(x**2+2*y**2+x*y)
         png=as_png(z,cmap="Blues",vmin=-2,vmax=2)
         print((repr(png[:10])))
         display.Image(png)
```

```
b'\x89PNG\r\n\x1a\n\x00\x00'
```

```
Out[20]:
```



```
In [21]: import os
         ipython=get_ipython()
         print(("HOME",os.environ["HOME"]))
         print(("IPython",ipython.ipython_dir))
         print(("PWD",ipython.config.ProfileDir.location))
```

```
HOME E:\SystemDocuments\
IPython E:\SystemDocuments\\.ipython
PWD <traitlets.config.loader.LazyConfigValue object at 0x0000024B0CAA3400>
```

```
In [23]: #import IPython.html.nbextentions as nb
         #ext="https://github.com/ipython-contrib/IPython-notebook-extensions/archive/3.x.zip"
         #nb.install_nbextension(ext,user=True)
```

```
In [24]: import numpy as np
         n=100000
         np.sum(4.0/np.r_[1:n:4,-3:-n:-4])
```

```
Out[24]: 3.14157265358979 39
```

```
In [25]: from scipy.integrate import quad
         quad(lambda x:(1-x**2)**0.5,-1,1)[0]*2
```

```
Out[25]: 3.141592653589797
```

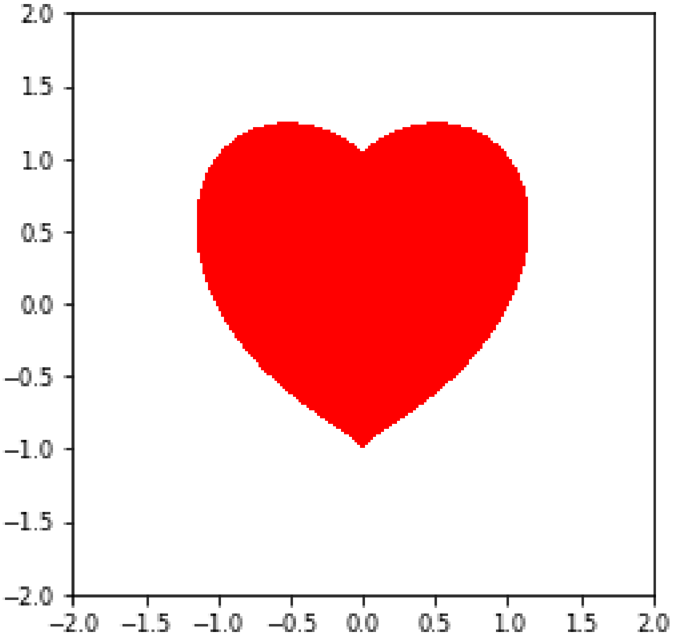
```
In [26]: from sympy import symbols,integrate,sqrt
         x=symbols("x")
         integrate(sqrt(1-x**2),(x,-1,1))*2
```

```
Out[26]: pi
```

```
In [27]: x,y=np.mgrid[-2:2:500j,-2:2:500j]
         z=(x**2+y**2-1)**3-x**2*y**3
```

```
In [28]: pl.contourf(x,y,z,levels=[-1,0],colors=['red'])
         pl.gca().set_aspect("equal")
```

```
d:\program files\python\lib\site-packages\matplotlib\font_manager.py:1297: UserWarning: findfo
  (prop.get_family(), self.defaultFamily[fontext]))
```



```
In [2]: %%mlab_plot
        print((1))
```

```
1
```

```
    ---------------------------------------------------------------------------

    IndexError                                Traceback (most recent call last)

    <ipython-input-2-2bac0694666f> in <module>()
----> 1 get_ipython().run_cell_magic('mlab_plot', '', 'print(1)')


    d:\program files\python\lib\site-packages\IPython\core\interactiveshell.py in run_cell_
   2101             magic_arg_s = self.var_expand(line, stack_depth)
   2102             with self.builtin_trap:
-> 2103                 result = fn(magic_arg_s, cell)
   2104             return result
   2105


    <decorator-gen-171> in mlab_plot(self, line, cell)


    d:\program files\python\lib\site-packages\IPython\core\magic.py in <lambda>(f, *a, **k
    185     # but it's overkill for just that one bit of state.
    186     def magic_deco(arg):
--> 187         call = lambda f, *a, **k: f(*a, **k)
    188
    189         if callable(arg):


    E:\Study\Computer\\Python\\Python\notebooks\scpy2\utils\nbmagics.py in mlab_plot(self,
    323                 ip.run_cell(cell)
    324                 from scpy2 import vtk_scene_to_array
--> 325                 img = vtk_scene_to_array(scene.scene)
    326                 return show_arrays([img])
    327             finally:


    E:\Study\Computer\\Python\\Python\notebooks\scpy2\tvtk\tvtkhelp.py in vtk_scene_to_arra
     42     arr = arr[::-1].copy()
     43     mask = np.all(arr == arr[0, 0], -1)
---> 44     x0, x1 = np.where(~np.all(mask, axis=0))[0][[0, -1]]
     45     y0, y1 = np.where(~np.all(mask, axis=1))[0][[0, -1]]
     46     if close:


    IndexError: index 0 is out of bounds for axis 0 with size 0
```

```
In [ ]: #import pandas as pd
```

```
#columns='user_id','age','sex','occupation','zip_code'
#df=pd.read_csv()
import vtk
import traits
import mayavi
```

In [ ]: