

Finding roots

Demonstration of two methods:

- `brentq()`, for a root known to lie an interval $[a, b]$, where a and b have opposite signs
- `newton()`, for a root known to be near some point a

Let's find the points where $\tan x = x$.

```
In [1]: import scipy as sp
        from scipy.optimize import brentq, newton

        import matplotlib as mpl          # As of July 2017 Bucknell computers use v. 2.x
        import matplotlib.pyplot as plt

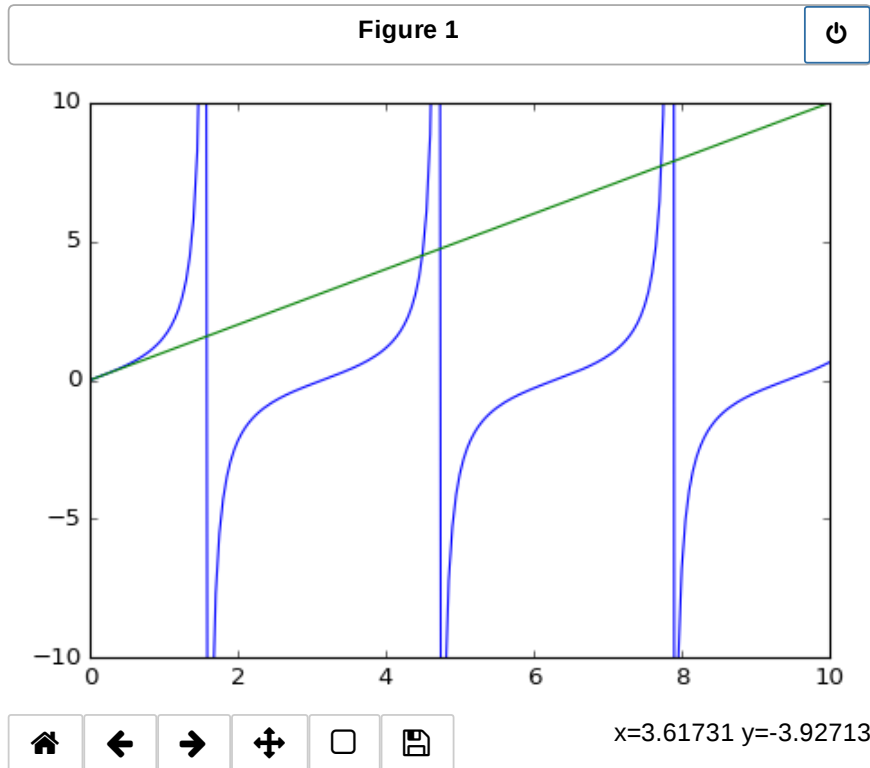
        # Following is an Ipython magic command that puts figures in the notebook.
        # For figures in separate windows, comment out following line and uncomment
        # the next line
        # Must come before defaults are changed.
        %matplotlib notebook
        #%matplotlib

        # As of Aug. 2017 reverting to 1.x defaults.
        # In 2.x text.ustex requires dvipng, texlive-latex-extra, and texlive-fonts-recommended
        # which don't seem to be universal
        # See https://stackoverflow.com/questions/38906356/error-running-matplotlib-in-latex-type
        mpl.style.use('classic')

        # M.L. modifications of matplotlib defaults using syntax of v.2.0
        # More info at http://matplotlib.org/2.0.0/users/deflt_style_changes.html
        # Changes can also be put in matplotlibrc file, or effected using mpl.rcParams[]
        plt.rc('figure', figsize = (6, 4.5))          # Reduces overall size of figures
        plt.rc('axes', labelsz=16, titlesize=14)
        plt.rc('figure', autolayout = True)          # Adjusts subplot parameters for new s.

In [2]: def f(x):
        return sp.tan(x) - x
```

```
In [3]: plt.figure(1)
x = sp.linspace(0,10,201)
y = sp.tan(x)
plt.ylim(-10,10)
plt.plot(x,y)
plt.plot(x,x);
```



```
In [4]: # Using interactive plot features helps here
root1 = sp.optimize.newton(f, 4.5)
root2 = sp.optimize.brentq(f,2,4.6)
```

```
In [5]: root1, root2
```

```
Out[5]: (4.4934094579093378, 4.4934094579089185)
```

Version Information

version_information is from J.R. Johansson (jrjohansson at gmail.com)

See Introduction to scientific computing with Python:

<http://nbviewer.jupyter.org/github/jrjohansson/scientific-python-lectures/blob/master/Lecture-0-Scientific-Computing-with-Python.ipynb> (<http://nbviewer.jupyter.org/github/jrjohansson/scientific-python-lectures/blob/master/Lecture-0-Scientific-Computing-with-Python.ipynb>)

for more information and instructions for package installation.

If version_information has been installed system wide (as it has been on linuxremotes), continue with next cell as written. If not, comment out top line in next cell and uncomment the second line.

```
In [6]: %load_ext version_information
```

```
#%install_ext http://raw.githubusercontent.com/jrjohansson/version_information/master/version_info
```

Loading extensions from ~/.ipython/extensions is deprecated. We recommend managing extensions like any other Python packages, in site-packages.

In [7]: `version_information scipy, matplotlib`

Out[7]:

Software	Version
Python	3.6.1 64bit [GCC 4.4.7 20120313 (Red Hat 4.4.7-1)]
IPython	6.1.0
OS	Linux 3.10.0 327.36.3.el7.x86_64 x86_64 with redhat 7.2 Maipo
scipy	0.19.1
matplotlib	2.0.2

Tue Aug 01 11:26:55 2017 EDT