

Solving non Linear equations

The solution to non linear equations is through matrix operations while for nonlinear they require solvers such as scipy optimize and fsolve to numerically find the solution Q1. **Solve the equation**
 $f(x) = x^2 - 5$

Solving a non linear eq involves first putting it in the form of $f(x)=0$

We know the function f and we want to find the value of x that gives $f(x)=0$

To solve this

- define the function $f(x)$
- set an initial guess for x .
- we need to import a library: from scipy.optimize import fsolve
- call the func fsolve: $x = \text{fsolve}(f, x_0)$
- f is the name of the function we are solving and x_0 is our initial guess

Function in single variables

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

%matplotlib inline

from scipy.optimize import fsolve
```

```
In [ ]: from re import X

def f(x):
    return x**2 - 5

x = fsolve(f, 2)

print(x, f(x))
```

```
[2.23606798] [-1.77635684e-15]
```

function in multiple variables

Q2. **solve the function** $x^2 + y^2 = 16$

```
In [ ]: def myf(z):
    x = z[0]
    y = z[1]

    F = np.empty((2))
    F[0] = x**2 + y**2 - 16
    return F

zguess = np.array([1,1])

z = fsolve(myf, zguess)

print(z, myf(z))
```

[1. 1.] [-14. 1.]

Exercises

solve single equations with one variable

1. $x^2 + x + 1 = 3$

2. $y + 2 * \cos(y) = 0$

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

%matplotlib inline

from scipy.optimize import fsolve
from re import X

def f(x):
    return x**2 - x - 2

x = fsolve(f,2)

print(x,f(x))

[2.] [0.]
```

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

%matplotlib inline

from scipy.optimize import fsolve
from re import X

def f(y):
    return y - 2*math.cos(y)

y = fsolve(f,2)

print(y,f(y))

[1.02986653] [0.]
```

```
In [ ]:
```