

```

# Bisection method

def bisection(f,a,b,eps,nmax):
    iterations = [a,b]
    fa = f(a)
    fb = f(b)
    if fa*fb>=0:
        print ('Bisection method fails!\n')
        print ('f(a)=%s and f(b)=%s'%(fa,fb))
        return (None,iterations[2:])
    n=0
    while (abs((b-a)/2)>eps and n<nmax):
        m = (a+b)/2
        iterations.append(m)
        fm = f(m)
        if (fm == 0 or iterations[-1]==iterations[-2]):
            return (m,iterations[2:])
        if fa*fm<0:
            b=m
            fb=fm
        elif fb*fm<0:
            a=m
            fa=fm
        n+=1
    return (m,iterations[2:])

f = lambda x: -x**5 - 9*x**4 - 28*x**3 - 35*x**2 - 15*x - 1
N=100
eps=1.0e-6
a=-3.5
b=-4

print ('f(x) =', '-(x^5+9x^4+28x^3+35x^2+15x+1)')
r = bisection(f,a,b,eps,N)
if r[0]!=None:
    print('r =',r[0], '(%d bisection iterations)'%len(r[1]))
#print (r[1])

from numpy import *

A = array([[ -2, 1, 0, 0, 0], [ 1, -2, 1, 0, 0], [ 0, 1, -2, 1, 0], [ 0, 0, 1, -2, 1], [ 0, 0, 0, 1, -
1]])
v = linalg.eig(A)[1][:,4]

```

```
A.dot(v) - linalg.eig(A)[0][4]*v
print(linalg.eig(A)[1])
print(linalg.eig(A)[0])
```

```
v = linalg.eig(A)
#print(v[0])
#print('--')
```

```
vc=[]
for i in range(5):
    vc.append(v[1][i][1])
```

```
print(linalg.eig(A)[0][0])
print(vc)
```

```
v[1]
print(A.dot(vc))
vclambda=[]
for i in range(5):
    vclambda.append(v[0][2]*vc[i])
print(vclambda)
```