In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
import math

expr = ""

def xaxis(inp):
    return 0

def func(inp):
    x=inp
    return eval(expr)


def curveplot():
    plt.rcParams["figure.figsize"] = [7.50, 3.50]
    plt.rcParams["figure.autolayout"] = True
    plt.xlabel("x")
    plt.ylabel("f(x)")
    y = np.linspace(-50, 50, 1000)
    f2 = np.vectorize(func)
    f3 = np.vectorize(xaxis)
    z=f2(y)
    plt.plot(y,z, color="blue")
    plt.plot(y, f3(y), color="green")


    plt.show()

def quadroot(r,s):
    #print("I am here")
    dis = r**2 + 4*s
    if dis>0:
        r1=(r+ math.sqrt(dis))/2
        r2=(r- math.sqrt(dis))/2
        print("Roots :","%.4f"%r1," and ","%.4f"%r2)
    else:
        r1=r/2
        r2=r/2
        i1=math.sqrt(abs(dis))/2
        i2=-i1
        print("Roots: ", "%.4f" %r1, " + i","%.4f" %i1," and ","%.4f" % r2," + i",'



def bairstow():
    coeff=[]
    b = []
    c = []
    deg=int(input("enter the degree of the polynomial"))
    i=0
    while(i<=deg):
        print("enter the coefficient a[",i,"]: ")
        coeff.append((float(input(" "))))
        b.append(0.0)
        c.append(0.0)

        i=i+1
    a=coeff
    error=float(input("enter relative percentage error"))
    N=int(input("maximum number of iterations"))
    n=deg
```

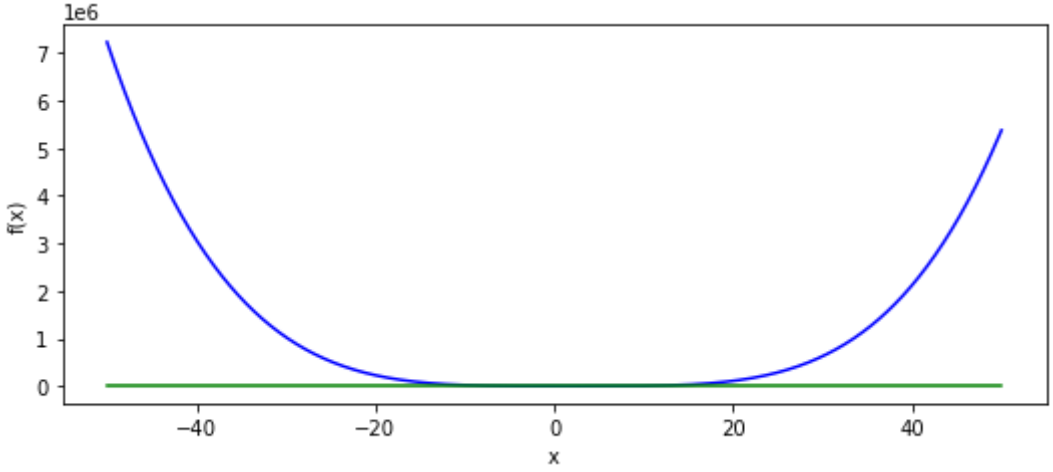
```
i=1
global expr
expr=""
while(i<=deg):
    expr = expr +"+"+str(coeff[i])+"*x**"+str(i)
    i+=1
expr=str(coeff[0])+expr
#print(expr)


while (n>=3) :
    i=0
    #if n<3 :
    #   break
    #else :
    rr = float(input("Enter initial guess for r"))
    ss = float(input("Enter initial guess for s"))
    r=rr
    s=ss
    condition = True
    while condition  :
            i+=1
            b[n]=a[n]
            b[n-1]=a[n-1] + r*b[n]
            c[n]=b[n]
            c[n-1]=b[n-1]+r*c[n]
            j=n-2
            while(j>=0):
                b[j]=a[j]+r*b[j+1]+s*b[j+2]
                c[j] = b[j] + r*c[j + 1] + s*c[j + 2]
                j-=1
            det=c[2]*c[2]-c[3]*c[1]

            if det!=0 :
                dr = ((-1)*b[1]*c[2] + b[0]*c[3])/det
                ds = ((-1)*b[0]*c[2] + b[1]*c[1])/det
                r=r+dr
                s=s+ds
            else :
                i=0
                r=r+1
                s=s+1

            if (((abs(dr/r)*100 < error) & (abs(ds/s)*100 < error)) | (i>N)) :
                condition = False
                b[n] = a[n]
                b[n - 1] = a[n - 1] + r * b[n]
                j = n - 2
                while (j >= 0):
                    b[j] = a[j] + r * b[j + 1] + s * b[j + 2]
                    j -= 1

    n-=2
    #print(b)
    quadroot(r, s)
    j=0
    while(j<=n):
        a[j]=b[j+2]
        j+=1
    #print(a)

if(n==2):
    r=-1*a[1]/a[2]
```

```
        s=-1*a[0]/a[2]
        quadroot(r,s)
    else:
        ro = (-1.0)*a[0]/a[1]
        print("Root: ","%.4f" %ro)

    curveplot()

def muller():


    coeff = []
    deg = int(input("enter the degree of the polynomial"))
    i = 0
    while (i <= deg):
        print("enter the coefficient a[", i, "]: ")
        coeff.append((float(input(" "))))
        i = i + 1

    x0 = float(input("enter the first value"))
    x2= float(input("enter the second value"))
    x1= float(input("enter the third value"))

    error = float(input("enter relative percentage error"))
    N = int(input("maximum number of iterations"))

    i = 1
    global expr
    expr=""
    while (i <= deg):
        expr = expr + "+" + str(coeff[i]) + "*x**" + str(i)
        i += 1
    expr = str(coeff[0]) + expr
    print(expr)

    i=1

    while (True):
         # Calculating various constants
         # required to calculate x3

        i=i+1
        f0 = func(x0)
        f1 = func(x1)
        f2 = func(x2)
        h0 = x1-x0
        h1 = x2-x1
        d0 = (f1-f0)/h0
        d1 = (f2-f1)/h1

        a= (d1-d0)/(h1+h0)
        b=a*h1+d1
        c=f2
        rad = math.sqrt(b*b-4*a*c)
        if abs(b+rad)>abs(b-rad):
            den=b+rad
        else:
            den=b-rad

        dxr=-2*c/den
        xr= x2+dxr
        print(abs(dxr/xr)*100, " ")
        if ( abs(dxr/xr)*100 < error or i>=N):
            break
```

```
        x0=x1
        x1=x2
        x2=xr
        print(x0, " ", x1, " ", x2, " ", dxr)

    print("The value of the root is", round(xr, 4));
    curveplot()
```

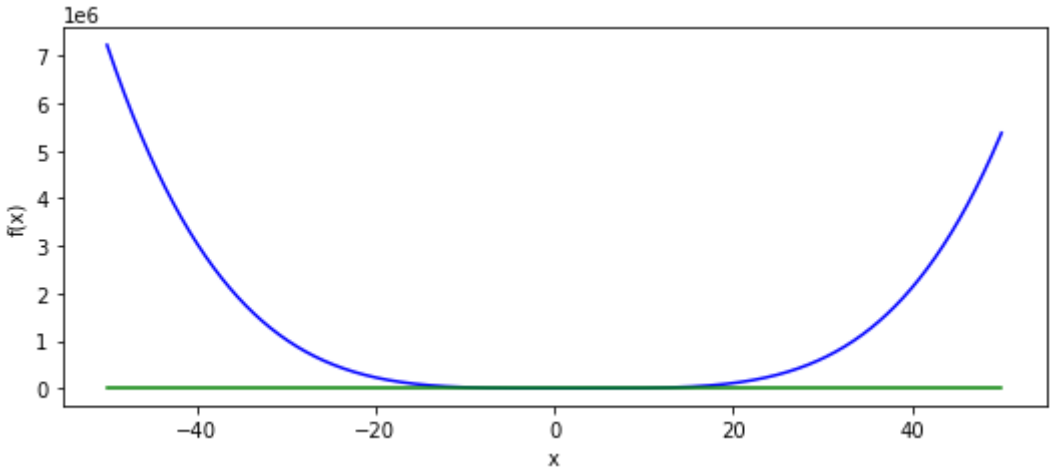
In [3]:
```
ch = int(input("Enter Choice\n1.Muller\n2.Bairstow"))
if ch==1:
    muller()
elif ch==2:
     bairstow()
else:
    print("Wrong Choice!")
```

```
Enter Choice
1.Muller
2.Bairstow1
enter the degree of the polynomial4
enter the coefficient a[ 0 ]:
 9.6448
enter the coefficient a[ 1 ]:
 -24.184
enter the coefficient a[ 2 ]:
 20.44
enter the coefficient a[ 3 ]:
 -7.4
enter the coefficient a[ 4 ]:
 1
enter the first value0
enter the second value1
enter the third value2
enter relative percentage error0.01
maximum number of iterations50
9.6448+-24.184*x**1+20.44*x**2+-7.4*x**3+1.0*x**4
10.208244913316612
2.0   1.0   0.9073731287405419   -0.09262687125945816
17.307887370109533
1.0   0.9073731287405419   0.7734971186359834   -0.13387601010455852
3.387998907590916
0.9073731287405419   0.7734971186359834   0.8006221896761415   0.02712507104015814
5
0.07752388999989888
0.7734971186359834   0.8006221896761415   0.800001997007984   -0.00062019266815746
39
0.0002496212192653354 4
The value of the root is 0.8
```



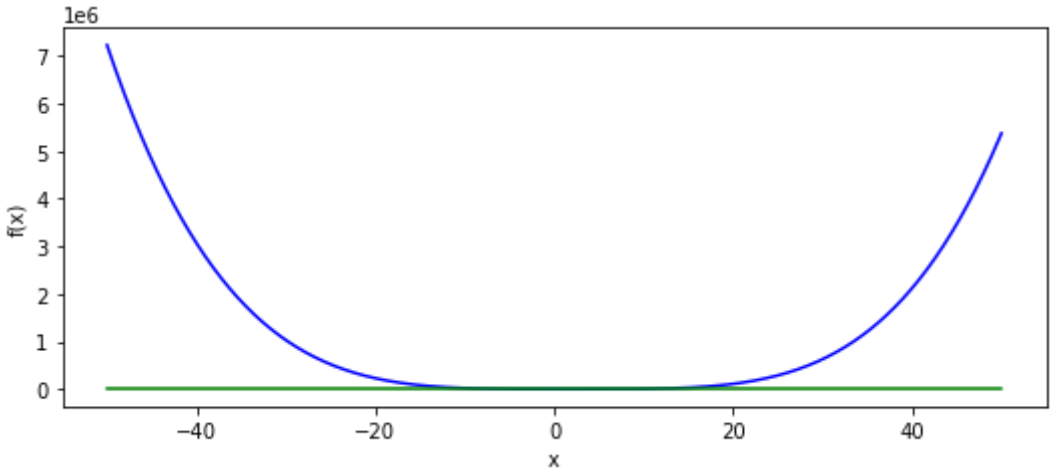
In [4]:
```
ch = int(input("Enter Choice\n1.Muller\n2.Bairstow"))
if ch==1:
    muller()
elif ch==2:
     bairstow()
else:
    print("Wrong Choice!")
```

```
Enter Choice
1.Muller
2.Bairstow2
enter the degree of the polynomial4
enter the coefficient a[ 0 ]:
 9.6448
enter the coefficient a[ 1 ]:
 -24.184
enter the coefficient a[ 2 ]:
 20.44
enter the coefficient a[ 3 ]:
 -7.4
enter the coefficient a[ 4 ]:
 1
enter relative percentage error0.01
maximum number of iterations50
Enter initial guess for r-5
Enter initial guess for s4
Roots : 2.2000  and  0.8000
Roots:  2.2000  + i 0.8000  and  2.2000  + i -0.8000
```




In [5]:
```
ch = int(input("Enter Choice\n1.Muller\n2.Bairstow"))
if ch==1:
    muller()
elif ch==2:
     bairstow()
else:
    print("Wrong Choice!")
```

```
Enter Choice
1.Muller
2.Bairstow2
enter the degree of the polynomial4
enter the coefficient a[ 0 ]:
 9.6448
enter the coefficient a[ 1 ]:
 -24.184
enter the coefficient a[ 2 ]:
 20.44
enter the coefficient a[ 3 ]:
 -7.4
enter the coefficient a[ 4 ]:
 1
enter relative percentage error0.01
maximum number of iterations50
Enter initial guess for r-2
Enter initial guess for s2
Roots : 2.2000  and  0.8000
Roots:  2.2000  + i 0.8000  and  2.2000  + i -0.8000
```




In [ ]: