

```

# -*- coding: utf-8 -*-
"""
Created on Sat Feb  2 18:21:39 2019
HW2 Bisection
@author: Ryan Dalby
"""
import numpy as np
import matplotlib.pyplot as plt

def bisection(func, a, b, iterations = 0):
    """
    Given a function and bounds(a and b) for which a root exists between will return the value of f
    """
    if(func(a)*func(b) >= 0):
        raise Exception("There is not root between a and b given(or root was given as a or b)")

    stringFormat = "{:<15}\t{:<15}\t{:<15}" #formatting string for output
    currentA = a
    currentB = b
    lastC = 0 #last iteration's root estimate
    i = 1 # loop counter
    print(stringFormat.format("Iteration", "Root estimate", "Approximate Relative Error(%)"))
    while(i <= iterations):
        c = (currentA + currentB) / 2

        #here we move either a or b to c
        if(func(a)*func(c) > 0): #if the function at a and c are the same sign then move a to c
            currentA = c
        elif(func(a)*func(c) < 0): #if the function at a and c are different signs then move b to c
            currentB = c
        else: #else the function at c is 0 and c is the root
            return c

        Ea = ((c - lastC) / c) * 100 #this is the current approximate relative error in percent
        print(stringFormat.format(i, c, Ea))
        lastC = c #set lastC for next loop
        i += 1 #set counter value for next loop

    return lastC

f = lambda x: x**3 - 13*x - 12
bisection(f, 2, 7, 5)
xVals = np.arange(-5,5, .1)
plt.plot(xVals, f(xVals))
plt.grid()
plt.title("x**3 - 13*x - 12")
plt.xlabel("x")
plt.ylabel("f(x)")
plt.show()

```