75-APROX-biseccion-newton

February 5, 2018

Método de bisección

In [1]:
```
def subint(f,a,b):
    if f(x=a)*f(x=b) > 0:
        return "ERROR: el intervalo no sirve para aplicar Bolzano"
    else:
        if f(x=a)*f(x=(a+b)/2) < 0:
            return a,(a+b)/2
        elif f(x=(a+b)/2)*f(x=b) < 0:
            return (a+b)/2,b
        elif f(x=(a+b)/2)== 0:
            return "f tiene un cero en: %s" %str((a+b)/2)
        else:
            return "ERROR"
```

In [2]:
```
f(x)=x^2+0.0000001*x-1
```

In [3]:
```
subint(f,0.0,2.0)
```

Out[3]:
```
(0.000000000000000, 1.00000000000000)
```

In [4]:
```
def iterador(f,a,b,e):
    while abs(a-b) > e:
        a,b = subint(f,a,b)
    return a,b,abs(a-b)
```

In [5]:
```
iterador(f,0.0,2.0,0.000000001)
```

Out[5]:
```
(0.999999949708581, 0.999999950639904, 9.31322574615479e-10)
```

In [6]:
```
iterador(f,0.0,2.0,0.000000000000001)
```

Out[6]:
```
(0.999999950000001, 0.999999950000002, 8.88178419700125e-16)
```

Método de Newton

In [19]:
```
f(x)=x^2+0.0000001*x-1

def newton(f,x0,epsilon,N,precision):
```

```
        f1 = diff(f,x)
        R = RealField(prec=precision)
        x0 = R(x0)
        for muda in xsrange(N):
               x0 = x0-R(f(x=x0)/f1(x=x0))
               if abs(f(x=x0))< epsilon:
                   return x0
        return "No hemos conseguido la precisión prefijada"
```

```
newton(f,2.0,10^(-20),10^20,1000)
```

Out[19]: 0.99999995000000127986634219503625919780429620659935416071865607601232284196157706901

```
In [12]: g(x)=exp(x)-17
```

```
In [20]: newton(g,2.0,10^(-20),10^20,1000)-log(17).n(prec=1000)
```

Out[20]: 3.94583165744370938381260154984375968358289028185609601208786496123323326110946376551

```
In [21]: newton(g,2.0,10^(-50),10^20,1000)-log(17).n(prec=10000)
```

Out[21]: 7.78479373444248535775416555954244964051784962879633049821278835387291746587158033104

In [ ]: