Solución Laboratorio 5
FISI 6510

Guillermo Fidalgo

18 de febrero de 2021

Exercise 4.2 Quadratic Equations

a) Write a program that takes as input three numbers, a, b, and c, and prints out the two solutions to the quadratic equation $ax^2 + bx + c = 0$ using the standard formula

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

Use your program to compute the solutions of $0.001x^2 + 1000x + 0.001 = 0$.

b) There is another way to write the solutions to a quadratic equation. Multiplying top and bottom of the solution above by $-b \mp \sqrt{b^2 - 4ac}$, show that the solutions can also be written as

$$x = \frac{2c}{-b \mp \sqrt{b^2 - 4ac}}.$$

Add further lines to your program to print these values in addition to the earlier ones and again use the program to solve $0.001x^2 + 1000x + 0.001 = 0$. What do you see? How do you explain it?

c) Using what you have learned, write a new program that calculates both roots of a quadratic equation accurately in all cases.

[1]:
```
from numpy import array
```

a)

[2]:
```
a=float(input('ingrese coeficiente "a" de la cuadrática'))
b=float(input('ingrese coeficiente "b" de la cuadrática'))
c=float(input('ingrese coeficiente "c" de la cuadrática'))
```

```
ingrese coeficiente "a" de la cuadrática0.001
ingrese coeficiente "b" de la cuadrática1000
ingrese coeficiente "c" de la cuadrática0.001
```

[3]:
```
def quadratic(a,b,c):
    pm=array([+1,-1])
    x = -b + pm*(b**2 - 4*a*c)**.5
    x= x/(2*a)
    return x
```

[4]:
```
x= quadratic(a,b,c)
print(x)
```

```
[-9.99989425e-07 -1.00000000e+06]
```

b)

[5]:
```
def quadratic2(a,b,c):
    pm=array([+1,-1])
    x = -b - pm*(b**2 - 4*a*c)**.5
    x= (2*c)/x
    return x
```

[6]:
```
x = quadratic2(a,b,c)
print(x)
```

```
[-1.00000000e-06 -1.00001058e+06]
```

Observamos que este método solamente provee la primera raíz (el número pequeño) con el valor correcto mientras que para el primer método se devuelve la segunda raíz (el número grande) con el valor correcto.

Esto se explica con que el primer método trabaja bien calculando números grandes pero el segundo método trabaja bien para calcular números pequeños. Esto es debido a errores de redondeo para cada método.

c)

[7]:
```
def quadratic_adj(a,b,c):
    x=quadratic(a,b,c)[1],quadratic2(a,b,c)[0]
    return x
```

[8]:
```
x=quadratic_adj(a,b,c)
print(x)
```

```
(-999999.999999, -1.000000000001e-06)
```