Tarea 1: Método de Raphson para una variable

IDI 2

Alumno: Ricardo De León

Descripción:

Realice código en Python que, recibiendo una función f dada, un valor inicial x_0 y una exactitud (error) dado E, encuentre una aproximación de exactitud menor a E para x cuando $f(x) = 0$ usando el método de Newton-Raphson. Asegúrese que cuenta el número de iteraciones realizadas.

Use su código para resolver los siguientes ejercicios (en todos los casos indique el(los) valor(es) inicial(es) que utilizó y el número de iteraciones que fueron necesarias para alcanzar la respuesta).

Escriba sus respuestas con **10 cifras significativas**.

Aplique el método de Newton-Raphson para encontrar ***todas*** las soluciones exactas dentro de 10^{-4} para:

1. $x^3 - 2x^2 - 5 = 0$
2. $x = \cos x$
3. $x - 0.8 = 0.2 \sin x$
4. $\ln(x-1) + \cos(x-1) = 0$
5. $e^x = 3x^2$

Encuentre una aproximación de $\sqrt{5}$ correcta con exactitud 10^{-4} usando el algoritmo de Newton-Raphson.

Encuentre el único cero negativo de $f(x) = \ln(x^2+1) - e^{0.4x}\cos(\pi x)$ con exactitud de 10^{-6} usando Newton-Raphson.

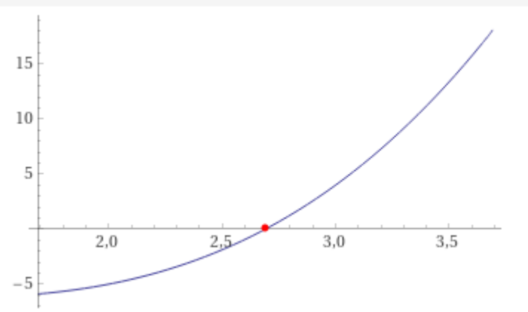
1.- $x^3 - 2x^2 - 5 = 0$

<u>Con **WolframAlpha**:</u>

Entrada

$x^3 - 2x^2 - 5 = 0$

Representación gráfica de la raíz



Solución real

$x \approx 2{,}6906$

<u>Con Python **Newton Raphson:**</u>

f(x) = x**3-2*x**2-5
inputs: tolerancia - 0.0001, X0 (inicial) 3
Iter 5, root 2.690647448

Tarea 1: Método de Raphson para una variable

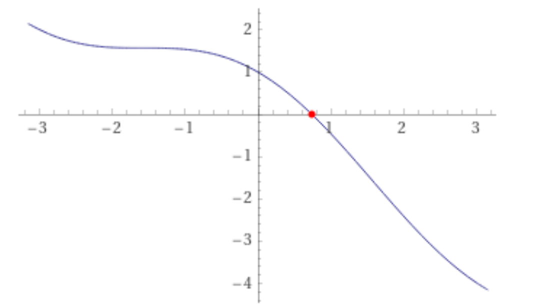
2.- $\boldsymbol{x - \cos(x) = 0}$

<u>Con **WolframAlpha**:</u>

Entrada

$\cos(x) - x = 0$

Representación gráfica de la raíz



Solución

$x \approx 0{,}739085$

<u>Con Python **Newton Raphson:**</u>

f(x) = cos(x)-x
inputs: tolerancia - 0.0001, X0 (inicial) 1
Iter 4, root 0.7390851334

Tarea 1: Método de Raphson para una variable

3.- $\boldsymbol{x - 0.8 - 0.2sen(x)}$

Con **WolframAlpha**:

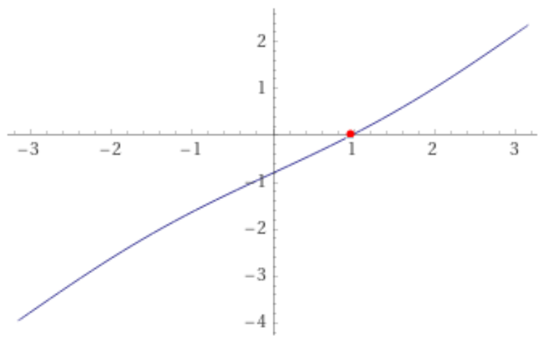
Entrada

$x - 0{,}8 - 0{,}2\,\mathrm{sen}(x) = 0$

Resultado

$-0{,}2\,\mathrm{sen}(x) + x - 0{,}8 = 0$

Representación gráfica de la raíz



Solución

$x \approx 0{,}964334$

Con Python **Newton Raphson:**

f(x) = 0.2*sin(x)-x+0.8
inputs: tolerancia - 0.0001, X0 (inicial) 1
Iter 4, root 0.9643338877

Tarea 1: Método de Raphson para una variable

4.- $\mathbf{\ln(x-1)+\cos(x-1)=0}$

Con **WolframAlpha**:

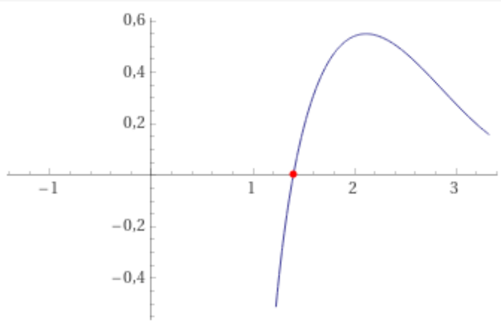
Entrada

$\log(x-1)+\cos(x-1)=0$

Resultado

$\log(x-1)+\cos(1-x)=0$

Representación gráfica de la raíz



Solución numérica

$x \approx 1{,}39774847595875\ldots$

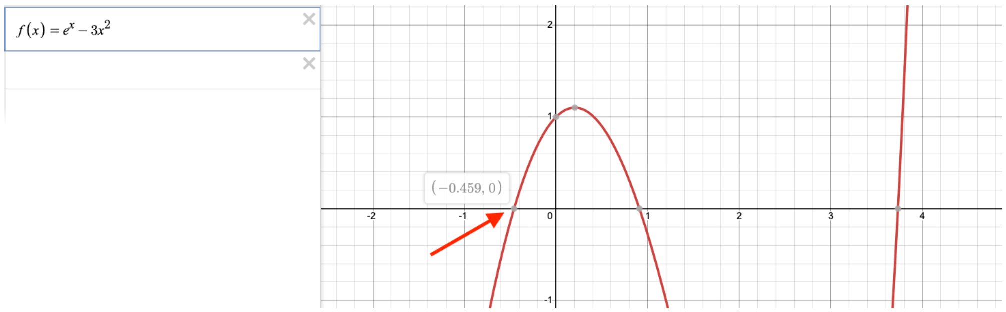
Con Python **Newton Raphson:**

f(x) = log(x-1)+cos(x-1)
inputs: tolerancia - 0.0001, X0 (inicial) 1.5
Iter 5, root 1.397748476

Tarea 1: Método de Raphson para una variable

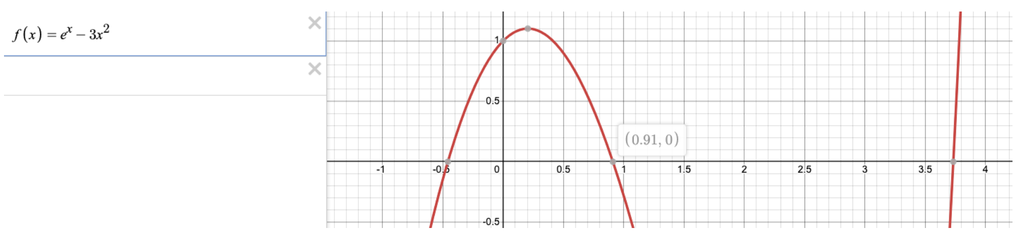
5.- $\boldsymbol{e^x - 3x^2 = 0}$

Con Desmos



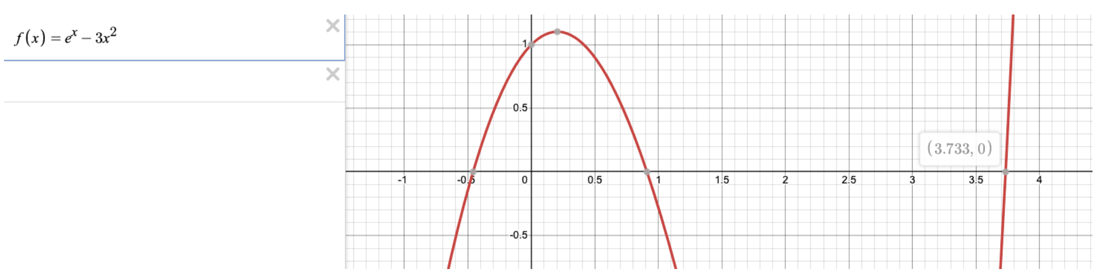
Con Python **Newton Raphson:**

f(x) = exp(x)-3*x**2
inputs: tolerancia - 0.0001, X0 (inicial) 0
Iter 6, root1 -0.4589622742



Con Python **Newton Raphson:**

f(x) = exp(x)-3*x**2
inputs: tolerancia - 0.0001, X0 (inicial) 1
Iter 4, root2 0.9100075725

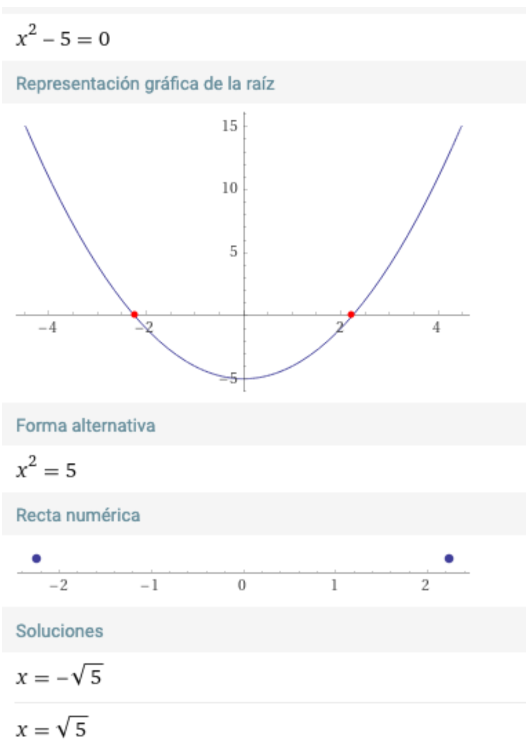


Con Python **Newton Raphson:**

f(x) = exp(x)-3*x**2
inputs: tolerancia - 0.0001, X0 (inicial) 4
Iter 5, root3 3.733079029

Tarea 1: Método de Raphson para una variable

- Con **WolframAlpha** para:



Con Python **Newton Raphson:**

f(x) = x**2-5
inputs: tolerancia - 1e-06, X0 (inicial) -2
Iter 5, root1 -2.236067977

f(x) = x**2-5
inputs: tolerancia - 1e-06, X0 (inicial) 2
Iter 5, root2 2.236067977

Tarea 1: Método de Raphson para una variable

- Con **WolframAlpha** para:

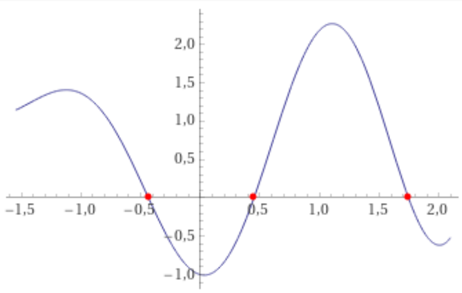
Entrada

$\log(x^2+1) - \exp(x \times 0{,}4)\cos(\pi x) = 0$

Resultado

$\log(x^2+1) - e^{0{,}4x}\cos(\pi x) = 0$

Representación gráfica de la raíz



Solución numérica

$x \approx -0{,}434143047285729\ldots$

$x \approx 0{,}450656747889936\ldots$

$x \approx 1{,}74473805336883\ldots$

$x \approx 2{,}23831979507414\ldots$

$x \approx 3{,}70904120137595\ldots$

$x \approx 4{,}32264895939424\ldots$

$x \approx 5{,}61993533089735\ldots$

$x \approx 6{,}40693361417946\ldots$

Con Python **Newton Raphson:**

f(x) = log(x**2+1)-exp(x*0.4)*cos(pi*x)
inputs: tolerancia - 1e-06, X0 (inicial) -0.01
Iter 6, root -0.4341430473

Tarea 1: Método de Raphson para una variable

Código:

Código para los primeros 6 ejercicios, el 7mo cambia la tolerancia a 10^{-6}

```python
import sympy as sp
from math import *
from sigfig import round

tolerancia = 0.0001

def newton_raphson(x0):
    x = sp.symbols('x')
    f = input('Enter the function with variable x: ')
    print(f'f(x) = {f}')
    df = sp.diff(f)
    f = sp.lambdify(x, f)
    df = sp.lambdify(x, df)
    tramo = abs(2 * tolerancia)
    iter_count = 1;
    while not(tramo <= tolerancia):
        x1=x0-f(x0)/df(x0)
        tramo = abs(x1-x0)
        x0 = x1
        iter_count += 1
    print(f'inputs: tolerancia - {tolerancia}, X0 (inicial) - {x0}')
    print(f'# Iter {iter_count}, root {round(x1, sigfigs=10)}')

newton_raphson(x0)
```

En donde **x0** es el valor inicial en la posición en el eje de las ‘x’.