

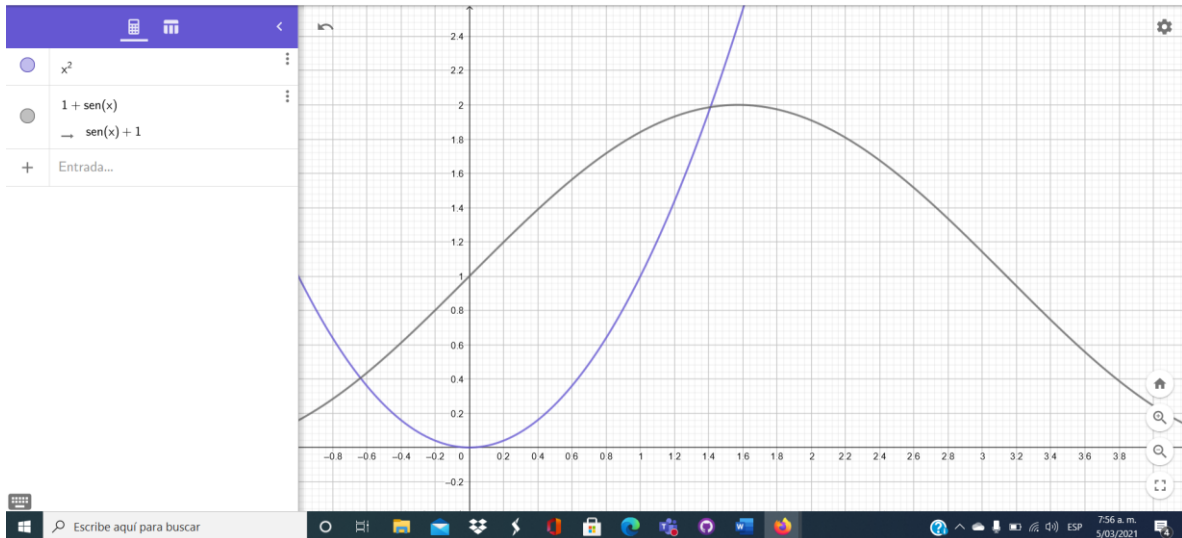
CRISTIAN CAMILO BENITEZ PEÑA – PARCIAL 1.

Punto 1.

2. Para cada uno de los siguientes ejercicios: utilice el algoritmo señalado para encontrar la intersección entre $f(x) = x^2$ y $g(x) = 1 + \sin x$, en el intervalo $[1, 2]$ con $E < 10^{-16}$, determinar el número de iteraciones realizadas, una grafica que evidencie el tipo de convergencia del método, debe expresarla en notación $O()$

a)

$$x_n = x_{n-1} - \frac{f(x_{n-1})(x_{n-1} - x_{n-2})}{f(x_{n-1}) - f(x_{n-2})} \quad (1)$$



Secant Method is open method and starts with two initial guesses for finding real root of non-linear equations.

In Secant method if x_0 and x_1 are initial guesses then next approximated root is obtained by following formula: x_0 x_1 x_2

$$x_2 = x_1 - (x_1 - x_0) * f(x_1) / (f(x_1) - f(x_0))$$

And an algorithm for Secant method involves repetition of above process i.e. we use x_1 and x_2 to find x_3 and so on until we find the root within desired accuracy. x_1 x_2 x_3

The image shows the Spyder Python IDE interface. The left pane displays a Python script named 'punto 1-CristianBenitez.py'. The script implements a secant method for finding the root of a function $f(x) = \sin(x) + 1$. The right pane shows the console output, which lists iterations from 3 to 37, showing the values of x_2 and $f(x_2)$ converging towards zero. The final output is 'Raiz aprox.: -1.57079633'.

```
#Cristian Camilo Benitez
from math import *

# Defining Function
def f(x):
    return sin(x)+1

# Implementing Secant Method
def secant(x0,x1,e,N):
    print('\n\n*** Metodo de la secante ***')
    step = 1
    condition = True
    while condition:
        if f(x0) == f(x1):
            print('Divide by zero error!')
            break
        x2 = x0 - (x1-x0)*f(x0)/( f(x1) - f(x0) )
        print('Iteracion-%d, x2 = %.6f y f(x2) = %.6f' % (step, x2, f(x2)))
        x0 = x1
        x1 = x2
        step = step + 1

        if step > N:
            print('No Convergente!')
            break
        condition = abs(f(x2)) > e
    print('\n Raiz aprox.: %.8f' % x2)

x0 = input('Primer valor: ')
x1 = input('Segundo valor: ')
N = input('Max iteraciones: ')

x0 = float(x0)
x1 = float(x1)
e = 10**-16
N = int(N)
```

Iteration-3, x2 = -1.444322 and f(x2) = 0.007987
Iteration-4, x2 = -1.499514 and f(x2) = 0.003221
Iteration-5, x2 = -1.521729 and f(x2) = 0.001204
Iteration-6, x2 = -1.540352 and f(x2) = 0.000463
Iteration-7, x2 = -1.552011 and f(x2) = 0.000176
Iteration-8, x2 = -1.559188 and f(x2) = 0.000067
Iteration-9, x2 = -1.563619 and f(x2) = 0.000026
Iteration-10, x2 = -1.566360 and f(x2) = 0.000010
Iteration-11, x2 = -1.568055 and f(x2) = 0.000004
Iteration-12, x2 = -1.569102 and f(x2) = 0.000001
Iteration-13, x2 = -1.569749 and f(x2) = 0.000001
Iteration-14, x2 = -1.570149 and f(x2) = 0.000000
Iteration-15, x2 = -1.570396 and f(x2) = 0.000000
Iteration-16, x2 = -1.570549 and f(x2) = 0.000000
Iteration-17, x2 = -1.570644 and f(x2) = 0.000000
Iteration-18, x2 = -1.570702 and f(x2) = 0.000000
Iteration-19, x2 = -1.570738 and f(x2) = 0.000000
Iteration-20, x2 = -1.570768 and f(x2) = 0.000000
Iteration-21, x2 = -1.570774 and f(x2) = 0.000000
Iteration-22, x2 = -1.570783 and f(x2) = 0.000000
Iteration-23, x2 = -1.570788 and f(x2) = 0.000000
Iteration-24, x2 = -1.570791 and f(x2) = 0.000000
Iteration-25, x2 = -1.570793 and f(x2) = 0.000000
Iteration-26, x2 = -1.570794 and f(x2) = 0.000000
Iteration-27, x2 = -1.570795 and f(x2) = 0.000000
Iteration-28, x2 = -1.570796 and f(x2) = 0.000000
Iteration-29, x2 = -1.570796 and f(x2) = 0.000000
Iteration-30, x2 = -1.570796 and f(x2) = 0.000000
Iteration-31, x2 = -1.570796 and f(x2) = 0.000000
Iteration-32, x2 = -1.570796 and f(x2) = 0.000000
Iteration-33, x2 = -1.570796 and f(x2) = 0.000000
Iteration-34, x2 = -1.570796 and f(x2) = 0.000000
Iteration-35, x2 = -1.570796 and f(x2) = 0.000000
Iteration-36, x2 = -1.570796 and f(x2) = 0.000000
Iteration-37, x2 = -1.570796 and f(x2) = 0.000000
Raiz aprox.: -1.57079633

Punto 2.

c) $f(x) = \ln(1 + 2x)$ en $[-0.5, 0.5]$ para $x = 0.005, 0.0001, 0.499999999$

Ir

Ocultar definición

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \dots$$


4. Considere la función $f(x) = \sqrt{x^2 + 1} - 1$ para valores $x_0 = 1.1, 1.111111111, 1.11111111111111$ determine una forma de evaluar la función en los valores indicados, utilizando una precisión doble y que no se presente pérdida de significancia. (debe imprimir las soluciones que muestren la pérdida de significancia y la solución a este problema)

Metodo de Newton-Raphson x

Calculadora CAS - GeoGebra x

aylor ln(1+2x), 0 - Calculad... x

Polinomio de Taylor - GeoGel... x

sqrt(x^2+1)-1 - Wolfram|Alpha x

Tecorema de taylor wolfram... x

https://www.wolframalpha.com/input/?i=sqrt(x^2%2B1)-1

Comenzar a usar Firefox

LastPass password ma...

Express VPN

Blackboard Learn

Correo: CRISTIAN CA...

YouTube

LastPass password ma...

Express VPN

Blackboard Learn

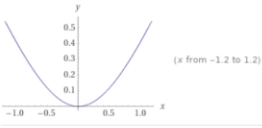
Correo: CRISTIAN CA...

YouTube


Input:

$$\sqrt{x^2 + 1} - 1$$

Plots:



(x from -1.2 to 1.2)



(x from -8.2 to 8.2)

Root:

$x = 0$

Properties as a real function:

Domain

\mathbf{R} (all real numbers)


Range


$\{y \in \mathbf{R} : y \geq 0\}$ (all non-negative real numbers)


DISCOVER
WHAT'S
POSSIBLE
with Wolfram|Alpha


Take the Tour

Step-by-Step
Solutions for...

 Calculus

 x^2-1 Algebra


 Trigonometry

 Equation Solving

This website uses cookies to optimize your experience with our services on the site, as described in our Privacy Policy.

Accept & Close

Escribe aquí para buscar



9:39 a.m.
5/03/2021

ESP