Universidad Autónoma de Baja California

Facultad de Ciencias Químicas e Ingeniería



**METODOS NUMERICOS**

**Regla de Simpson**

**Docente:** Itzel Barriba Cazares

**Alumno:** Gómez Cárdenas Emmanuel Alberto

Matrícula: 1261509

# Descripción de la practica

Aplicar la regla de Simpson 1/3 y 3/8 para la Integración numérica, mediante los recursos tecnológicos, identificando los elementos y criterios y ventajas de este, para resolver situaciones problemáticas de ingeniería en donde se requiera la determinación del área bajo la curva, en forma creativa y responsable.

# Código

# GOMEZ CÁRDENAS EMMANUEL ALBERTO          01261509

# MÉTODOS NUMÉRICOS                   07/MAYO/2020

# MÉTODO DEL TRAPECIO SIMPLE Y COMPUESTO

# Aplicación de la regla de Simpson 1/3 y 3/8 para

# la integración numérica, utilizando la función que

# que el usuario ha escogido

#Capturar entero

def getInt(str):

  return int(input(str + ": "))

#Capturar flotante

def getFloat(str):

  return float(input(str + ": "))

#Evalúa la ecuación proporcionada en el número deseado

def evaluateFunction(equation, number):

  res = 0

  for j in range(len(equation)):

    res += (equation[j] \* (number \*\* (j)))

  return res

#Regla de simpson 1/3 de una sola aplicación

def Simp13 (start, end, eq):

    h = (end-start)/2

    return float( (end-start)/6 \* ( evaluateFunction(eq, start) + 4\*evaluateFunction(eq, h) + evaluateFunction (eq, end) ))

#Regla de simpson 1/3 de aplicación múltiple

def Simp13m (start, end, n, eq): #Si el numero de segmentos es impar, se redondea al par siguiente

  if (n%2 > 0): #La regla de simpson 1/3 solo se debe aplicar cuando número de segmentos es par

    n += 1

  h = (end-start)/n

  xi = start

  sum = evaluateFunction(eq, xi)

  for i in range (1, n):

    if( (i%2) ==  0 ):

      xi += h

      sum += 2\*evaluateFunction(eq, xi)

    else:

      xi += h

      sum += 4\*evaluateFunction(eq, xi)

  sum += evaluateFunction(eq, end)

  return (h\*sum/3)

#Regla de simpson 3/8 de una sola aplicación

def Simp38 (start, end, eq):

    h = (end-start)/3

    return float( (h\*3/8)\*(evaluateFunction(eq, start) + 3\* ( evaluateFunction(eq, start + h) + evaluateFunction(eq, end-h) ) + evaluateFunction(eq, end) ) )

#Método del trapecio simple

def SingleTrap(start, end, eq):

    return (end-start)\*((evaluateFunction(eq, end) + evaluateFunction(eq, start))/2)

#Regla de simpson para un número de segmentos, ya sea pares o impares

def SimpInt (start, end, n, eq):

  sum = 0

  h = (start - end) / n

  if (n == 1):

    sum = SingleTrap(start, end, eq)

  else:

    m = n

    if ((n / 2 - int(n/2)) > 0):

      odd = False

    else:

      odd = True

    if ( (odd == False) & (n>1) ):

      sum += Simp38(start, end, eq)

      m = n - 3

    if ( m > 1 ):

      sum += Simp13m(start, end, m, eq)

    if (odd == False):

      return sum/2

  return sum

#función main

g = 1 + getInt("Introduzca numéricamente el grado de su ecuación")

equation = []

for i in range(g):

  equation.append(getFloat("Coeficiente del termino x^" + str(i)))

start = getInt("Valor inicial (a)")

end = getInt("Valor final (b)")

n = getInt("Número de segmentos (n)")

if (start >= end):

    print("El valor final debe ser más grande que el inicial...")

    exit(1)

print("\n\t¿Que método desea utilizar?")

print("\n\t1) Simpson 1/3 de una sola aplicación \n\t2) Simpson 1/3 aplicación múltiple")

print("\t3) Simpson 3/8 de una sola aplicación  \n\t4) Simpson aplicación multiple para un numero de segmentos tanto pares como impares")

ans = getInt("Seleccione una opción")

if (ans == 1):

    print("Aproximacion utilizando la regla de Simpson 1/3 para una sola aplicación: " + str(Simp13(start, end, equation)) )

elif (ans == 2):

    if(n<2):

        print("Para utilizar la regla de Simspon 1/3 de aplicación múltiple, debe de haber al menos dos segmentos")

        quit(1)

    else:

        print("Aproximación utilizando la regla de Simpson 1/3 de aplicación múltiple: " + str(Simp13m(start, end, n, equation)))

elif (ans == 3):

    print("Aproximación utilizando la regla de Simpson 3/8 para una sola aplicación: " + str(Simp38(start, end, equation)))

elif (ans == 4):

    print("Aproximación utilizando la regla de simpson de aplicación múltiple para un número de segmentos tanto pares como impares: " + str(SimpInt(start, end, n, equation)))

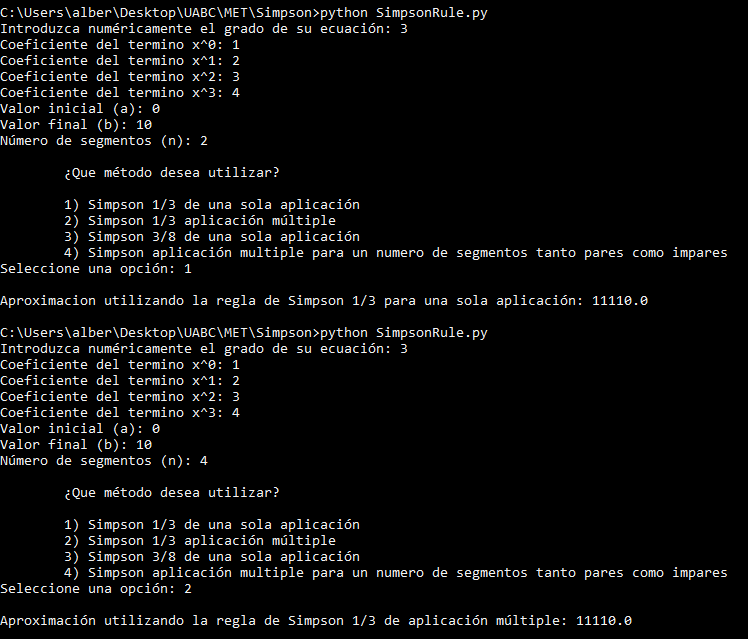
else:

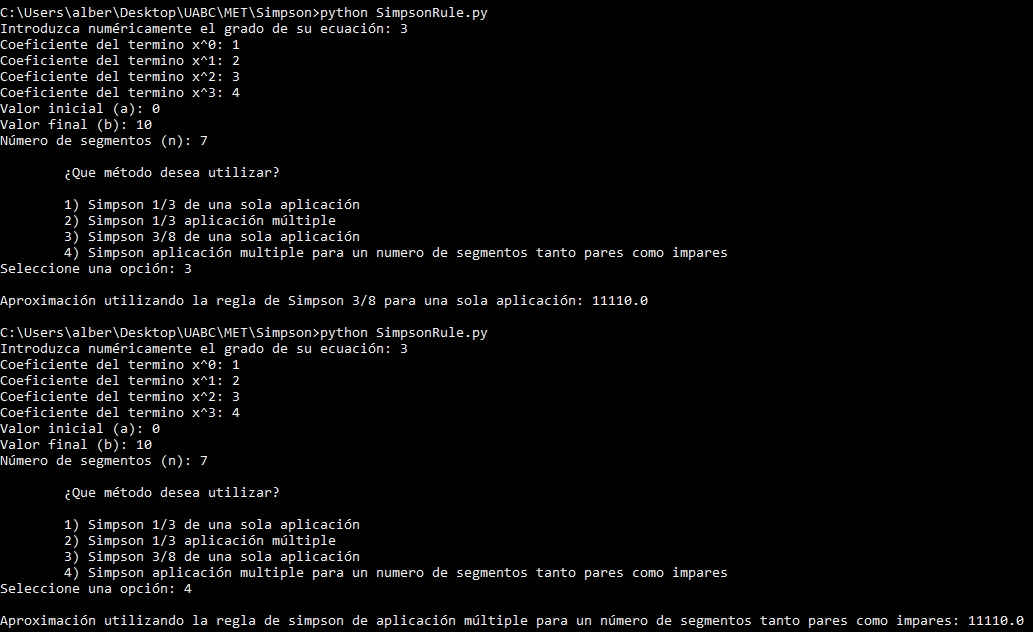
    print("\nPor favor, seleccione una opción válida")

    quit(1)

# Pruebas de funcionamiento

Integral para aproximar:





# Resultados y conclusiones

La regla de Simpson y sus variantes pueden ser muy útiles a la hora de aproximar áreas bajo la curva, sin embargo, es de suma importancia el método a utilizar ya que el porcentaje de error difiere entre cada uno. Es muy interesante como el método calcula el área por medio de solapamientos de curvas y como es de suma importancia que el numero de segmentos sea par, ya que también juega un papel a la hora de hacer el cálculo del error.