Taller 1 Análisis numérico

- Angela Sofia Moreno

  https://github.com/sofiamoreno199/AnalisisNumerico

- Anggie Carolina Correa Sánchez

https://github.com/AnggieCorrea/Analisis-Numerico

Problemas:

1) Suponga que un dispositivo solo puede almacenar únicamente los cuatro primeros dígitos decimales de cada número real, y trunca los restantes (esto es redondeo inferior). Calcule el error de redondeo si se quiere almacenar el número 536.78

   Contextualización: los métodos numéricos operan con datos que pueden ser inexactos para representar a los números reales. El error de redondeo se atribuye a la imposibilidad de almacenar todas las cifras de estos números y a la imprecisión de los instrumentos de medición con los cuales se obtienen los datos. Para el presente problema, se contempla el escenario de un dispositivo que solo puede almacenar los cuatro primeros dígitos decimales de cada número real, y trunca los restantes.

   Entradas: 536.78. Número a truncar.

   Salidas: 0.08. Error de truncamiento

   Análisis:

   Si se normaliza la entrada a 0.53678×103 esto da 5 cifras decimales. Para satisfacer el problema del dispositivo, se descompone el número de la siguiente forma: $0.5367 \times 103 + 0.00008 \times 103$ y 0.00008×103 1 equivale a 0.08.

2) Implemente en cualquier lenguaje el siguiente algoritmo que sirve para calcular la raíz cuadrada. Aplíquelo para evaluar la raíz cuadrada de 7, analice su precisión, como podría evaluar la convergencia y validez del algoritmo.

   Contextualización: En el campo de la matemática, la raíz cuadrada se identifica como el número que, al ser multiplicado una vez por si mismo, da como resultado un primer número. Para calcular la raíz en el ámbito de la programación se utiliza un método iterativo que produce un valor más cercano a la respuesta

Entradas:

n = 7. Dato (número que va a ser evaluado).

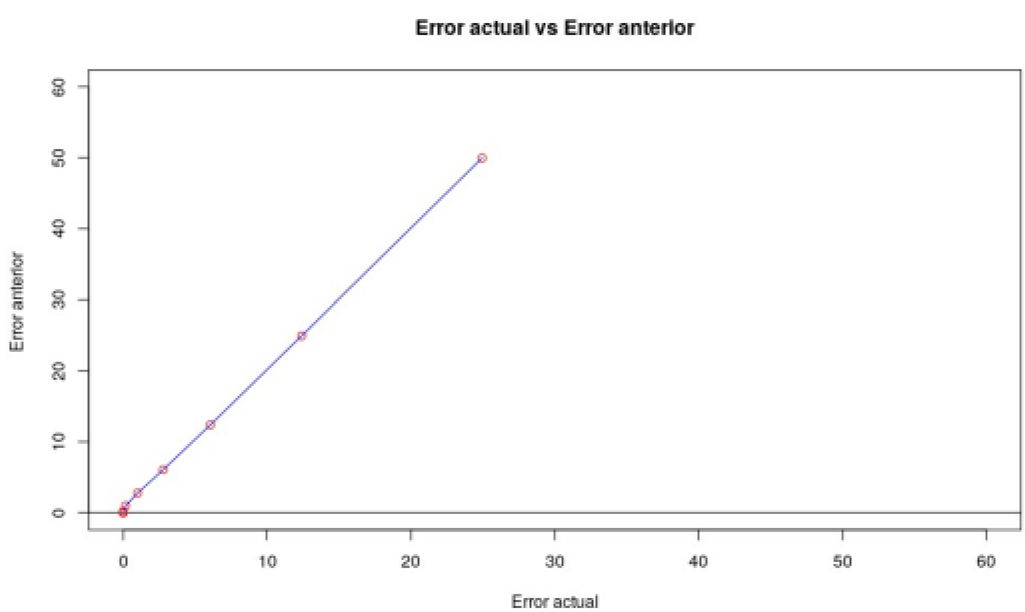
e = $1 \times 10-8$. Valor de tolerancia, es decir, el error permitido.

x = 100. Valor inicial.

Salidas: Valor redondeado = 2.645751.

Análisis:

Error Actual	Error Anterior
24.9475490	49.9650000
12.4042135	24.9475490
6.0656640	12.4042135
2.7798920	6.0656640
1.0068318	2.7798920
0.1790470	1.0068318
0.0060445	0.1790470
0.0000069	0.0060445
0.0000000	0.0000069



- Análisis de precisión: la formula converge, pero el resultado final no es la raíz cuadrada de 7, pues la precisión es insuficiente. El valor resultante es 2.645751, pero como bien se sabe es una aproximación deficiente pues el resultado es un número no exacto, lo que implica un sin fin de números decimales que lo conforman.
- Evaluación de la convergencia: como se puede apreciar en la figura 2, el algoritmo iterativo utilizado tiene una convergencia lineal. Los métodos iterativos tienen la propiedad de producir resultados cada vez más cercanos a la respuesta esperada.
- Validez del algoritmo: La poca cantidad de cifras obtenidas hace que el método sea deficiente, pero valido. Eso quiere decir que se acerca a los dígitos más significativos del valor real. La precisión insuficiente plantea la importancia de verificar la formulación del método numérico y la validación de la respuesta obtenida.

3) Utilizando el teorema de Taylor hallar la aproximación de e 0 .5 con cinco cifras significativas

Contextualización: A lo largo de la historia, se ha usado el teorema de Taylor con el fin de realizar aproximaciones de diferentes tipos de funciones alrededor de un punto, en el cual, dicha función es diferenciable. Este teorema genera un resultado bastante aproximado al resultado real, y entre mayor es el orden del polinomio, más aproximado es el resultado.

Entradas:

f = e x . Función a la cual se le realizar á la aproximación.

x0 = 0.5. Punto en el que se evalúa la función.

a = 1. Valor alrededor de x.

n = 6. Orden del polinomio.

Salidas: Resultado = 1.6487.

4) Calcule el tamaño del error dado por las operaciones aritméticas, para la solución del siguiente problema

La velocidad de una partícula es constante e igual a 4 m/s, medida con un error de 0.1 m/s durante un tiempo recorrido de 5 seg. medido con error de 0.1 seg. Determine el error absoluto y el error relativo en el valor de la distancia recorrida.

v = 4m/s.

Ev = 0.1m/s.

t = 5s.

Et = 0.1s.

d = vt

Contextualización: En los métodos directos debe considerarse el error que se propaga en las operaciones aritméticas, el cual puede ser significativo cuando la cantidad de cálculos requeridos es grande. Para este problema en particular, se desarrolló un problema de dinámica elemental en el cual posee las siguientes características.

Entradas:

v = 4m/s. Velocidad.

Ev = 0.1m/s. Error de medición de la velocidad.

4 t = 5s. Tiempo de recorrido.

Et = 0.1s. Error de medición del tiempo.

Salidas:

d = 20m. Distancia recorrida.

ea = 0.9. Tamaño del error dado por las operaciones aritméticas.

El intervalo resultante es [19.1 , 20.9].

er = 4.5 % . Porcentaje de error.

5) Evaluar el valor de un polinomio es una tarea que involucra para la máquina realizar un número de operaciones la cual debe ser mínimas. Como se puede evaluar el siguiente polinomio con el número mínimo de multiplicaciones.

P(x) = 2x 4 − 3x 2 + 3x − 4, x0 = − 2

Contextualización: Para éste ejercicio se usa el método de Horner, este consiste en aplicar un algoritmo que permita calcular el resultado de un polinomio evaluado en un valor específico de x. Además de ello, el algoritmo consigue hacer su labor con la cantidad mínima de operaciones posible, lo cual lo convierte en un método muy eficiente.

Entradas:

coeficiente = 2, 0, -3, 3, -4. Vector que contiene los coeficientes del polinomio.

x0 = − 2. Valor a ser evaluado en el polinomio.

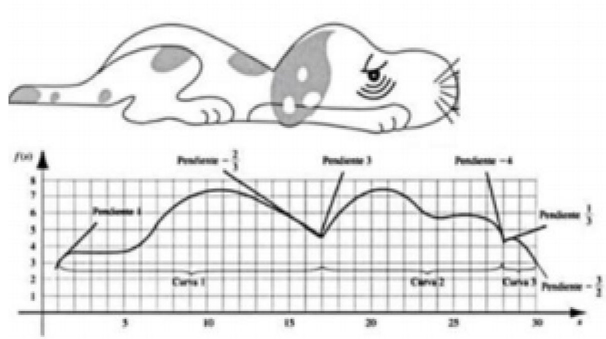
Salidas:

Resultado = 10.

El número mínimo de operaciones es 8, y se compone de 4 sumas y 4 multiplicaciones.

6) Reconstruir la silueta del perrito utilizando la menor cantidad de puntos para reproducir el dibujo del contorno completo del perrito sin bigotes, con la información dada:

Coordenadas:
y=c(3,3.7,3.9,4.5,5.7,6.69,7.12,6.7,4.45,7,6.1,5.6,5.87,5.15,4.1,4.3,4.1,3) x=c(1,2,5,6,7.5,8.1,10,13,1



Contextualización: En el mundo real existen muchas investigaciones en diferentes campos que necesitan de análisis estadísticos, para evaluar grandes cantidades de datos recolectados, como por ejemplo en los estudios de variación climática, los sensores ubicados en puntos estratégicos recolectan por día cantidades suficientes de datos que permitan estudiar el comportamiento actual del clima y hacer predicciones del mismo. En este caso se requiere encontrar la silueta de un perro a partir de unos datos dados a demás, se necesita completar los datos con la cantidad mínima pero suficiente para resolver el problema.

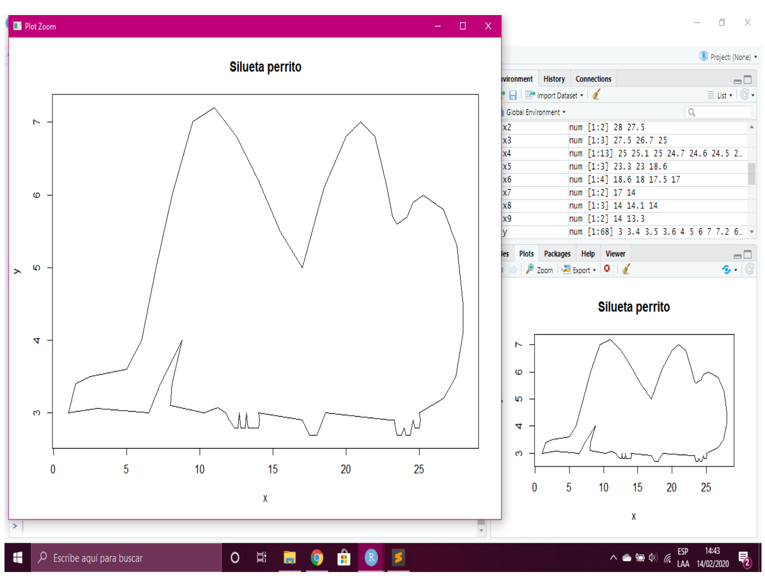
Entradas: vectores:

x = c (01.00 , 01.50 , 02.50 , 05.00 , 06.00 , 07.00 , 08.10 , 09.50 , 11.00 , 12.50 , 14.00 , 15.50 , 17.00 , 18.50 , 20.00 , 21.00 , 22.00 , 22.80, 23.20 , 23.50 , 24.20 , 24.60 , 25.30 , 26.70 , 27.60 , 28.00 , 28.00 , 27.50 , 26.70 , 25.00 , 25.10 , 25.00 , 24.70 , 24.60 , 24.50 , 24.40 , 24.10 , 24.00 , 23.80 , 23.50 , 23.40 , 23.30 , 23.00 , 18.60 , 18.00 , 17.50 , 17.00 , 14.00 , 14.10 , 14.00 , 13.30 , 13.20 , 13.10 , 12.80 , 12.70 , 12.60 , 12.40 , 12.00 , 11.80 , 11.20 , 10.30 , 08.00 , 08.10 , 08.80 , 07.30 , 06.50 , 03.00 , 01.00)

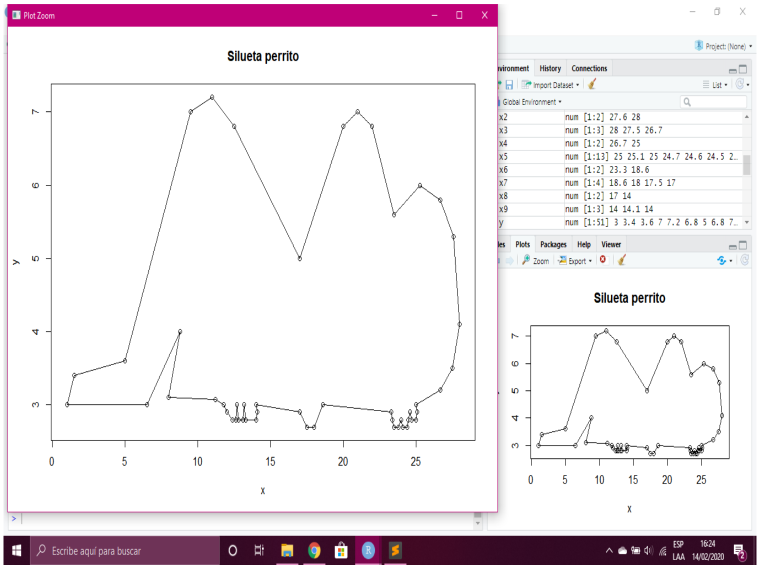
y = c (03.00 , 03.40 , 03.50 , 03.60 , 04.00 , 05.00 , 06.00 , 07.00 , 07.20 , 06.80 , 06.20 , 05.50 , 05.00 , 06.10 , 06.80 , 07.00 , 06.80 , 06.10 , 05.70 , 05.60 , 05.70 , 05.90 , 06.00 , 05.80 , 05.30 , 04.50 , 04.10 , 03.50 , 03.20 , 03.00 , 02.90 , 02.80 , 02.80 , 02.90 , 02.80 , 02.70 , 02.70 , 02.80 , 02.70 , 02.70 , 02.80 , 02.90 , 02.90 , 03.00 , 02.70 , 02.70 , 02.90 , 03.00 , 02.90 , 02.80 , 02.80 , 03.00 , 02.80 , 02.80 , 03.00 , 02.80 , 02.80 , 02.90 , 03.00 , 03.07 , 03.00 , 03.10 , 03.40 , 04.00 , 03.40 , 03.00 , 03.06 , 03.00)

Salidas:

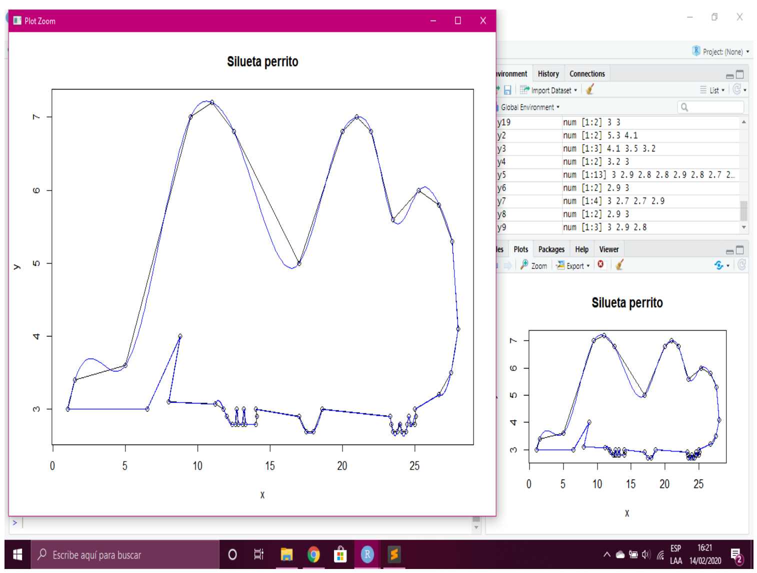
Resultado con puntos requeridos para buena distinción de la silueta:



Resultado con mínima cantidad de puntos:



Resultado de la interpolación:



Análisis:

Primero se intentó ubicar cada uno de los puntos requeridos para poder dar forma completa a la silueta del perrito dando como resultado 68 puntos, después se intentó disminuir la máxima cantidad de puntos sin dañar la figura del perro, a cuyo resultado encontramos 51 puntos, los cuales se subdividen en pequeños vectores para poder hacer la interpolación de la que podemos concluir los subintervalos no podían ser de igual longitud, ya que la función resultante en cada concentración de puntos siempre es distinta, y no se puede representar con una sola función para todos los puntos.

Ejercicios parte 1:

1) Número de operaciones

1.1 Teorema de Horner:

Contextualización:El método de Horner consiste en aplicar un algoritmo que permita calcular el resultado de un polinomio evaluado en un valor específico de x. Además de ello, el algoritmo consigue hacer su labor con la cantidad mínima de operaciones posible, lo cual lo convierte en un método muy eficiente, ya que reduce la cantidad de tareas que debe ejecutar el procesador de un dispositivo para obtener la solución del polinomio._

1) **Utilice el método de inducción matemática para demostrar el resultado del método 2**

$$\text{Sea P0(x)} = \text{a0x 0} = \text{a0}.$$

El número de multiplicaciones para hallar P0(x0) es igual a 0. Por lo que se cumple para el primer caso k = 0.

Se asume por lo tanto que Pk(x) = a0 + a1x + ... + akx k y que Pk(x0) = a0 + a1x0 + ... + akx k 0 tiene k multiplicaciones y que el método de Horner se expresa de la siguiente manera:

$$1 \text{ bk} = \text{ak bk}- 1 = \text{ak}- 1 + \text{bk} * \text{x0} \ldots \text{b0} = \text{a0} + \text{b1} * \text{x0}$$

Y se debe llegar a la forma k + 1 del polinomio:

$$\text{Pk+1(x0)} = \text{a0} + \text{a1x0} + \ldots + \text{ak+1x k+1 0} \; (4)$$

o reescrito de otra forma:

$$\text{Pk+1(x0)} = \text{a0} + \text{x(a1} + \text{x(a2} + \ldots + \text{x(ak} + \text{xak+1)))} \; (5)$$

Se reescribe Pk(x0) y se reemplaza ak por bk (Primera instrucción del método):

$$\text{Pk(x0)} = \text{a0} + \text{x0(a1} + \ldots + \text{x0(ak}- 1 + \text{x0} * \text{bk))} \; (6)$$

Se añade una iteración al método de Horner para bk+1, añadiendo una multiplicación más al método (multk+1 = k + 1) bk+1 = ak+1 bk = ak + bk+1 ∗ x0 ... b0 = a0 + b1 ∗ x0 Y se reemplaza bk en Pk(x0):

$$\text{Pk(x0)} = \text{a0} + \text{x0(a1} + \ldots + \text{x0(ak}- 1 + \text{x0} * \text{(ak} + \text{bk+1} * \text{x0))} \; (7)$$

Despejando la ecuación se llega a la forma:

$$\text{Pk(x0)} = \text{a0} + \text{x0(a1} + \ldots + \text{x0(ak}- 1 + \text{x0} * \text{(ak} + \text{bk+1} * \text{x0))} \; (8)$$

La cual es equivalente a Pk+1(x0), quedando demostrado el número de multiplicaciones iguales a k, el grado del polinomio.

2) **Implemente en R o Python para vericar los resultados del método de Horner**

En la carpeta de parte 1 del taller se encuentra la implementación del método Horner.

3) **Evaluar en x = 1.0001 con P(x) = 1+x+x2 +...+x50. Encuentre el error de cálculo al compararlo con la expresión equivalente Q(x) = (x51 − 1)/(x− 1)**

Evaluando P(x)= 1+x+x2 +...+x50 para x=1.0001 el resultado de P(x)=50.1227 y de Q(x)=(x51-1)/(x-1) es Q(x)=49.0001, con estos valores calculamos error normalizado porcentual que nos da como resultado 2.24%

1.2 Números binarios:

Contextualización: Los números decimales se convierten de base 10 a base 2 con el n de almacenar números en una computadora y para simplicar las operaciones hechas por la computadora, como la suma y la multiplicación Los números binarios se expresan como:

...b2b1b0b− 1b− 2...,

Donde, cada dígito binario, o bit, es 0 o 1. El equivalente en base 10 de un número es:

...b222 + b121 + b020 + b− 12− 1 + b− 22− 2...

1) **Encuentre los primeros 15 bits en la representación binaria de π**

La representación de los primero 15 bits de pi en binario es: 11.00100100001111

2) **Convertir los siguientes números binarios a base 10: 1010101;1011.101;10111.010101...;111.1111...**

Binario	**Base 10**
1010101	85
1011.101	11
10111.010101	23
111.1111	7

3) **Convierta los siguientes números de base 10 a binaria: 11.25;;30.6;99.9**

Base 10	**Binario**
11.25	1011
2/3	10110
30.6	1011011110
99.9	10110111101100011

1.3 Representación del Punto Flotante de los Números Reales / Epsilon de una máquina:

1) **¿ Cómo se ajusta un número binario innito en un número nito de bits?**

Según la norma IEEE-754 un número infinito (+inf o -inf) se representa de forma que los bits correspondientes al exponente mínimo se colocan todos en 1 (8 para 32 bits, 11 para 64 bits) y el bit del signo indicará el signo del infinito

2) **¿Cual es la diferencia entre redondeo y recorte?**

En el proceso de redondeo se usa el valor de la siguiente para obtener un resultado más aproximado del valor. El corte (truncamiento) corta el número de cifras decimales sin tener en cuenta las siguientes para el valor de la última cifra. Se puede decir que el redondeo es más exacto en el valor decimal final, sin embargo, un redondeo a n decimales y corte a n decimales podrán ser los mismos dependiendo de la regla usada para el redondeo.

3) **¿ Cómo se ajusta un número binario innito en un número nito de bits?**

Según el proceso para convertir un número decimal a su representación en binario el valor será:

$$
\begin{gathered}
0{,}4 * 2 = 0{,}8 \rightarrow 0 \\
0{,}8 * 2 = 1{,}6 \rightarrow 1 \\
0{,}6 * 2 = 1{,}2 \rightarrow 1 \\
0{,}2 * 2 = 0{,}4 \rightarrow 0 \\
0{,}8 * 2 = 1{,}6 \rightarrow 1 \\
\ldots \\
= 0{,}0110011001100110011001100110011\ldots \\
= 1{,}10011001100110011001100110011\ldots * 2^{-2}
\end{gathered}
$$

Sin embargo, el carácter de este número es periódico infinito, por lo que para poder ser representado en un computador se utiliza la norma IEEE 754 para representaciones de números en coma flotante. Usando un tamaño de número de 64 bits, divididos en 1 bit para el signo, 11 bits para el exponente mínimo (Biased Exponent) y 52 bits para la precisión de la fracción (Fraction) se encuentra que:

Signo = 0

BiasedExponent = (1021)10 = (01111111101)2

Fraccin = (1001100110011001100110011001100110011001100110011001)

4) **Indique el número de punto otante (IEEE) de precisión doble asociado a x, el cual se denota como fl(x); para x(0.4)**

se tiene que el valor de fl(0,4) es igual en el estándar IEEE 754 a:

(0,4)10 ≈ (001111111101100110011001100110011001...)2 = (0,40000000000000002220446049250313)10

5) **Error de redondeo En el modelo de la aritmética de computadora IEEE, el error de redondeo relativo no es más de la mitad del épsilon de máquina:**

$$\frac{|fl(x) - x|}{|x|} \leq \frac{1}{2}\epsilon_{maq}$$

Teniendo en cuenta lo anterior, encuentre el error de redondeo para x = 0.4

Para encontrar el error de conversión al estándar IEEE 754, se usará la fórmula:

$$\frac{|fl(x)-x|}{|x|} \leq \frac{\epsilon_{maq}}{2}$$
$$\frac{|0,40000000000000002220446049250313-0,4|}{|0,4|} \leq \frac{2^{-52}}{2}$$
$$\frac{|0,40000000000000002220446049250313-0,4|}{|0,4|} \leq 2^{-53}$$
$$5,551115123125783 * 10^{-17} \leq 1,110223 * 10^{-16}$$

El error de redondeo es por lo tanto ≈ 5,551 ∗ 10− 17 o 5,551 ∗ 10− 15 % .

6) **Encuentre la representación en número de máquina hexadecimal del número real 9.4**

La notación hexadecimal se basa en 16 (hexadec es la palabra griega para 16). Esto significa que hay 16 símbolos (dígitos hexadecimales): 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E y F. La importancia de la notación hexadecimal se hace evidente cuando se convierte un patrón de bits a notación hexadecimal.

El número hexadecimal que corresponde al número real 9.4 es: 9.6666666666668

7) **Encuentre las dos raíces de la ecuación cuadrática x2 + 912x = 3 Intente resolver el problema usando la aritmética de precisión doble, tenga en cuenta la pérdida de signicancia y debe contrarrestar.**

Entradas:

f(x)= función a la cual se le calcularán las raíces

f(x)=x2 + 912x = 3

salidas:

x1= 6.00328946 (x=-456+\ sqrt{ 207939})

x2=-906.003 (x=-456-\ sqrt{ 207939})

2. Raíces de una ecuación

1) **Implemente en R o Python un algoritmo que le permita sumar únicamente los elementos de la sub matriz triangular superior o triangular inferior, dada la matriz cuadrada An. Imprima varias pruebas, para diferentes valores de n y exprese f(n) en notación O() con una gráca que muestre su orden de convergencia.**

2) **Implemente en R o Python un algoritmo que le permita sumar los n2 primeros números naturales al cuadrado. Imprima varias pruebas, para diferentes valores de n y exprese f(n) en notación O() con una gráca que muestre su orden de convergencia.**

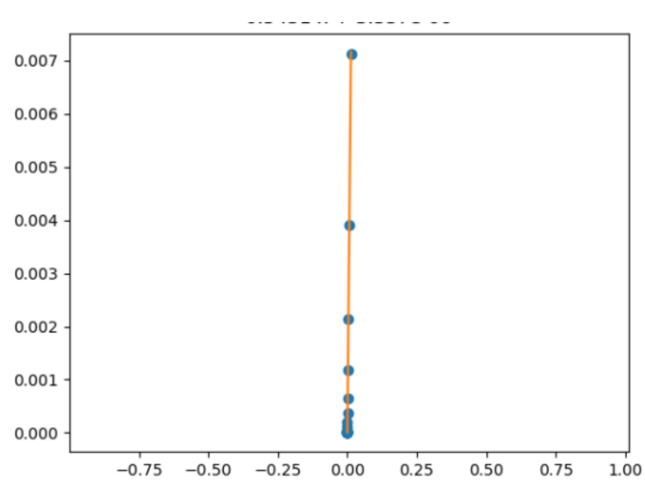
n	Sumatoria
1	1
2	5
3	14
4	30
5	55
6	91
7	140
8	204
9	285
10	385

Para hallar esta sumatoria de los primeros números naturales al cuadrado utilizamos la fórmula

$$S = \frac{n(n+1)(2n+1)}{6}$$

así:

```
 def suma(s):
num=(s* (s+1)* (2* s+1)/6)
return num
numero=int(input())
print(suma(numero))
```



3. Convergencia de métodos iterativos

1.1 *Iteraciones:*

Contextualización: El método de Newton es una fórmula iterativa eciente para encontrar r (raíz real de una ecuación). Es un caso especial del método del punto jo en el que la ecuacion f(x) = 0 se reescribe en la forma x = g(x) eligiendo g de tal manera que la convergencia sea de segundo orden. Se considera uno de los mejores métodos que muestra mejor velocidad de convergencia llegando (bajo cierta condiciones) a duplicar, en cada iteración, los decimales exactos. El método se explica mediante

```
r ← x - f(x)/f'(x)
Mientras |r - x| > E
     x ← r
     r ← x - f(x)/f'(x)
Fin
```

Ver en la carpeta de parte 1 la implementación de Newton

4. Convergencia de métodos iterativos

[1]F. Walter, Introduccion a los metodos numericos, Implementaciones en R. Revista digital, 2013.

[2]L. Rodrι guez, Analisis numerico basico, 3rd ed. Escuela Superior Politecnica del Litoral Phyton, 2014.