1. Numero de Operaciones

Las operaciones fundamentales de la arimética de valores reales, son la suma + y el producto $*$. Estas también, son las operaciones necesarias para evaluar un polinomio $P(x)$ en un valor particular x.Es por esto, que es importante saber cómo se evalua un polinomio y más aun hacerlo de manera eficiente, para que el número de sumas y productos requeridas sea mínima.

Teorema de Horner

Este método, llamado multiplicación anidada o método de Horner,consiste en determinar de manera eficiente el valor del polinomio de grado n utilizando el menor numero de productos. Un polinomio de grado n puede evaluarse en n multiplicaciones y n adiciones

Sea $P(x) = a_0x^n + a_1x^{n-1} + ... + a_n$ un polinomio entonces,

$$b_0 = a_0$$
$$b_k = a_k + b_{k-1}x_0(k = 1, ..., n-1)$$
$$b_n = P(x_0)$$

Así el cociente $P(x)/(x - x_0)$ es: $Q(x) = b_0x^{n-1} + b_1x^{n-1} + ... + b_{n-1}$

Ejemplo: Sea $P(x) = 2x^4 - 3x^2 + 3x - 4$ u polinomio incompleto de grado 4 para cualquier x_0 el polinomio puede ser evaluado de diferentes maneras:

Método 1: $P(x_0) = 2 * x_0 * x_0 * x_0 * x_0 - 3 * x_0 * x_0 + 3 * x_0 - 4$

Método 2: $P(x_0) = -3 * x * (x) + 0 * x * (x^2) + 2 * x * (x^3) + 3 * x - 4$

Método 3: $P(x_0) = -4 + x * (3 - x * (-3 + x * (x * (2))))$

En resumen:

Método	Multiplicaciones
Método 1	$\frac{n(n+1)}{2}$
Método 2	$2n - 1$
Método 3	n

Ejercicio:

1. Utilice el método de inducción matemática para demostrar el resultado del método 2
2. Implemente en R o Python para verificar los resultados del método de Horner
3. Evaluar en $x = 1.0001$ con $P(x) = 1 + x + x^2 + ... + x^{50}$. Encuentre el error de cálculo al comparalo con la expresión equivalente $Q(x) = (x^{51} - 1)/(x - 1)$

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

$$...b_2b_1b_0b_{-1}b_{-2}...,$$

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

$$...b_2 2^2 + b_1 2^1 + b_0 2^0 + b_{-1} 2^{-1} + b_{-2} 2^{-2}...$$

Ejemplo:

1. De decimal a binario: $13.7_{10} = 13_{10} + 0.7_{10}$luego, para la parte entera dada por 13_{10} se tiene que:

$$\begin{aligned} 13 \div 2 &= 6R1 \\ 6 \div 2 &= 3R0 \\ 3 \div 2 &= 1R1 \\ 1 \div 2 &= 0R1 \end{aligned}$$

$\therefore 13 = 1101 = 1x2^3 + 1x2^2 + 0x2^1 + 1x2^0$
La parte fraccional:

$$\begin{aligned} .7x2 &= .4 + 1 \\ .4x2 &= .8 + 0 \\ .8x2 &= .6 + 1 \\ .6x2 &= .2 + 1 \\ .2x2 &= .4 + 0 \\ .4x2 &= .8 + 1 \\ .8x2 &= .6 + 1 \\ &\vdots \\ &\vdots \end{aligned}$$

Tenga en cuenta que el proceso se repite despues de cuatro pasos de manera indefinida exactamente del mismo modo. Por lo tanto:

$$0.7_{10} = .10110011001100110\cdots_2$$

2. Parte fraccional finita binaria a decimal:

$$.1011_2 = 1 * \frac{1}{2} + 0 * \tfrac{1}{4} + 1 * \frac{1}{8} + 1 * \frac{1}{16} = (\frac{11}{16})_{10}$$

Ejercicio:

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

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

c) Convierta los siguientes números de base 10 a binaria: $11.25; \frac{2}{3}; 30.6; 99.9$

Representación del Punto Flotante de los Números Reales

El análisis numérico se refiere a cómo resolver un problema numéricamente, es decir, cómo desarrollar una secuencia de cálculos numéricos para obtener una respuesta satisfactoria.Parte de este proceso es la consideración de los errores que surgen en estos cálculos, de los errores en las operaciones aritméticas o de otras fuentes. En este sentido, hay que considerar que las computadoras usan aritmética binaria, que representa cada número como un número binario (una suma finita de potencias enteras de 2). Algunos números se pueden representar exactamente, pero otros no. Y, por supuesto, están los números trascendentales como π que no tienen representación finita en el sistema numérico decimal o binario.

Las computadoras usan 2 formatos para los números

Punto Fijo:Los números de punto fijo se usan para almacenar enteros.Por lo general, cada número se almacena como máximo 2^{32} números diferentes.

Punto Flotante:La representación de punto flotante (en inglés floating point) es una forma de notación científica usada en los computadores con la cual se pueden representar números reales extremadamente grandes y pequeños de una manera muy eficiente y compacta, y con la que se pueden realizar operaciones aritméticas

El estándar del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE 754, por sus siglas en inglés) es uno de los formatos más utilizados para la arimética de computadora de números en punto flotante. Un número de punto flotante consta de tres partes:

1. Signo $(+o-)$
2. Mantisa, contiene la cadena de bits significativos
3. Exponente

Las tres partes se almacenan juntas en una palabra de computadora y los niveles de uso general para los números de punto flotante son:

Precisión	Signo	Exponente	Mantisa
Sencilla (32 bits)	1	8	23
Doble (64 bits)	1	11	52
Extendida (80 bits)	1	15	64

Adicionalmente, hay que considerar cuando este número es irracional por ejemplo,$\sqrt{3}$ donde no se puede representar con exactitud.Es por esto que los errores de redondeo se tienen en cuenta, ya que surgen dado que la máquina solo puede almacenar un número finito de cifras.

Epsilon de una Máquina
El número épsilon de una máquina, el cual se denota como ϵ_{maq} es la distancia entre 1 y el menor número de punto fotante mayor que 1. Para el punto flotante de precisión doble se tiene que:

$$\epsilon_{maq} = 2^{-52}$$

De aquí surgue la pregunta ¿Cómo se ajusta el número binario infinito que representa 9.4 en un número finito de bits?. Es necesario truncar el número de alguna manera pero, necesariamente tiene error. El **recorte**, consite en eliminar los bits que estan más allá de cierto umbral final.
El método **redondeo**es un método alternativo, esta tecnica tiene reglas dependiendo del sistema númerico.
Ejercicios

1. ¿ Cómo se ajusta un número binario infinito en un número finito de bits?
2. ¿Cuál es la diferencia entre redondeo y recorte?
3. ¿ Cómo se ajusta un número binario infinito en un número finito de bits?
4. Idique el número de punto flotante (IEEE) de precisión doble asociado a x, el cual se denota como $fl(x)$; para $x(0.4)$
5. **Error de redondeo** En el modelo de la arimética 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 $x = 0.4$
6. Verificar si el tipo de datos básico de R y Python es de precisión doble IEEE y Revisar en R y Phython el format long
7. Encuentre la representación en número de máquina hexadecimal del núemro real 9.4
8. Encuentre las dos raíces de la ecuacion cuadrática $x^2 + 9^{12}x = 3$ Intente resolver el problema usando la arimética de precisión doble, tenga en cuenta la pérdida de significancia y debe contrarestarla.
9. Explique cómo calcular con mayor exactitud las raíces de la ecuación:

   $$x^2 + bx - 10^{-12} = 0$$

   Donde b es un número mayor que 100

2. Raíces de una Ecuación

La solución de ecuaciones no lineales o sistemas es un problema importante en el cálculo cientifico y la solución de ecuaciones es un tema central en análisis numérico. Existen métodos iterativos y métodos directos para solucionar el problema $f(x) = 0$.
Tanto Python como R tiene varias herramientas para la aproximación numérica de los ceros de una función.En particular, en Python **SciPy** es una biblioteca open source de herramientas y algoritmos matemáticos para Python. SciPy contiene módulos para optimización, álgebra lineal, integración, interpolación, funciones especiales, FFT, procesamiento de señales y de imagen, resolución de ODEs y otras tareas para la ciencia e ingeniería. Las funciónes que calculan raíces están incluídas dentro del módulo de optimización (scipy.optimize).
En el caso de R, para aproximar los ceros de un polinomio se puede usar la función base **polyroot()** que usa el algoritmo de Jenkins-Traub (1972).
Para aproximar los ceros de una función $f(x)$ está la función uniroot() en la base de R . La documentación indica que esta función usa el método de Brent, este método es un método híbrido que combina bisección e interpolación cuadrática inversa.
No obstante, el uso de librerias especiales hay que considerar aspectos como el costo computacional, implementación, el orden de convergencia del método, complejidad.
En los calculos numericos existen dos metodos de solución:
i. Metodos Iterativos Consite en acercarse a la solución mediante aproximaciones sucesivas, a partir de un valor inicial estimado. En cada iteración se puede usar el resultado anterior para obtener una mejor aproximación
ii. Métodos directos La aproximación a la respuesta se produce a través de una serie finita de operaciones arimeticas y la eficiencia del método depende de la cantidad de operaciones el cual se puede asociar al tamaño del problema , en notación$O()$.
Ejercicios

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 A_n. Imprima varias pruebas, para diferentes valores de n y exprese $f(n)$ en notación $O()$ con una gráfica que muestre su orden de convergencia.

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

3. Para describir la trayectoria de un cohete se tiene el modelo:

$$y(t) = 6 + 2,13t^2 - 0.0013t^4$$

   Donde, y es la altura en $[m]$ y t tiempo en $[s]$. El cohete esta colocado verticalmente sobre la tierra. Utilizando dos métodos de solución de ecuación no lineal, encuentre la altura máxima que alcanza el cohete.

3. Convergencia de Metodos Iterativos

Este capítulo se dedica a investigar el orden de convergencia de los esquemas de iteración funcional y, con la idea de obtener una convergencia rápida, reutilizar el método por ejemplo, de Newton. Consideraremos también maneras de acelerar la convergencia del meetodo de Newton en circunstancias especiales. Pero, antes, tenemos que definir un procedimiento para medir la rapidez de la convergencia.

Definición.Supongamos que la sucesión $\{x_n\}_{n=0}^{\infty}$ una sucesión que converje a x. Si existen constantes positivas λ y α tales que:

$$lim_{n\to\infty}\frac{|x_{n+1}-x|}{|x_n-x|^{\alpha}} = lim_{n\to\infty}\frac{|E_{n+1}|}{|E_n|^{\alpha}} = \lambda$$

Entonces la sucesion $\{x_n\}_{n=0}^{\infty}$ converge a x con orden α y con una constante de error asintótico λ
En general, una sucesión con un orden de convergencia α grande convergerá más rápidamente que una sucesión con un orden más bajo. La constante asintótica afectará la rapidez de convergencia, pero no es tan importante como el orden.
Teorema
Sea p una solución de $x = g(x)$. Supongamos que $g^{'}(p) = 0$ y $g^{''}$ es continua en un intervalo abierto que contiene a p. Entonces existe un $\delta > 0$ tal que, para $p_0 \in [p-\delta, p+\delta]$,la sucesión definida por $x_n = g(x_{n-1})$, para toda $n \geq 1$, es convergente cuadráticamente.
Ejemplo: Suponga que la ecuación $f(x) = 0$ tiene una raíz en $x = p$ y que además $f^{'}(p) \neq 0$ y si consideramos el método del punto fijo:

$$x_n = g(_{n-1})$$

Además, g es de la forma

$$g(x) = x - \phi(x)f(x)$$

Siendo ϕ (x) una función arbitraria, entonces

$$g^{'}(x) = 1 - \phi^{'}(x)f(x) - \phi(x)f^{'}(x)\ g^{'}(p) = 1 - \phi(p)f^{'}(p)$$

Luego $g^{'}(p) = 0$ si sólo si $\phi(x) = \dfrac{1}{f^{'}(p)}$
En particular, se obtendra una convergencia cuadrática para el proceso

$$x_{n-1} = g(x_{n-1}) = x_{n-1} - \frac{f(x_{n-1})}{f^{'}(x_{n-1})}$$

el cual puede reconocerse como el método de Newton.
En particular, el método de Newton y Secante traerán generalmente problemas si $f^{'}(p) = 0$ al tiempo que $f(p) = 0$

Ejercicios
Para cada uno de los siguientes ejercicios implemente en R o Python, debe determinar el número de iteraciones realizadas,una grafica que evidencie el tipo de convergencia del método y debe expresarla en notación $O()$

1. Sean $f(x) = ln(x+2)$ y $g(x) = sin(x)$ dos funciones de valor real.

   a) Utilice la siguiente formula recursiva con $E = 10^{-8}$ para el punto de intersección.:

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

   b) Aplicar el método iterativo siguiente con $E = 10^{-8}$ para encontrar el punto de intersección:

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

2. **Newton**: Determine el valor de los coeficientes a y b tal que
   $f(1) = 3$ y $f(2) = 4$ con $f(x) = a + (ax+b)e^{ax+b}$. Obtenga la respuesta con $E = 10^{-6}$

Multiples soluciones Se dice que una solución p de la ecuación $f(x) = 0$ es un cero de multiplicidad m de f si $f(x)$ puede escribirse como $f(x) = (x-p)^m q(x) \forall x \neq p$ donde $lim_{x \to p} q(x) \neq 0$ Los que tienen multiplicidad uno. A estos ceros se les llama simples

Teorema

Sea $f \in C^1[a,b]$ tiene un cero simple $p \in (a,b)$ si sólo si $f(p) = 0$ pero $f^{'}(p) \neq 0$ Es claro que la solución de los ceros de $f(x)$ presenta dificultades cuando se tienen raíces múltiples, luego una manera de atacar el problema onsiste en definir una función

$$\mu(x) = \frac{f(x)}{f^{'}(x)}$$

Supónga que p es una raíz de multiplicidad $m \geqq 1$ entonces,

$$\frac{(x-p)^m q(x)}{m(x--p)^{m-1} q(x)} + (x--p)^m q^{'}(x)$$

Luego, μ tendrá también una raíz en p, pero de multiplicidad uno. Ahora, si el método de Newton se aplica a la función μ se tiene

$$g(x) = x - \frac{\mu(x)}{\mu^{'}(x)} = \frac{f(x)f^{'}(x)}{(f^{'}(x))^2 - f(x)f^{''}(x)}$$

La anterior formula se conoce como la formula de Newton generalizada.

Ejercicio Sea $f(x) = e^x - x - 1$

1. Demuestre que Tiene un cero de multiplicidad 2 en $x = 0$
2. Utilizando el método de Newton con $p_0 = 1$ verifique que converge a cero pero no de forma cuadrática
3. Utilizando el método de Newton generalizado, mejora la tasa de rendimiento

4. Convergencia Acelerada

Existen básicamente dos metodologías para acelerar la convergencia.
Metodo de Δ^2 Aitken. Técnica que se usa para acelerar la convergencia de cualquier sucesión que converja linealmente, independientemente de su origen.
Suponga una sucesión $\{x_n\}_{n=0}^{\infty}$ linealmente convergente, supongamos que n es lo suficientemente grande para que el cociente pueda usarse para aproximar el límite. Si suponemos tambié que todas las E_n en tienen el mismo signo, entonces existe una sucesión $\{A_n\}_{n=0}^{\infty}$ talque que converje más rápidamente que $\{x_n\}_{n=0}^{\infty}$ y esta dada por:

$$\{A_n\}_{n=0}^{\infty} = x_{n+2} - \frac{(x_{n+2} - x_{n+1})^2}{x_{n+2} - 2x_{n+1} +_x n}$$

Ejercicio: Dada la sucesión $\{x_n\}_{n=0}^{\infty}$ con $x_n = cos(1/n)$

1. Verifique el tipo de convergencia en $x = 1$ independiente del origen
2. Compare los primeros términos con la sucesión $\{A_n\}_{n=0}^{\infty}$
3. Sean $f(t) = 3sin^3t - 1$ y $g(t) = 4sin(t)cos(t)$ para $t \geq 0$ las ecuaciones paramétricas que describe el movimiento en una particula. Utilice un método de solución numérico con error de 10^{-6} para determinar donde las coordenadas coiciden

Metodo de Steffersen Aplicando el método $\triangle^2$ de Aitken a una sucesión que converge linealmente obtenida de la iteración de punto fijo, podemos acelerar la convergencia a cuadrática. Este procedimiento es conocido como el método de Steffersen y difiere un poco de aplicar el de Aitken directamente a una sucesión de iteración de punto fijo que sea linealmente convergente.
Ejercicio Utilice el algoritmo de Steffersen para resolver $x^2 - cosx$ y compararlo con el metodo de Aitken
Nota Debe implementar los códigos en R, utilice **Rcpp** para mejorar el rendimiento