Taller 1 Análisis Numérico

Cristian Da Cámara, Kenneth Leonel, Camilo Sandoval

Primera parte:

Problema 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

Solución 1: En primer lugar, convertimos el numero 536.78 a notación científica

x=536.78

x=0.53678x10^3

Descomponemos el número como dos sumas lo que nos queda:

x=0.5367x10^3 + 0.00008x10^3

A partir de esto queremos calcular el error de redondeo al remover el excedente ya que este dispositivo solo almacena los cuatro primeros dígitos decimales, por lo que utilizamos la propiedad “En general, si n es la cantidad de enteros del número normalizado con potencias de 10,y m es la cantidad de cifras decimales que se pueden almacenar en el dispositivo, entonces si se truncan los decimales sin ajustar la cifra anterior, el error de redondeo absoluto está acotado por: |E|<1*10^n-m, al utilizar esta propiedad se obtiene que el error es de:

E=0.00008x10^3

E=0.8x10^3-m

E=0.8x10^3-4

E=0.8x10^-1

E=0.08

El código en R se encuentra en el archivo Redondeo_Inferior.R

Problema 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.

Solución 2: La precisión del dato correcto depende del error permitido, al igual que el valor inicial, ya que con base a esto el ciclo calcula el número de iteraciones que hace el programa para aproximarnos a la respuesta real

Por último, el resultado obtenido lo elevamos al cuadrado, comparándolo con el valor esperado para comprobar la validez del algoritmo

	cont	errores	arreglosy
1	1	0.0461538462	2.646154
2	2	0.0004025045	2.645751

cont= número de iteraciones

errores=error absoluto por iteración

arreglosy= Valor de la raíz por iteración

El código en R se encuentra en el archivo problemaRaiz.R

Problema 3: Problema: Utilizando el teorema de Taylor hallar la aproximación de e^0.5 con cinco cifras significativas.

Solución 3: Para este ejercicio usamos el método de Taylor, en donde partimos con un valor teórico, hacemos un ciclo donde se aplica el método y la condición de parada es que la diferencia entre el valor teórico y el resultado obtenido sea menor que la tolerancia, fuimos almacenando cada valor obtenido por iteración hasta obtener la mejor aproximación del valor esperado, este algoritmo se encuentra en el archivo Taylor.R

	Experimental	teorico	tolerancia
1	1.648721	1.648721	1e-06

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

Solución 4: Para solucionar este problema necesitamos calcular el error, en el cual utilizamos la propagación del error de redondeo, utilizamos las formulas

$$ErrorAbsoluto = xE_y + yE_x$$

$$ErrorRelativo = \frac{E_y}{y} + \frac{E_x}{x}$$

- ➔ $ErrorAbsoluto = vE_t + tE_v$
- ➔ $ErrorRelativo = \frac{E_v}{v} + \frac{t}{t}$

El código para solucionar este problema se encuentra en el archivo Problema_Distancia.R, los resultados obtenidos fueron los siguientes:

	ErrorA	ErrorR	d	dsumado	drestado
1	0.9	4.5	20	20.9	19.1

ErrorA= Error absoluto d=distancia drestado=distancia-error absoluto

ErrorR= Error relativo d=distancia+error absoluto

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

Solución 5: Para solucionar este problema, se usa el número de coeficientes del polinomio que en este caso son 5 los cuales son:

Coeficientes= {2,0,-3,3,-4}

Usando este arreglo de coeficientes, hacemos un ciclo con el número de iteraciones igual a la cantidad de coeficientes y dentro de ese ciclo utilizamos la siguiente formula:

resultado=resultado*x+Coeficientes[i]

De ahí obtuvimos los siguientes resultados

resultado=10

cantidad de sumas o restas=5

Cantidad de multiplicaciones=5

Cantidad total de operaciones=10

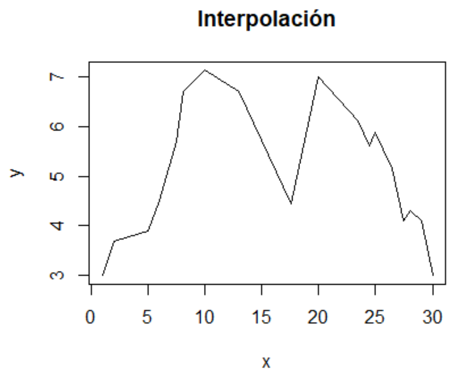
	resultado
1	2
2	-4
3	5
4	-7
5	10

resultado= cantidad de operaciones por iteración

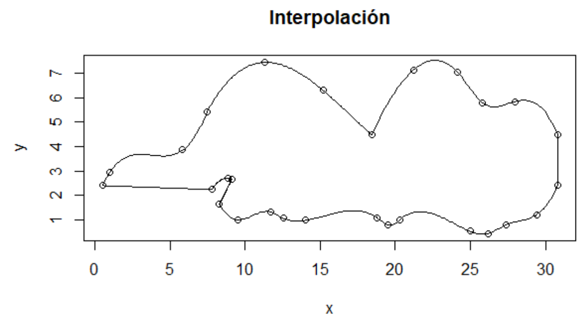
La solución a este problema se encuentra en el archivo Horner_Problema1.R

Problema 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:

Solución 6: Inicialmente organizamos los valores X y Y en dos arreglos, en donde se colocaron puntos extras, se graficaron estos puntos unidos por rectas por lo que nos llevó a tomar intervalos del arreglo en donde mediante una curva uníamos dichos puntos, así sucesivamente hasta completar la silueta



Grafica 1: Solo rectas



Grafica 2: Curvas y puntos adicionales

Segunda parte:

Problema 7: Número de operaciones

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 compararlo con la expresión equivalente Q(x) = (x^51 − 1) / (x − 1)

Solución 7:

1. Inducción
Sea P(x)= a0x^0=a0. El número de multiplicaciones para halla P0(x0) es de 0.

Esto implica que se cumple para el caso 1 k=0

Asumimos que Pk(x)=a0+a1x+...akx^k y Pk(x0) =a0+a1x0+...akx0^k posee k multiplicaciones

Por el método de Horner

bk=ak

bk-1=a(k-1) +bk*x0

...

b0=a0+b1*x0

Y se quiere llegar a k+1

Pk+1(x0) =a0+a1x0+...+ak+x0^(k+1)

Reescribimos Pk(x0) y remplazamos ak por bk

Pk(x0) =a0+x0(a1+...+x0(a(k-1)+x0*bk))

Agregamos una multiplicación mas

bk+1=ak+1

bk=ak+b(k+1) *x0

...

b0=a0+b1*x0

Reemplazamos bk en Pk(x0)

Pk(x0) =a0+x0(a1+...+x0(a(k-1)+x0*(ak+b(k+1)*x0)))

2. Esta implementación está en el archivo Horner_Taller1.R
3. Esta implementación está en el archivo Horner_Taller1.R

	i	errorA	errorR	resultadoIterativo	resultadoDirecto
1	51	7.105427e-14	1.389741e-15	51.12771	51.12771

errorA= error absoluto

error= error relativo

resultadoIterativo= Resultado realizado por la fórmula: P(x) = 1 + x + x^2 + ... + x^50

resultadoDirecto= Resultado realizado por la fórmula: Q(x) = (x^51 − 1) / (x − 1)

Problema 8: Números binarios

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

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

3. Convierta los siguientes números de base 10 a binaria: 11.25; 2/3; 30.6; 99.9

Solución 8:

1-2-3. La implementación de estos programas está en el archivo Problemas_Binarios.R

Problema 9: Épsilon de una Máquina

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. Indique el número de punto flotante (IEEE) de precisión doble asociado a x, el cual se denota como f(x); para x (0.4)

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

$|f(x) - x|/|x| \leq 1/2Emaq$. 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 número real 9.4

8. Encuentre las dos raíces de la ecuación cuadrática x^2 + 912x = 3 Intente resolver el problema usando la aritmética de precisión doble, tenga en cuenta la pérdida de significancia y debe contrarrestarla.

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

Solución 9:

1.Para representar un número infinito en un numero finito de bits, se debe colocar el número 1 en todos los bits del exponente y el número 0 en todos los del significado, y dependiendo del primero bit será infinito negativo o positivo

2.El redondeo consiste en llevar un numero al más cercano terminado en cero, redondear es sustituir un numero por el mas próximo. En cambio, el recorte sin importar que decenas, centenas, unidades de mil etc. dependiendo de cuantos dígitos, este será recortado sin ningún tipo de aproximación.

3.

signo	Exponente	Mantisa
0	11111111	00000000000000000000000
1	11111111	00000000000000000000000

4. 00111110110011001100110011001101

6. El tipo de dato básico de R y Python es de precisión doble

7.La representación hexadecimal del número 9.4 es igual a 5E

Problema 10: Raíces de Ecuación

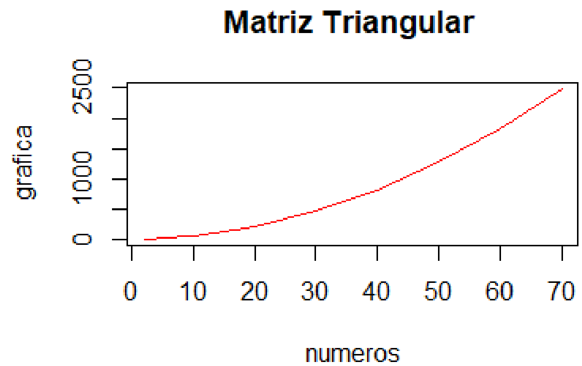
1. Implemente en R o Python un algoritmo que le permita sumar únicamente los elementos de la sub matriz triangular superior o triangular inferior, dada la matriz cuadrada An. Imprima varias pruebas, para diferentes valores de n y exprese f(n) en notación O() con una gráfica que muestre su orden de convergencia.

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.

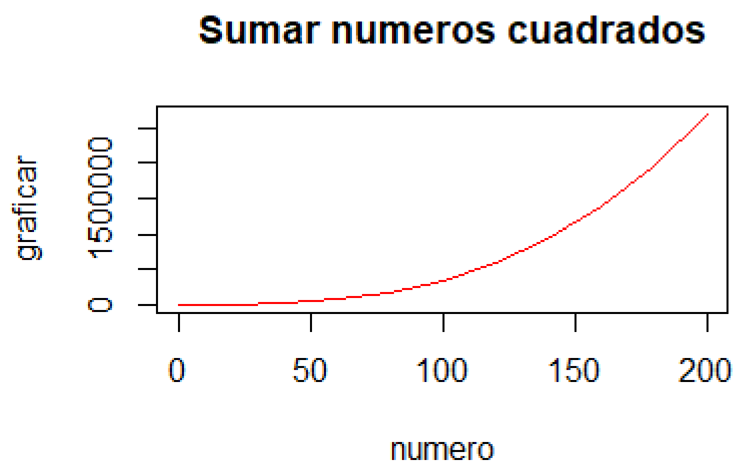
3. Para describir la trayectoria de un cohete se tiene el modelo: y(t) = 6 + 2, 13t 2 − 0.0013t 4 Donde, y es la altura en [m] y t tiempo en [s]. El cohete está colocado verticalmente sobre la tierra. Utilizando dos métodos de solución de ecuación no lineal, encuentre la altura máxima que alcanza el cohete

Solución 10:

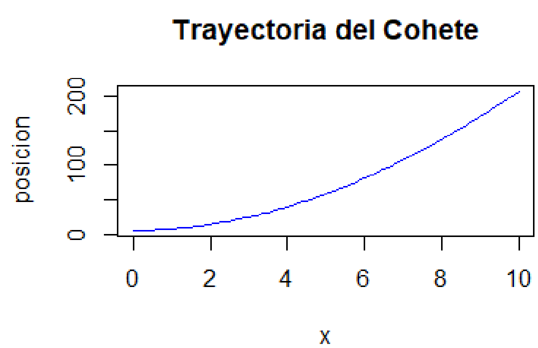
1. De la gráfica podemos concluir que tiene una convergencia positiva puesto que la suma de los elementos de la sub matriz son positivos y va en forma creciente



2.Asi mismo como la gráfica anterior, la sumatoria de los números cuadrados va a tener convergencia positiva debido a que los resultados nunca van a ser negativos por ser al cuadrado, lo que nos da la gráfica creciente



3.La altura máxima alcanzada es de 8778647 metros,



Problema 11: Convergencia de Métodos Iterativos

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:

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

b) Aplicar el método iterativo siguiente con E = 10^−8 para encontrar el punto de intersección:

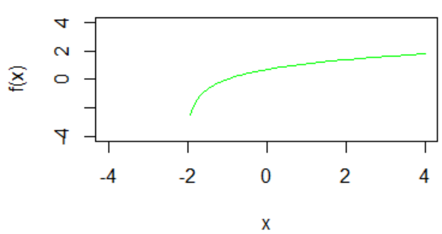
$$x_{n+1} = x_n - f(x_n)\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}$$

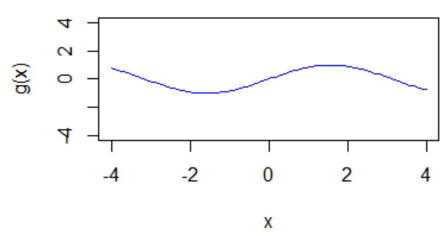
2. Newton: Determine el valor de los coeficientes a y b tal que f (1) = 3 y f(2) = 4 con f(x) = a + (ax + b)e ax+b . Obtenga la respuesta con E = 10−6

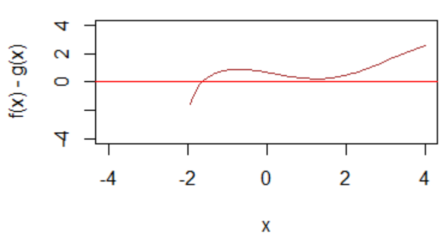
Solución 11:

1.

a-b) Iniciamos graficando las dos funciones f(x) y g(x) las cuales posteriormente restamos para saber el punto donde se intersectan ambas funciones q(x)

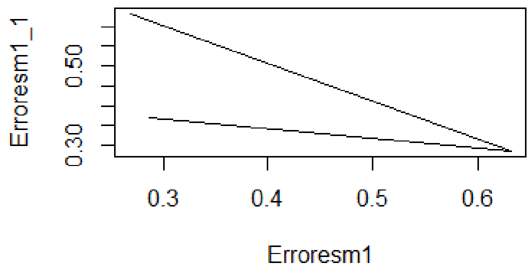


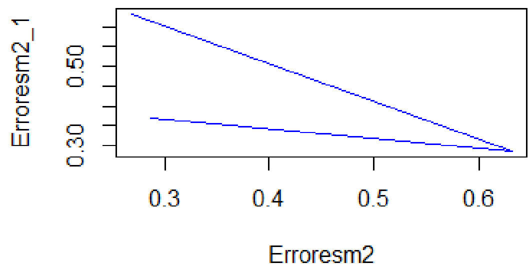




Esta grafica es la que nos muestra el punto donde se intersectan las funcione en el eje x

Posteriormente evaluamos ambas funciones en cada método, el iterativo y el recursivo para mirar cuál de los métodos es más óptimo y en base a ello sacar una conclusión, por lo que gracias a los errores y a su grafica pudimos concluir que ambos métodos son iguales en cuestión de eficiencia





	Erroresm1	Erroresm1_1	Erroresm2	Erroresm2_1
1	0.2685564	0.6314436	0.2685564	0.6314436
2	0.6314436	0.2868540	0.6314436	0.2868540
3	0.2868540	0.3703587	0.2868540	0.3703587
4	0.3703587	NA	0.3703587	NA

Erroresm1: Errores en la primera posición del método recursivo

Erroresm1_1: Errores en la siguiente posición del método recursivo

Erroresm2: Errores en la primera posición del método iterativo

Erroresm2_1: Errores en la siguiente posición del método iterativo

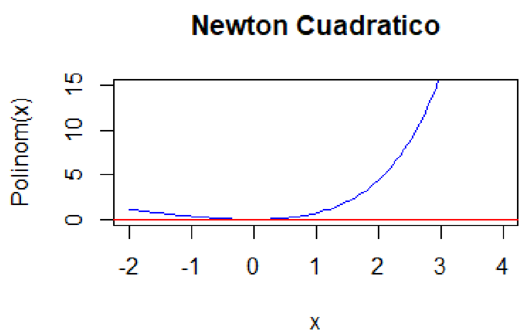
Problema 12: Sea $f(x) = (e^x) - x - 1$

1. Demuestre que Tiene un cero de multiplicidad 2 en $x = 0$

2. Utilizando el método de Newton con $p0 = 1$ verifique que converge a cero, pero no de forma cuadrática

Solución 12:

2. Para verificar si la gráfica converge por el método de Newton se hace un abline(recta de color rojo en la gráfica inferior) en donde se verifica que esta gráfica tiene una tendencia hacia 0



Problema 13: Convergencia Acelerada

Dada la sucesión {xn} ∞ n=0 con xn = cos(1/n)

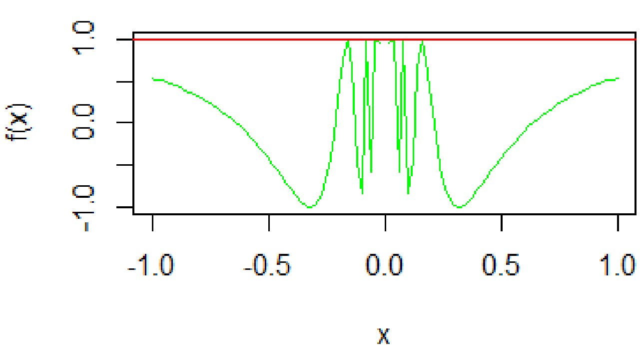
1. Verifique el tipo de convergencia en x = 1 independiente del origen

2. Compare los primeros términos con la sucesión {An} ∞ n=0

3. Sean f(t) = (3sin^3) * t − 1 y g(t) = 4sin(t)cos(t) para t ≥ 0 las ecuaciones paramétricas que describe el movimiento en una partícula. Utilice un método de solución numérico con error de 10−6 para determinar donde las coordenadas coinciden

Solución 13:

1.Hicimos un abline para saber si se intercepta con la función coseno por lo que la graficamos en una escala de -1 a 1 para corroborar esto



2.

	error	errormas1	resultados	aiken
1	7.500000e-01	3.750000e-01	0.8500000	0.6000000
2	3.750000e-01	1.875000e-01	0.4750000	0.6000000
3	1.875000e-01	9.375000e-02	0.6625000	0.6000000
4	9.375000e-02	4.687500e-02	0.5687500	0.6625000
5	4.687500e-02	2.343750e-02	0.6156250	0.6312500
6	2.343750e-02	1.171875e-02	0.6390625	0.6312500
7	1.171875e-02	5.859375e-03	0.6273438	0.6390625
8	5.859375e-03	2.929688e-03	0.6332031	0.6390625
9	2.929688e-03	1.464844e-03	0.6361328	0.6371094
10	1.464844e-03	7.324219e-04	0.6375977	0.6361328
11	7.324219e-04	3.662109e-04	0.6368652	0.6366211
12	3.662109e-04	1.831055e-04	0.6364990	0.6366211
13	1.831055e-04	9.155273e-05	0.6366821	0.6366211
14	9.155273e-05	4.577637e-05	0.6365906	0.6366211

15	4.577637e-05	2.288818e-05	0.6366364	0.6366211
16	2.288818e-05	1.144409e-05	0.6366135	0.6366211
17	1.144409e-05	5.722046e-06	0.6366249	0.6366211
18	5.722046e-06	2.861023e-06	0.6366192	0.6366211
19	2.861023e-06	1.430511e-06	0.6366220	0.6366192
20	1.430511e-06	7.152557e-07	0.6366206	0.6366192
21	7.152557e-07	3.576279e-07	0.6366199	0.6366197
22	3.576279e-07	1.788139e-07	0.6366195	0.6366199
23	1.788139e-07	8.940697e-08	0.6366197	0.6366198
24	8.940697e-08	4.470348e-08	0.6366198	0.6366198
25	4.470348e-08	2.235174e-08	0.6366198	0.6366198
26	2.235174e-08	1.117587e-08	0.6366198	0.6366198
27	1.117587e-08	5.587935e-09	0.6366198	NA
28	5.587935e-09	NA	0.6366198	NA

De aquí podemos concluir que a pesar de que el método de Aiken tiene menos iteraciones en converger, no habría mucha diferencia en usar algún método puesto que la diferencia del método de bisección y el método de Aiken no tienen una distancia significativa en el número de sus iteraciones, como conclusión podemos decir que podemos el método a nuestra preferencia

3.

```
> vectorg
[1]  2.375449  7.049333  8.658635 13.332493 14.941820 19.615698
> ###############
> vectorh
 [1]  1.570816  3.141592  4.712389  6.283186  7.853982  9.424777 10.995575 12.566371 14.137167 15.707962
[11] 17.278760 18.849556
> vectorsolu
[1] 0
```

Al probar e implementar los códigos con rangos de números determinados, luego guardamos las raíces en dos arreglos y en un ciclo comparamos el contenido de ambos arreglos y se llena un arreglo nuevo si ambos arreglos tienen la misma raíz, por lo que, tras muchas pruebas, llegamos a la conclusión de que no hay raíces en común por lo que el arreglo nuevo queda vacío