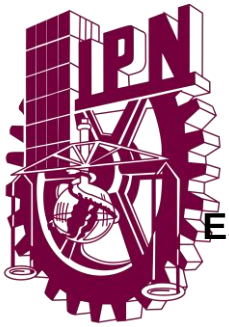


Instituto Politécnico Nacional
Escuela Superior de Ingeniería Mecánica y Eléctrica
Unidad Culhuacán

Practica #1

Errores

Materia: Análisis numérico

Alumno: Serrano Lagunas Fernando Alberto

Profesor: Dr. Trejo Baños Alejandro

Grupo: 4CM31

Introducción:

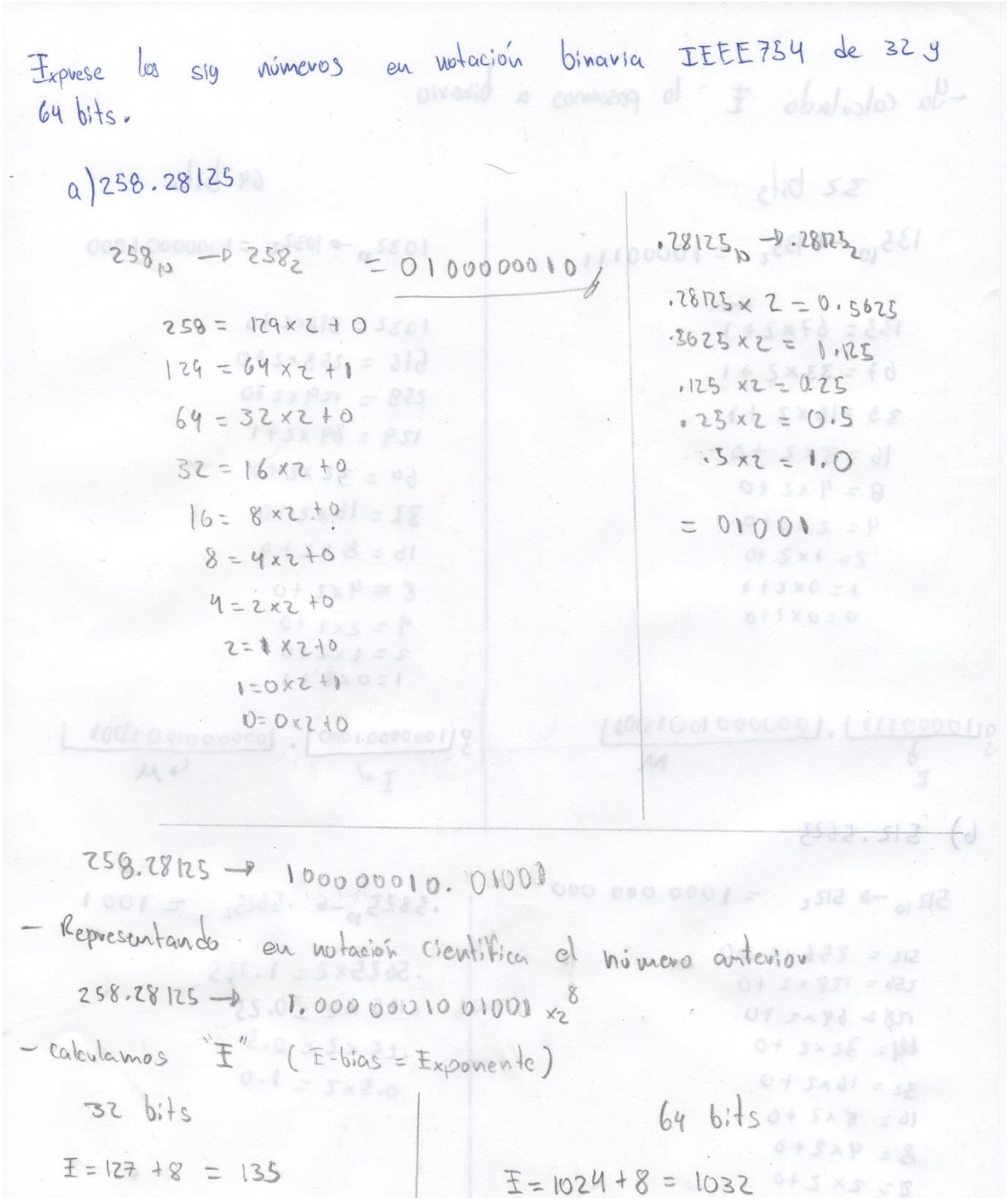
Se realizaran transformaciones de números de base 10 a base 2, para denotarlos en la manera IEEE754 de 32 y 64 bits, también se podrá observar porque la resta nos puede dar significativamente un error en el programa y como las series de Taylor nos pueden dar un valor exacto con solo conocer un valor exacto de la función a evaluar, así como su exactitud.

Practica 1: Errores

Errores de redondeo:

1.- Exprese los siguientes números en notación binaria IEEE754 de 32 y 64 bits (para la mantisa escriba únicamente los dígitos significativos)

a) 258.28125
b) 512.5625
c) 379.1875



-Ya calculado E lo pasamos a binario

32 bits

$135_{10} \rightarrow 135_2 = 10000111$

135 = 67 x 2 + 1
67 = 33 x 2 + 1
33 = 16 x 2 + 1
16 = 8 x 2 + 0
8 = 4 x 2 + 0
4 = 2 x 2 + 0
2 = 1 x 2 + 0
1 = 0 x 2 + 1
0 = 0 x 2 + 0

0 | 10000111 | . | 00000010010001 |
S | E | M

64 bits

$1032_{10} \rightarrow 1032_2 = 10000001000$

1032 = 516 x 2 + 0
516 = 258 x 2 + 0
258 = 129 x 2 + 0
129 = 64 x 2 + 1
64 = 32 x 2 + 0
32 = 16 x 2 + 0
16 = 8 x 2 + 0
8 = 4 x 2 + 0
4 = 2 x 2 + 0
2 = 1 x 2 + 0
1 = 0 x 2 + 1

0 | 10000001000 | . | 0000001001001 |
S | E | M

b) 512,5625

$512_{10} \rightarrow 512_2 = 1000\ 000\ 000$

512 = 256 x 2 + 0
256 = 128 x 2 + 0
128 = 64 x 2 + 0
64 = 32 x 2 + 0
32 = 16 x 2 + 0
16 = 8 x 2 + 0
8 = 4 x 2 + 0
4 = 2 x 2 + 0
2 = 1 x 2 + 0
1 = 0 x 2 + 1

$.5625_{10} \rightarrow .5625_2 = 1001$

.5625 x 2 = 1.125
.125 x 2 = 0.25
.25 x 2 = 0.5
0.5 x 2 = 1.0

$512.5625_{10} \rightarrow 512.5625_2 = 1000000000.1001$

$= 1.0000000001001 \times 2^9$

Calculamos E

32 bits

$E = 127 + 9 = 136$

$136_{10} \rightarrow 136_2 = 10001000$

0	10001000	00000000 1001
S	E	M

64 bits

$E = 1024 + 9 = 1033$

$1033_{10} \rightarrow 1033_2 = 10000001001$

0	100 0000 1001	000 000 000 1001
S	E	M

c) 379.1875

$379_{10} \rightarrow 379_2 = 00\,1111011$

$.1875_{10} \rightarrow .1875_2 = 0011$

$379.1875_{10} \rightarrow 379.1875 = 101111011.0011$

$= 1.01111011001 \times 2^8$

32 bits

$E = 127 + 8 = 135 = 10000111$

0	10000111	01111011001
S	E	M

64 bits

$E = 1024 + 8 = 1032 = 10000001000$

0	1000000 1000	01111011001
S	E	M

2.-El error de redondeo surge al expresar números infinitos en el sistema binario, al no tener memoria infinita las máquinas realizan un corte o un redondeo, introduciendo un error. El error es aún más dramático en ciertas situaciones como cuando se restan dos números prácticamente iguales, situación que se presenta por ejemplo en:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \qquad (1)$$

Cuando $b^2 >> 4ac$

Para demostrarlo realizar lo siguiente:

a) Evalúe de manera exacta las raíces de $x^2 + 3000.001x + 3 = 0$

$X_1 \approx -.001; \quad X_2 = -3000;$

b) Realice un programa que resuelva la misma ecuación evaluando la formula (1) UTILIZANDO NUMEROS DE TIPO FLOAT o precisión simple.

$X_1 = .00097562; \quad X_2 = -3000;$

c) Compare el valor exacto de las dos raíces con el valor calculado por la computadora, calcule el error relativo porcentual de la raíz cuyo valor es distinto al exacto.

$E_r = 2.34\%$

d) Realice el cálculo con precisión doble.

$X_1 = -.001; \quad X_2 = -3000;$

e) Compare los resultados con el valor exacto y la precisión simple, verifique el porcentaje de error.

$E_r = 0\%$

f) Repita el cálculo con precisión simple pero para la primer raíz utilice la siguiente fórmula:

$$x_1 = \frac{-2c}{b \pm \sqrt{b^2 - 4ac}}$$

, compare los resultados con los del inciso b) y el exacto

$X_1 = -.001;$

Responda las siguientes preguntas:

¿Cuál es la mejor opción para evaluar esta ecuación en un sistema con poca memoria?, ¿Por qué?

Usar la fórmula del inciso f, con esta podemos usar float y nos da un error del 0%

Pero hay que tener cuidado, ya que en unos casos no da un error exacto.

¿Cuál es la mejor opción si el sistema tiene la memoria suficiente?, ¿Por qué?

Usar doublé con la fórmula del inciso f, este ya no sufre de ningún problema en los dos casos evaluados debido a su cantidad de bits que puede almacenar.

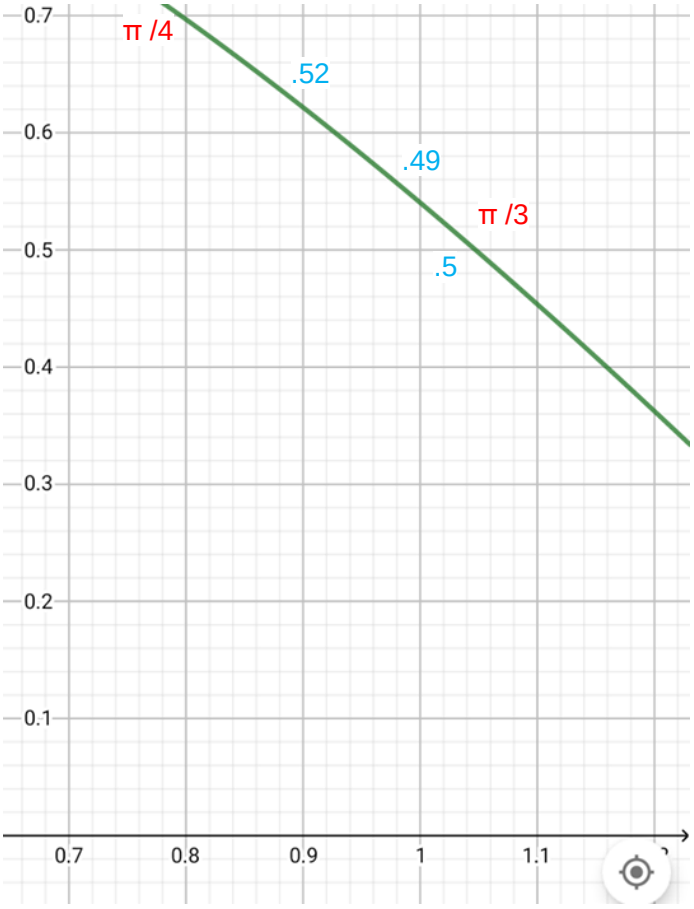
Errores de truncamiento

1.-Realice un programa que evalúe las aproximaciones de orden 0 a 5 de la serie de Taylor de la siguiente función, escriba una tabla del valor de las aproximaciones en $x=x_{max}$.

$$f(x) = \cos(x)\ a = \frac{\pi}{4},\ x_{max} = \frac{\pi}{3}, \qquad (2)$$

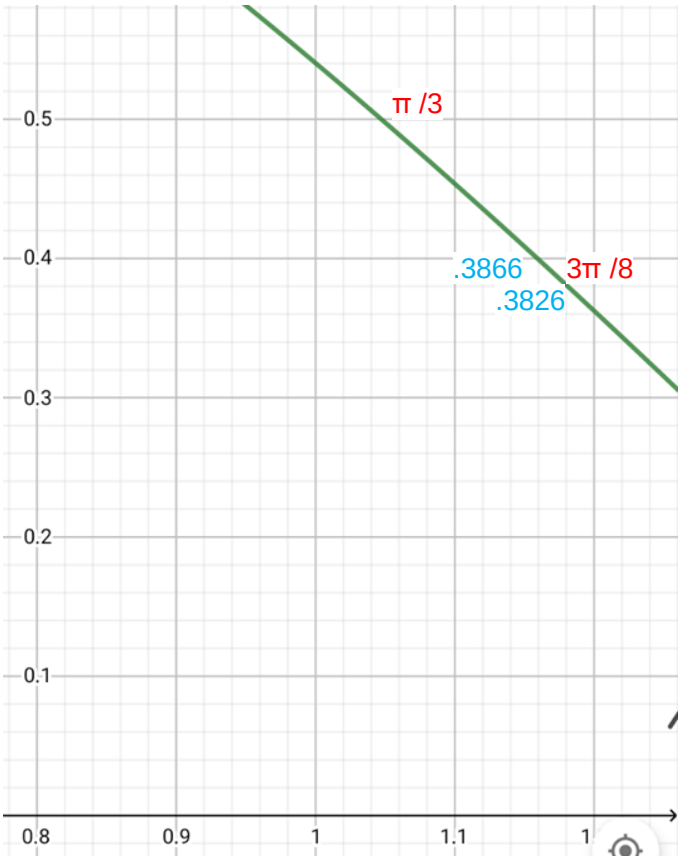
ITERACION	F(X)
0	.07172
1	.521987
2	.497754
3	.499869
4	.500008
5	.5

2.-Grafique la función en el intervalo dado y sus aproximaciones de orden 0 al 5 ¿Qué tanto se acercan los valores aproximados al valor real en cada una de las aproximaciones?



3.-Cambie $a = \frac{\pi}{3}, x_{max} = \frac{3\pi}{8}$ calcule las aproximaciones de orden 0 a 5, grafique y evalúe cada una de las aproximaciones en $x=x_{max}$, ¿Qué diferencias observa respecto a los resultados de 1 y 2?

ITERACION	F(X)
0	.5
1	.386638
2	.382354
3	.382678
4	.382684
5	.382683



R= Hay un error menor en la aproximación al valor real.

Conclusión:

Se realizaron exitosamente las transformaciones de un numero de base 10 a IEEE754, en cuanto las raíces de las ecuación planteada se puede observar como el poco uso de memoria en una resta nos puede quitar bits de memoria, mientras con una suma esta sale exactamente sin necesidad de usar “doublé”, así como en las series de Taylor donde se puede observar que usar un valor más aproximado a el valor real nos puede dar un valor más exacto en las primeras iteraciones.