Ejercicios métodos numéricos

ERRORES

1. Convertir los siguientes números en formato IEEE 754 a decimal.

   a) 0 0000 0011 0111 1011 0000 0000 0000 000
   b) 1 0101 0101 0111 0110 0000 0000 0001 011

2. Calcular el número positivo más grande que se puede representar en formato IEEE 754 con precisión doble.
3. Representar el número 34,343 en formato IEEE 754 de precisión simple.
4. Obtener la representación binaria del número decimal (-0.00015) en el formato IEEE754 para coma flotante de 32 bits.
5. Convertir el número -2.5675e15 al formato IEEE 754 de 32 bits.
6. Representar el número 0.1 en formato IEEE 754 de 32 bits. ¿Qué consecuencias sacas del resultado?
7. Convertir en número 1100 0001 1001 1110 0000 0000 0000 0000 en formato IEEE de 32 bits a formato decimal.
8. Calcular el error absoluto y el error relativo al realizar las siguientes aproximaciones
   a. $\tilde{\boldsymbol{x}} = .1, \boldsymbol{x} = .\mathbf{09934}$
   b. $\tilde{\boldsymbol{x}} = 534, \boldsymbol{x} = \mathbf{533.9888}$
   c. $\tilde{\boldsymbol{x}} = 34.454, \boldsymbol{x} = \mathbf{34.455}$
   d. $\tilde{\boldsymbol{x}} = 0, \boldsymbol{x} = .\mathbf{0001}$
9. Demostrar que el error absoluto de la suma de dos valores a y b, $\mathbf{x} = \mathbf{a} + \mathbf{b}$ verifica la desigualdad $|e_{abs}(\boldsymbol{x})| \leq |e_{abs}(\boldsymbol{a})| + |e_{abs}(\boldsymbol{b})|$
10. De dos números a y b sabemos que sus aproximaciones son $\tilde{\boldsymbol{a}} = \mathbf{4.56}$ **y** $\tilde{\boldsymbol{b}} = \mathbf{1.23}$. sabiendo que $|e_{abs}(\boldsymbol{a})| \leq 0.14$ y $|e_{abs}(\boldsymbol{b})| \leq 0.03$.
    a. Calcular en que rango de valores se encuentran a y b
    b. Calcular una cota para el valor absoluto del error absoluto de a+b.
    c. Indicar en que rango de valores de encuentra a+b
11. Calcular una cota para el valor absoluto del error relativo del cociente de dos números a y b.
12. Demostrar que para la función $f(x) = x^n$ se verifica $e_{rel}(\boldsymbol{y}) \cong$ = n $e_{rel}(\boldsymbol{x})$ para todo n > 1
13. Un algoritmo A se puede definir como la aplicación reiterada de una cierta función φ(x) sobre un valor inicial x_0, es decir $A(x_0)$= φ(φ(φ(… n veces… φ(x_0)))) Suponiendo que $\left|\frac{d\varphi(x)}{dx}\right| \leq C$ entonces, $e_{abs}(\mathrm{A}(x_0)) \cong C^n e_{abs}(x_0)$
14. Decidir para que valores de p y q el algoritmo definido por la siguiente función está bien condicionado $f(p,q) = -p + \sqrt{p^2 + q}$
15. Error por truncamiento. Demostrar que, si en lugar de redondeo usamos truncamiento para aproximar un número, es decir si $x = sign(x)a * 10^b$ con $a = 0.\alpha_1\alpha_2\ \alpha_3 \ldots.\ \alpha_i$ $\alpha_{i+1} \ldots.con\ 0 \leq \alpha_i \leq 9$ y $\alpha_1 \neq 0$ y aproximamos $\tilde{x} = sign(x)\tilde{a} * 10^b$ con $\tilde{a} = 0.\alpha_1\alpha_2\ \alpha_3 \ldots.\ \alpha_t$ entonces $|e_{rel}(\boldsymbol{x})| \leq 10^{-(t-1)}$
16. Precisión en base 2. Demostrar si $x \notin \boldsymbol{A}$ y dicho número lo escribimos $x = sign(x)a * 2^b$ con $a = 0.\alpha_1\alpha_2\ \alpha_3 \ldots.\ \alpha_i\ \alpha_{i+1} \ldots.\ 0 \leq \alpha_i \leq 1$ y $\alpha_1 \neq 0$ y aproximamos $\tilde{x} = sign(x)\tilde{a} * 2^b con$

$$\tilde{a} = \begin{cases} 0.\alpha_1\alpha_2\ \alpha_3....\ \alpha_t & si\ \alpha_{t+1} = 0 \\ 0.\alpha_1\alpha_2\ \alpha_3....\ \alpha_t + 2^{-t} & si\ \alpha_{t+1} = 1 \end{cases}$$

Entonces $|e_{rel}(\boldsymbol{x})| \leq 2^{-t}$.

17. Demostrar que en el caso de aproximación por redondeo existen números $x \notin \boldsymbol{A}\ \boldsymbol{tales\ que}\ \tilde{x} \notin \boldsymbol{A}$ (Sugerencia: Trabajad con máquinas con t=3 y base 10).
18. Demostrar que, en general, para un número en base B, si aproximamos por redondeo, se verifica $|e_{rel}(\boldsymbol{x})| \leq \frac{B}{2} * B^{-t}$.
19. Demostrar, que en números en base 10 en una máquina de t dígitos significativos, el error absoluto cometido al redondear verifica $|e_{abs}(\boldsymbol{x})| \leq 5|x|10^{-t}$.