Tarea 1

Juan Pablo Brenes Coto

CE-3102 Análisis Numérico para Ingeniería

Prof. Dr. Pablo Alvarado Moya

7 de Agosto de 2018

---

1. **Encuentre la representación binaria exacta de 0.1_{10}**

   Se sabe que:

   $$0{,}1 = \frac{1}{10} = \frac{0001_2}{1010_2} \tag{1}$$

   Por lo tanto, aplicando división binaria se obtiene:

   $$\begin{array}{r|l} & 0.0\overline{0011} \\ 1010 & 1.00000000 \\ & \underline{0} \\ & \underline{10} \\ & \underline{0} \\ & \underline{\mathbf{100}} \\ & \underline{0} \\ & \underline{1000} \\ & \underline{0} \\ & \underline{10000} \\ & \underline{1010} \\ & \underline{1100} \\ & \underline{1010} \\ & \underline{\mathbf{100}} \\ & \cdot \\ & \cdot \\ & \cdot \end{array}$$

   El número $0{,}1_{10}$ no posee una representación binaria exacta debido a que durante la división binaria el termino "100_2"vuelve a reaparecer como cociente, por lo se repite la misma división. Por lo tanto, como resultado se obtiene que la representación binaria de $0{,}1_{10}$ es $0{,}0\overline{0011}_2$, donde el termino "0011_2"se repite infinitamente.

2. **En GNU/Octave, introduzca los siguiente comandos:**

   ```
   output_precision(30)
   a=single(0.1)
   b=double(0.1)
   double(a)-b
   ```

   Explique qué hace cada línea y qué resultado produce cada una. Indique por qué el resultado final no es igual a cero.

   output_precision(30): Establece en 30 el número de cifras significativas a mostrar en la consola, ya que por defecto octave solo muestra 5 cifras.

a=single(0.1): Asigna a la variable "a" el valor 0.1 con una precisión simple en coma flotante, es decir, su representación en binario ocupa 32 bits, con una precisión de 24 bits (mantisa).
Resultado: 1.00000001490116119384765625000e-01

b=double(0.1): Asigna a la variable "b" el valor 0.1 con una precisión doble en coma flotante, por lo tanto su representación en binario ocupa 64 bits de espació, con una mantisa de 53 bits.
Resultado: 1.00000000000000005551115123126e-01

double(a)-b: El valor almacenado en "a" (0.1 en precisión simple) lo representa con una precisión doble y le resta el valor almacenado en "b" (0.1 en precisión doble).
Resultado: 1.49011611383365050187421729788e-09

Debido a que el número $0{,}1_{10}$ no posee una representación binaria exacta, sino que posee una cantidad inifinita de terminos $(0{,}0\overline{0011}_2)$, al almacenar dicho valor en una variable estos terminos tambien son almacenados, y la cantidad que es almacenada depende de la precisión utilizada. En el caso de la variable "**a**", al tener una precisión simple almacena solamente 24 bits significativos de la cantidad inifinita que posee $0{,}1_{10}$ en binario; los cuales al ser convertidos a base 10 dan como resultado un numero que es un poco mayor 0.1.

Mientras que la variable "**b**", al tener una precisión doble almacena 54 bits significativos, los cuales al convertirse a base 10 dan como resultado un número que sigue siendo un poco mayor al valor real de 0.1, pero mucho más pequeño que el valor de "a" y más proximo al valor real.

Cuando se ejecuta la operación "**double(a)**" se obtiene la representación en precisión doble del valor almacenado por la variable "a", el cual es un número mayor tanto del valor real de 0.1 como de su representación en precisión doble, por lo que como resultado de esta operación se obtiene un número que sigue siendo mayor que el almacenado por la variable "b", por lo que al restarle el valor almacenado por "b" se obtiene un número positivo extremadamente pequeño pero difrente de cero.