Computación Científica

Práctica 1

Instituto de Educación

Universidad Nacional de Hurlingham

1) Sea $m = 112$. Escribir todos los números en coma flotante con una precisión $t = 4$ y un exponenete $e \in \{-1, 0, 1, 2, 3\}$ que pueden generarse con dicho m.

2) Escribir los siguientes números en la forma normalizada $0.d_1d_2... \times 10^e$:

   a) 101.2
   b) 355
   c) 122.4
   d) 0.901

   Reescribirlos como $\pm m \times 10^{e-t}$

3) Escribir los siguientes números en el estándar IEEF754 de 32 bits y 64 bits:

   a) Los números del ejercicio 2
   b) 1.001
   c) 10100
   d) 0.001
   e) 1
   f) +0
   g) -0
   h) ∞ y -∞

4) Convertir los siguientes números IEEE 754 a decimal. Cuando haya puntos suspensivos, indican que se completa con ceros.

   a) 0 10000011 01000000000000000000000
   b) 1 01111111 00000000000000000000000
   c) 1 00110100001 1010... 00101 (64 bits)
   d) 0 00000100001 1010010... 0010 (64 bits)

   Investigue qué representa una codificación con todos unos en el exponente y mantisa distinta de cero.

5) Sea β la base en la que trabaja una máquina imaginaria y llamemos F a al conjunto de todos los números de máquina. Pruebe que la distancia entre 1 y el siguiente número de máquina es $\varepsilon_M = \beta^{1-t}$. Esta cantidad se llama épsilon de máquina. Cuando no haya ambigüedad escribiremos simplemente ε. Describa a ε para el estándar de precisión simple y doble. (Recuerde que si queremos ser consistentes con la notación de punto flotante normalizada para cualquier base, en precisión simple tenemos $t = 24$ y en precisión doble $t = 53$, sin embargo, sabemos que al pasar al estándar IEEEF754, se optimiza no guardar el bit que vale 1, por lo que allí, la longitud de la mantisa es 23 y 52 respectivamente.)

6) (Continuación) En la bibliografía se define a ε como el mínimo elemento y de F tal que $1 + y \neq 0$. Explique por qué tiene sentido esta definición.

7) Hallar el error (absoluto) y el error relativo en los siguientes casos:

   a) $x = 3.141592$ y $\hat{x} = 3.14$
   b) $y = 1000000$ y $\hat{y} = 999996$
   c) $z = 0.000012$ y $\hat{z} = 0.000009$

   Determine para cada caso, si se está en presencia de una buena aproximación o no (Sugerencia: interprete el error relativo como porcentaje de x). Considere ahora una máquina que maneja la cantidad de dígitos significativos de las entradas. ¿Se cumplen las cotas para los errores relativos? ¿Qué sucede si se utiliza aritmética de dos dígitos?

8) En una máquina de cinco decimales que redondea correctamente los números al número de máquina más cercano, ¿qué números reales x tendrán la propiedad $\mathrm{fl}(1+x)=1$?

9) Supongamos que estamos utilizando una máquina que maneja los números de la forma $x = 0.1d_2d_3d_4 \times 2^e$, significando de 4 dígitos). Calcule el error relativo al realizar los siguientes cálculos:

   a) $\frac{1}{3}+\frac{1}{5}$
   b) $\frac{1}{3}\times\frac{1}{5}$
   c) $\frac{1}{5}-\frac{1}{3}$
   d) $\frac{1}{8}+4$
   e) $\left(\frac{1}{8}+4\right)-4$
   f) $\frac{1}{8}+(4-4)$

10) Realizar las siguientes operaciones en Python y comparar el resultado esperado con el obtenido.

   a) $1+\frac{\varepsilon}{2}$
   b) $\left(1+\frac{\varepsilon}{2}\right)+\frac{\varepsilon}{2}$
   c) $1+\left(\frac{\varepsilon}{2}+\frac{\varepsilon}{2}\right)-1$
   d) $\left(\left(1+\frac{\varepsilon}{2}\right)+\frac{\varepsilon}{2}\right)-1$
   e) $1+\left(\frac{\varepsilon}{2}+\left(\frac{\varepsilon}{2}-1\right)\right)$

11) Calcular el antecesor de 1 en una máquina de 64-bits.

12) El número e se puede definir por $e=\sum_{n=0}^{\infty}\left(\frac{1}{n!}\right)$. Calcule el error absoluto y el error relativo con truncamiento de 4 dígitos en las siguientes aproximaciones de e:

   a) $\sum_{n=0}^{5}\left(\frac{1}{n!}\right)$
   b) $\sum_{n=0}^{10}\left(\frac{1}{n!}\right)$

13) Muestre que considerar $x(1+\delta)$ para una representación de x, es equivalente a decir que se está tomando x contaminado con un error relativo igual a δ.

14) El concepto de dígitos significativos suele ser bastante elusivo. Es también llamado figuras significativas. Sean $x = 12.071$ e $y = 0.012071$. Considere los mismos en la escritura en la cual están presentados (es decir, sin normalizar) y confeccione una tabla con 3 columnas: precisión, redondeo a cifras significativas, redondeo a cifras decimales. Comenzando con una precisión de 7, y terminando en cero. ¿Cuántas cifras significativas se pierden si restamos a x 12.02 y cuáles son respecto de su relevancia? ¿Y si restamos 12.021? Para reflexionar: ¿tiene sentido distinguir entre cifras significativas y decimales en notación normalizada?

15) Sea $x = 12.121135768$ y supongamos que estamos trabajando con una máquina que maneja aritmética de 5 dígitos decimales. ¿Cuántos dígitos significativos se pierden al restarle a x los truncados 10, 12, 12.1, 12.12, 12.121, etc? ¿Cuáles son dichos dígitos?

16) Ahora considere que tiene la precisión necesaria para realizar los cálculos de forma exacta. Examine el error absoluto y el error relativo luego de realizar las sucesivas restas en el ejercicio anterior. ¿Es posible establecer una relación entre coincidir con k cifras significativas y alguno de estos errores? Cree una (o varias si desea separar en casos) función en Python que tome como INPUT un número y devuelva un print de una tabla donde se aprecie la evolución del INPUT, los truncados, y los errores al hacer las restas.

17) ¿Es una buena idea definir las siguientes funciones como se presentan?

    a) $f(x) = \frac{1-\sin(x)}{x}$ cerca de 0.

    b) $f(x) = \ln(x) - 1$ cerca de 1.

    Justifique su respuesta de forma que se pueda cuantificar que tan buena o no es. En caso de no serlo, proponga mejoras y justifique.

18) Si x, y y z son números de máquina en una computadora de 32 bits, ¿qué límite superior se puede dar para el error de redondeo relativo al calcular $z(x + y)$?

19) Suponiendo que se trabaja con una máquina de 4 cifras decimales, calcular el error relativo al resolver $x^2 + 100.1x + 1 = 0$ utilizando la fórmula resolvente y decidir si su precisión es buena. ¿Cuántas formas de mejorar el cálculo hay?

20) Sabemos que en precisión simple el significando ocupa 24 bits. Esto es lo que llamamos precisión, pero está en binario. ¿Cuál sería la precisión aproximada en dígitos decimales para este estándar? ¿Y para el estándar de precisión doble? [Ayuda: piense cuántos dígitos binarios son necesarios para representar las unidades, luego las unidades y las decenas, etc. ¿Hasta dónde se llega con 24 dígitos?]