Tarea-Práctica 4.
Programación para la física computacional
Errores numéricos

Fecha de entrega: **3 de octubre de 2024**

1. **Representación de punto flotante**

   (a) Convierte los siguientes `números de maquina` en su representación decimal:

   - 0 10000000 01011011111100001010100
   - 1 10000000 10010010000111111011011
   - 0 01111111 10011110001101110111101
   - 1 10000000011 1011100100010001000100000111000000000000101010011000
   - 0 10010010011 1000100101010111000100000111000001010001001010101001

   (b) Convierte las siguientes **constantes físicas** en su *representación de punto flotante* (`números de maquina`) de 32 bits:

   - La constante de Cavendish $G = 6.6743 \times 10^{-11}\ m^2/Kg^2$
   - La velocidad de la luz en el vacío $c = 299792458\ m/s$
   - La constante de Boltzmann $k_\mathrm{B} = 1.380649 \times 10^{-23} J/K$
   - La constante de Planck $h = 6.62607015 \times 10^{-34}\ Js$
   - El número de Avogadro $N_\mathrm{A} = 6.022140857 \times 10^{23}\ mol^{-1}$

2. Escribe un programa en la que definas dos funciones que realicen lo siguiente:

   (a) Toma como argumentos un valor flotante y el numero 32 o 64. Entonces devuelve su *representación de punto flotante* (`números de maquina`) [1] de 32 o 64 bits.

   (b) Toma como argumentos un `número de maquina`[2] y el numero 32 o 64 que representa si es un número en estándar simple (32 bits) o doble (64 bits). Entonces devuelve su representación decimal.

[1] Para la *representación de punto flotante* (`número de maquina`) de valores usa una cadena de texto. Por ejemplo: `num =` "0 10000000 01011011111100001010100".

[2] Aquí también usa una cadena de texto.

3. **Factorial (valores flotantes)**

   Durante el curso, hemos escrito un par de programas para calcular e imprimir el factorial de un número ingresado por el usuario, usando valores *enteros* y no de *punto flotante*. Usa alguno de esos programas para calcular el factorial de 200 y modificalo para **usar variables de *punto flotante*** y calcula nuevamente el factorial de 200. ¿Qué es lo que encuentras?, explica.

4. **Ecuaciones cuadráticas**

   Considera una ecuación cuadrática $ax^2 + bx + c = 0$ que tiene soluciones reales.

   (a) Escribe un programa que tome como entrada tres números, a, b y c, e imprima las dos soluciones de la ecuación cuadrática $ax^2 + bx + c = 0$; usando la fórmula estándar:

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

   Utiliza tu programa para calcular las soluciones de $0.001x^2 + 1000x + 0.001 = 0$.

   (b) Existe otra forma de escribir las soluciones de una ecuación cuadrática.

   Demuestra que multiplicando la parte superior e inferior de la solución anterior por $-b \pm \sqrt{b^2 - 4ac}$, las soluciones también se pueden escribir como:

   $$x_{1,2} = \frac{2c}{-b \pm \sqrt{b^2 - 4ac}}.$$

   Agrega unas lineas a tu programa para imprimir también estos valores, además de los anteriores y entonces usa nuevamente el programa para resolver $0.001x^2 + 1000x + 0.001 = 0$.**¿Qué es lo que ves? ¿Cómo lo explicas?**.

   (c) Usando lo que has aprendido, escribe un nuevo programa que calcule con precisión ambas raíces de la ecuación cuadrática en todos los casos.

   Este es un buen ejemplo de cómo las computadoras no siempre funcionan como se espera. Si simplemente se aplica la fórmula estándar para la ecuación cuadrática, la computadora a veces obtendrá una respuesta incorrecta. En la práctica, el método que has desarrollado aquí es la forma correcta de resolver una ecuación cuadrática en una computadora, aunque es más complicado que la fórmula estándar. Si estuvieras escribiendo un programa que implicara resolver muchas ecuaciones cuadráticas, este método podría ser un buen candidato para convertirlo en una función: podrías poner los detalles del método dentro de dicha función para ahorrarte la molestia de seguirlo paso a paso, cada vez que tengas una nueva ecuación que resolver.

5. **Cálculo de derivadas**

   Supongamos que tenemos una función $f(x)$ y queremos calcular su derivada en un punto x_0. Esto lo podemos hacer con lápiz y papel si conocemos la forma matemática de la función, o podemos hacerlo en la computadora haciendo uso de la definición de la derivada:

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \lim_{\delta x \to 0} \frac{f(x_0 + \delta) - f(x_0)}{\delta x}.$$

   En la computadora no podemos tomar el límite cuando δx llega a cero, pero podemos obtener una aproximación razonable simplemente haciendo que δx sea pequeño.

   (a) Escribe un programa que defina una función `f(x)` y que devuelva el valor $x(x-1)$, luego calcula la derivada de la función en el punto $x_0 = 1$ usando la fórmula anterior con $\delta x = 10^{-2}$. Calcula analíticamente el valor real de la misma derivada y compáralo con la respuesta que da tu programa. Ambos no estarán perfectamente de acuerdo. **¿Por qué no?**

   (b) Repite el cálculo para $\delta x = 10^{-4}$, 10^{-6}, 10^{-8}, 10^{-10}, 10^{-12} y 10^{-14}. Deberías ver que la precisión del cálculo inicialmente mejora a medida que δx se hace más pequeña, pero luego vuelve a empeorar. **¿Por qué pasa esto?**

   Estudiaremos las derivadas numéricas con más detalle más adelante en el curso, donde examinaremos¸ técnicas para abordar estos problemas y maximizar la precisión de nuestros cálculos.