Práctica 1

Introducción

Con la realización de esta práctica se espera que el estudiante comprenda el efecto que los errores de redondeo tienen sobre los cálculos y como estos pueden conducir a resultados erróneos. Aunque existen muchas causas de errores en la ejecución de un programa, los ejercicios planteados en esta práctica abordan solo algunas de las más severas. Por otro lado, en ciertos casos se recurre a escenarios hipotéticos con el fin de visualizar las dificultades en un ambiente controlado que permita interpretar los resultados fácilmente. Con este tipo de ejercicios se espera que el estudiante sea capaz de extrapolar el resultado a un escenario real, donde el problema sigue siendo el mismo solo que la cantidad de dígitos manejados 'retrasa' su aparición. En este contexto el término retrasar puede hacer referencia al tiempo (se requieren muchas iteraciones antes de que aparezcan las dificultades) o al número de dígitos (las cifras significativas superan la cantidad de bits de almacenamiento en las primeras operaciones).

1 Causas de errores graves en computación

Para entender algunas de las causas más serias de errores en un programa de cómputo considere el siguiente escenario hipotético. Imagine que utiliza una computadora imaginaria que trabaja con números en el sistema decimal, de tal forma que la mantisa tiene cuatro dígitos decimales y la característica dos, el primero de los cuales es utilizado para el signo. Al sumar a estos seis bits el empleado para el signo del número se tendrá una palabra de 7 bits. Con esta computadora realice los cálculos que se piden a continuación.

a) **Suma de números muy distintos en magnitud**. Determine la suma de los números $x = 5$ y $y = 0.8712 \times 10^7$ y haga un análisis de los resultados obtenidos.

b) **Resta de números casi iguales**. Considere los números $a = 0.1943$ y $b = 0.1942$, use la computadora imaginaria para restarlos. Observe que la resta tiene un solo dígito significativo (¿por qué?). Suponga que la expresión aritmética es parte de un programa que requiere calcular el valor de $d = (a - b) \cdot c$, donde $c = 0.1 \times 10^5$ y halle dicho valor. ¿Qué ocurre si el cálculo de uno de los datos se realizó con un error muy pequeño? Por ejemplo, suponga que el valor calculado de a fue $a^* = 0.1944$. Determine el error absoluto y relativo en el cálculo de a, y calcule nuevamente el valor de d. ¿Qué ocurrió con el resultado? ¿Confiaría en la exactitud de esta máquina?

c) ***Overflow* o desbordamiento**. Algunas operaciones aritméticas con números válidos podrían dar lugar a números muy grandes como en este ejercicio. Dados los números $a = 0.2289 \times 10^7$ y $b = 0.2314 \times 10^8$, determine su producto usando la computadora imaginaria. ¿Qué resultado devolvería esta computadora? Ahora, tome $c = 0.2289 \times 10^{-7}$ y realice la división de b y c. ¿Cambia el resultado de la computadora en este caso? Justifique sus respuestas.

c) ***Underflow* o desbordamiento negativo**. El *underflow* puede aparecer en la multiplicación o la división y no suele ser tan serio como el *overflow*, por lo que la computadora casi nunca envía este tipo de mensajes de error. Sin embargo, en algunas situaciones puede conducir a errores serios aunque también se debe señalar que a veces es reparable.

Con el fin de que se familiarice con este problema suponga que como parte de un programa de cómputo nuestra máquina imaginaria debe calcular la expresión: $x = a \cdot b \cdot c$, donde $a = 0.1235 \times 10^{-4}$ $b = 0.9477 \times 10^{-6}$ y $c = 0.1942 \times 10^{5}$. ¿Qué respuesta devuelve este cálculo? Observe que al realizar cambios en el orden de los factores el resultado puede cambiar significativamente. Con base en esta observación proponga un cambio del orden y determine el valor de x. ¿Qué conclusiones puede sacar de este ejercicio?

d) **División entre un número muy pequeño**. Para ilustrar este problema suponga que en un programa la computadora imaginaria debe hallar el valor de la variable $x = a - b/c$ donde $a = 0.8749 \times 10^{6}$, $b = 0.1942 \times 10^{9}$ y $c = 0.2221 \times 10^{-3}$ son valores exactos. Suponga, además, que en un paso anterior la computadora cometió un error absoluto en el cálculo de c del orden de 0.0001 y que el valor almacenado es $c^* = 0.2220$. Determine los valores de x y $x^* = a - b/c^*$ y halle los errores absoluto y relativo de estos cálculos. ¿De qué tamaño son los errores original y el propagado? ¿Es confiable el resultado?

2 Estabilidad y convergencia

a) La ecuación de recurrencia:

$$I_n = 1 - nI_{n-1}, \qquad n = 2, 3, ... \tag{1}$$

donde $I_1 = 1/e$, permite aproximar la secuencia $I_n,\ n = 1, 2, ...$, definida mediante:

$$I_n = \int_0^1 x^n e^{x-1} dx \tag{2}$$

Escriba un programa que calcule la sucesión de iterados I_n para $n = 2 : N$ y úselo para aproximar I_{20}. Tome como aproximación inicial $I_1 = 1/e \approx 0.36787$. Argumente por qué los resultados obtenidos no son confiables.

b) La fórmula (1) se puede reescribir como:

$$I_{n-1} = \frac{1 - I_n}{n}, \qquad n = \ldots, 3, 2 \tag{3}$$

Use esta desigualdad con $I_{20} = 0.04762$ para aproximar el valor de I_9. Compare su resultado con el valor real: $I_9 = 0.0916$ (calculado con aritmética exacta usando 3 dígitos significativos). Haga un análisis del comportamiento del error al utilizar la fórmula (3). ¿Qué puede concluir acerca de la estabilidad del método?