Curso de Física Computacional

Acumulación de errores

M. en C. Gustavo Contreras Mayén

curso.fisica.comp@gmail.com

Números reales

El formato para un número real en una computadora difiere según el diseño de hardware y software. Los números reales se almacenan en el formato de punto flotante normalizado en binario. En precisión simple, se usan 4 bytes, es decir, 32 bits. Si se introduce un número decimal como dato, primero se convierte al binario más cercano en el formato normalizado:

$$(\pm 0.abbbbbbbb \ldots bbbb)_2 \times 2^2$$

donde *a* siempre es 1, cada *b* es un dígito binario $(0, 1)$ y *z* es un exponente que también se expresa en binario.

Distribución de dígitos

Existen 24 dígitos para la mantisa, incluyendo la a y las b

Los 32 bits se distribuyen de la siguiente manera:

- El primer bit se usa para el signo de la mantisa.
- Los siguientes 8 bits para el exponente z.
- Los últimos 23 para la mantisa.

11111111 11111111 11111111 11111111

En el formato de punto flotante normalizado, el primer dígito de la mantisa siempre es 1, por lo que no se almacena físicamente, por tanto, una mantisa de 24 bits, se almacena en 23.

Exponentes en binario

Si los 8 dígitos asignados al exponente se usan sólo para enteros positivos, el exponente puede representar desde 0 hasta $2^8 - 1 = 225$, aunque puede incluir a números negativos.

Para el manejo de exponentes positivos o negativos, el exponente en decimal es sesgado (o sumado) con 128 y después convertido en binario (completo a dos)

Por ejemplo, si el exponente es -3, entonces $-3 + 128 = 125$ que se convierte a binario y se almacena en los 8 bits.

Los exponentes que se pueden almacenar en 8 bits, van desde $0 - 128 = -128$ hasta $255 - 128 = 127$.

Errores por redondeo

Consideremos el cálculo de $1 + 0{,}00001$. Las representaciones binarias de 1 y $0{,}00001$ son, respectivamente:

$$\begin{aligned} (1)_{10} &= (0{,}1000\ 0000\ 0000\ 0000\ 0000\ 0000)_2 \times 2^1 \\ (0{,}00001)_{10} &= (0{,}1010\ 0111\ 1100\ 0101\ 1010\ 1100)_2 \times 2^{-16} \end{aligned}$$

La suma de estos dos números es:

$$\begin{aligned} & (1)_{10} + (0{,}00001)_{10} \\ &= (0{,}1000\ 0000\ 0000\ 0000\ 0101\ 0011\ 1110\ 0010\ 1101\ 0110\ 0)_2 \times 2^1 \end{aligned}$$

Sin embargo, éste número se redondea ya que la mantisa usa 24 bits, por lo que el número se guarda como

$$(0{,}1000\ 0000\ 0000\ 0000\ 0101\ 0100)_2 \times 2^1$$

que es equivalente a $(1{,}0000100136)_{10}$

Conclusión

Por lo que siempre que se sume 0.00001 a 1, el resultado agrega 0.0000000136 como error, al repetir 10 000 veces la suma de 0.00001 a 1, se genera un error de diezmil veces.