Cuando las computadoras se equivocan: exactitud y precisión en computación científica

Pablo F. Argibay

En cualquier curso en el cual se comparen las capacidades del cerebro humano con las de una computadora, inmediatamente surge la cuestión de que la computadora no puede resolver la cantidad de problemas que el cerebro resuelve en paralelo. Pero, ¡claro!, no hay duda de que la computadora es inmensamente superior en velocidad de resolución de cálculos. Intuitiva e inmediatamente aparece la idea de que además las computadoras son infalibles a la hora de resolver cuentas. La mágica sucesión de numeritos que aparecen en la pantalla se nos antoja exacta y carente de error por alguna característica "científicamente demostrable" para algunos, mágica para otros, finalmente de fe en la tecnología para muchos. Pero… ¿es esto así, e inexorablemente cierto? ¿Carecen las computadoras de la posibilidad de error en los cálculos?

EL ERROR DE DHAHRAN

Antes de la Guerra del Golfo, los sistemas de defensa antimisiles eran un concepto de guerra sin probar. El objetivo de los misiles norteamericanos Patriot era abatir los misiles Scud lanzados por Iraq sobre Israel y Arabia Saudita. Algunos vimos imágenes televisivas, en enero de 1991, que mostraban un Patriot interceptando y destruyendo un misil Scud iraquí lanzado sobre Arabia Saudita. Parecía que se abría la era de los sistemas "inteligentes" de defensa antiaérea, los cuales detectaban el misil enemigo, lo perseguían y lo destruían en una zona segura.

Sin embargo, a poco tiempo del bautismo del sistema, en febrero de 1991, un misil Scud iraquí alcanzó un cuartel norteamericano en Dhahran, Arabia Saudita, matando a 28 soldados.

Una investigación posterior mostró que el Scud no había sido interceptado, a causa de un error de cálculo efectuado por el software del sistema de defensa del Patriot. Los cálculos habían sido hechos con una precisión de 20 decimales, lo que a pesar de lo tremendamente exacto de la cifra decimal había llevado a un error infinitamente pequeño en lo inmediato pero que, acumulado en el reloj del sistema luego de 100 horas de funcionamiento, había llevado a un retraso de 0.36 segundos. Este error de aproximadamente un tercio de segundo, relativizado a la velocidad del Scud, condujo a un error de estima de posición de casi 600 metros. El radar detectó al Scud, pero dado el error de estimación de posición, el sistema no logró detectar la posición estimada para la intercepción. Como resultado, el Scud impactó en una barraca de soldados norteamericanos y mató a 28 de ellos.

ERROR Y PRECISIÓN EN ARITMÉTICA COMPUTACIONAL

Las computadoras no pueden manejar números con precisión infinita. La aproximación que hacen de los números se "empaquetan" en un número fijo de *bits* (dígitos binarios) o *bytes* (paquetes de 8 *bits*). En general el programador puede elegir diferentes tipos de representaciones o tipos de datos. En general los tipos de datos difieren en:

- Número de *bits* (tamaño)
- Tipo de representación:
  - Punto fijo (*fixed point*). Ejemplo en programación en C++: int
  - Punto flotante (*floating point*). Ejemplo: *float* o *double*

Misil Patriot (Fuente http://es.wikipedia.org/wiki/MIM-104_Patriot)

Unidad de Investigación en Procesos Cognitivos. Instituto de Ciencias Básicas y Medicina Experimental. Hospital Italiano de Buenos Aires
Correspondencia: pablo.argibay@hospitalitaliano.org.ar

Se supone que un número representado como entero tiene representación exacta.
En el caso que nos interesa para este artículo describiremos la representación en "punto flotante".

1. Representación en punto flotante (*floating point*)
En este tipo de representación los números se representan a través de tres parámetros:
- Un signo + o -, (S)
- Un exponente entero y exacto, (E)
- Una mantisa binaria, (M). El número (n) quedaría representado de la siguiente manera:

  • $n= S \times M \times b^{E-e}$

Donde b es la base de la representación (en general b= 2), y e es el sesgo del exponente, un entero constante para toda máquina y representación.

2. Tipos de error en aritmética de punto flotante
- Errores de redondeo

En la computadora, las operaciones aritméticas entre números de punto flotante no son exactas. La precisión de la computadora (ε_m), se define como el número más pequeño en punto flotante que, sumado al 1.0, produce un resultado diferente de 1.0. En la estandarización IEEE 754,[1] *float* tiene una ε_m aproximada de 1.19×10^{-7} y *double* 2.22×10^{-6}. Toda operación entre números de punto flotante tiene un error fraccional de al menos ε_m y este error se denomina "error de redondeo". El problema con los errores de redondeo es que estos se acumulan a medida que se acumulan los cálculos. Ejecutando N operaciones aritméticas el error esperado de redondeo suele estar al menos en el orden de $\sqrt{N} \times \varepsilon_m$, aunque este número no es preciso y depende de varios factores como el tipo de operación.

- Errores de truncado

Se dice que los errores de redondeo son característicos del hardware. Sin embargo, independientemente del hardware, existe un tipo de error dependiente del algoritmo o programa utilizado. Algunos algoritmos numéricos computan aproximaciones discretas para una cantidad continua deseada. Por ejemplo, una función puede ser evaluada sumando un número finito de términos en su serie infinita, en lugar de todos sus términos infinitos. En estos casos se puede buscar un parámetro ajustable como por ejemplo el número de puntos o términos tales que la respuesta "verdadera" es obtenida cuando dicho parámetro tiende al infinito. Todo cálculo en principio puede hacerse con una elección de un número finito pero lo suficientemente grande de dicho parámetro. La discrepancia entre la respuesta "verdadera" y la respuesta obtenida con nuestro cálculo se denomina "error de truncado". En términos de programación la ventaja de este tipo de errores es que está "enteramente" bajo control del programador.
Ejemplo (Faires, 2002): π tiene una expresión decimal infinita tal que $\pi= 3.14159265\ldots$
En forma decimal: $\pi= 0.314159265\ldots \times 10^1$
El truncamiento en "punto flotante" a 5 cifras sería:
$fl(\pi)= 0.314159265\ldots \times 10^1= 3.1415$
Y el redondeo sería:
$fl(\pi)= (0.31415 + 0.00001) \times 10^1= 3.14\mathbf{16}$

> En 1985 el IEEE publicó el *Binary Floating Point Arithmetic Standard 754-1985* en el que se especifican los formatos para las precisiones "simple", "doble" y "extendida". Estos formatos han sido utilizados por los fabricantes de computadoras para el desarrollo de hardware. El coprocesador numérico de las computadoras personales (PC) utiliza una representación de 64 *bits* para números reales. Estos números (real largo) tienen un primer *bit*, indicador de signo, un exponente de 11 *bits* (la característica) y una fracción binaria de 52 *bits*, llamada mantisa. La base del exponente es 2. Como 52 dígitos binarios corresponden a 16 y 17 dígitos decimales, podemos suponer que un número representado en este sistema tiene al menos 16 cifras decimales de precisión.

En síntesis, la ciencia y sus parientes, la ingeniería y la tecnología, intentan describir el mundo y sus fenómenos mediante modelos matemáticos. La medicina no es ajena a este tipo de descripciones. A pesar de lo acotado de sus variables, los modelos permiten un conocimiento aproximado de la realidad y, a partir de la situación hipotética que plantean, estudiar, predecir y analizar la evolución de cualquier sistema. Sin ir más lejos, el problema antes mencionado del sistema de defensa "Patriot", parte de la modelización de una situación en la que se modelizan diferentes variables relacionadas con el misil enemigo y su intercepción (velocidades, distancias, características de emisión térmica, etc.). La contrastación final del modelo con la realidad es esta última: el éxito o el fracaso de las predicciones efectuadas por el modelo. La matemática aplicada es la rama de la matemática que se dedica a buscar y aplicar las herramientas más adecuadas a los problemas basados en estos modelos.
Desafortunadamente, no siempre es posible aplicar métodos analíticos clásicos por diferentes razones:

1 IEEE: siglas del inglés que corresponden al Instituto para ingenieros eléctricos y electrónicos.

- No son plausibles con el modelo concreto.
- Su aplicación resulta excesivamente compleja.
- La solución formal es tan complicada que hace imposible cualquier interpretación posterior.

Aquí aparecen las técnicas de análisis numérico que, mediante la aplicación de técnicas de cálculo, conducen a soluciones numéricas. Para este tipo de cálculos es necesario emplear computadoras. De hecho, sin el desarrollo que se ha producido en el campo de la informática resultaría difícilmente imaginable el nivel actual de utilización de las técnicas numéricas en ámbitos cada día más diversos. Sin embargo, como hemos esbozado más arriba, entre la aritmética computacional finita y las variables numéricas con las que se presenta la vida real existen diferencias que, acumuladas, conducen a errores a veces triviales y a veces letales o tremendamente costosos en términos económicos o sociales. El análisis, la prevención y la eventual corrección de dichos errores es un paso fundamental a la hora de desarrollar programas de computación científica.

BIBLIOGRAFÍA

- Burden RL, Faires JD. Análisis numérico. México, DC: Cengage; 2002.
- Díaz Villanueva W. Métodos numéricos [Internet]. Universitat de València. Departament d'Informàtica; 1998 May 11. [Consulta: 15/10/2010]. Disponible en: http://www.uv.es/diaz/mn/fmn.html
- Press WH, Tukolsky SA, Vetterling WT, et al. Numerical recipes: the art of scientific computing. New York: Cambridge University Press; 2007.
- Stewart C. US Aussie spy base relevations [Internet].[place unknown]: World history archives: the retrospective history of the Commonwealth of Australia. Hartford Web: 1999 Feb 17. [Consulta 15/10/2010]. Disponible en: http://www.hartford-hwp.com/archives/24/167.html
- Wikipedia. MIM-104 Patriot [Internet]. [Consulta: 15/10/2010]. Disponible en: http://es.wikipedia.org/wiki/MIM-104_Patriot.