

¡Ojos que no ven, computadoras que mienten!

Dr. Humberto Madrid de la Vega
Centro de Investigación en Matemáticas Aplicadas
Universidad Autónoma de Coahuila
hmadrid@gmail.com

Resumen

El uso de la tecnología computacional en el proceso de enseñanza aprendizaje del Cálculo plantea un problema singular: simular procesos continuos con herramientas finitas. Si el programador no conoce la aritmética computacional, y sus diferencias con la aritmética del continuo, puede cometer errores mayúsculos. También tienen que replantearse cuidadosamente los conceptos básicos del cálculo como el concepto de límite y de derivada. En esta plática veremos algunos ejemplos de errores que se pueden cometer por el desconocimiento de la aritmética computacional y su correspondiente explicación, así como ejemplos de cómo se deben replantear conceptos y teoremas del Cálculo.

1 Introducción

La computadora se ha vuelto una herramienta indispensable en muchas actividades. Una de ellas es la educación, en donde se utiliza para tratar de mejorar el proceso de enseñanza aprendizaje. Sin embargo, esto no está exento de problemas, particularmente cuando el tema es el del Cálculo Diferencial. Uno de los problemas principales es que en Cálculo se trabaja con los números reales (el continuo) y con procesos infinitos. En cambio la computadora trabaja con una aritmética finita, la cual difiere de la aritmética real. Ignorar esta diferencia puede causar efectos desastrosos. En este trabajo mostraremos algunos ejemplos que pueden conducir a los estudiantes a concepciones erróneas o a situaciones que más que aclarar pueden causar confusión.

2 Algunos desastres atribuibles a computación numérica errónea

A continuación se presentan algunos ejemplos de la vida real y de lo que pasa cuando los algoritmos numéricos no son aplicados correctamente.

2.1 Falla del misil PATRIOT

El 25 de febrero de 1991, durante la Guerra del Golfo, en Dharan, Arabia Saudita, un misil Patriot norteamericano falló al intentar interceptar un misil Iraquí y causó la muerte de 28 soldados norteamericanos, hiriendo a unas 100 personas más. Una investigación sobre el caso concluyó que el problema se debió al manejo ineficiente de los errores de redondeo. Información adicional puede encontrarse en <http://www.ima.umn.edu/~arnold/disasters/patriot.html>



2.2 La explosión del Ariane 5

El 4 de junio de 1996, un inmanejable cohete Ariana 5 lanzado por la Agencia Espacial Europea, explotó justamente después de 40 minutos de haber sido lanzado de Kourou, Guyana Francesa.



El cohete estaba en su primer viaje después de 10 años de desarrollo, el cual costó **siete billones de dólares**. El costo de la destrucción del cohete fue valuado en

quinientos millones de dólares. Una comisión investigadora concluyó que el problema fue causado por un mal diseño de software que ocasionó un *desbordamiento numérico*. Más información en <http://www.ima.umn.edu/~arnold/disasters/ariane.html>

3 Otros desastres

Mostraremos algunos ejemplos de desastres que tienen relación con el tema que nos ocupa, es decir, el Cálculo.

3.1 Un ejemplo clásico

La ecuación $4x - 1 = x$ tiene como solución $x = 1/3$, ya que

$$4 \left(\frac{1}{3} \right) - 1 = \frac{1}{3}$$

así que si al lado derecho lo multiplicamos por 4 y le restamos 1, obtendremos de nuevo $1/3$, es decir

$$4 \left(\frac{4}{3} - 1 \right) - 1 = \frac{1}{3}$$

Repitiendo

$$4 \left(4 \left(\frac{4}{3} - 1 \right) - 1 \right) - 1 = \frac{1}{3}$$

$$4 \left(4 \left(4 \left(\frac{4}{3} - 1 \right) - 1 \right) - 1 \right) - 1 = \frac{1}{3}$$

Elemental ¿verdad? Ahora hagamos lo mismo en una computadora o en una calculadora:

$$\begin{array}{rcl}
 \frac{4}{3} - 1 & = & .33333333333333 \\
 & \swarrow & \\
 4 \left(\frac{4}{3} - 1 \right) - 1 & = & .33333333333333 \\
 & \swarrow & \\
 4 \left(4 \left(\frac{4}{3} - 1 \right) - 1 \right) - 1 & = & .33333333333332 \\
 & \swarrow & \\
 4 \left(4 \left(4 \left(\frac{4}{3} - 1 \right) - 1 \right) - 1 \right) - 1 & = & .33333333333329 \\
 \dots & = & .33333333333314 \\
 & & .333333333333258 \\
 & & .333333333333030 \\
 & & .333333333332121 \\
 & & .3333333333328483
 \end{array}$$

...y más delante

.33203125
 .328125
 .3125
 .25
 .0
 -1
 -5
 -21
 -85
 ⋮

¿Qué estará pasando? Se supone que el lado derecho debería ser siempre .3333333333333333. Antes de responder esta pregunta, veamos el extraño caso de la derivada.

3.2 Aproximación a la derivada

Recordemos que la derivada de una función $f(x)$ es

$$f'(x) = \lim_{h \rightarrow \infty} \frac{f(x+h) - f(x)}{h}$$

así que podemos aproximar $f'(x)$ mediante

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

tomando valores de h cada vez más pequeños

Ejemplo:

Consideremos la función $f(x) = e^x - 1$, queremos aproximar $f'(x)$ para $x = 1$. Como sabemos $f'(1) = e$

Incremento	Aproximación	Error
1	4.6707742704716	0.95249244201256
10^{-1}	2.85884195487388	0.140560126414833
10^{-2}	2.73191865578708	0.136368273280345
10^{-3}	2.71964142253278	0.00135959407373587
10^{-4}	2.71841774707848	0.000135918619437625

continúa ...

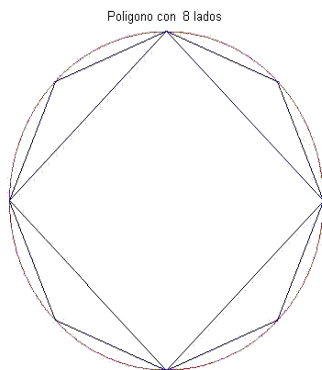
... continuación:

Incremento	Aproximación	Error
10^{-5}	2.71829 541991231	$1.35914532628512 \times 10^{-5}$
10^{-6}	2.718283 18698652	$1.35852747940035 \times 10^{-6}$
10^{-7}	2.71828196 396484	$1.35505795473279 \times 10^{-7}$
10^{-8}	2.718281777 44737	$-5.10116726637477 \times 10^{-8}$
10^{-9}	2.7182815998 1169	$-2.28647357047862 \times 10^{-7}$
10^{-10}	2.718278935 27643	$-2.89318261614824 \times 10^{-6}$
10^{-11}	2.718270053 49223	$-1.17749668131495 \times 10^{-5}$
10^{-12}	2.718270053 49223	$-1.17749668127054 \times 10^{-5}$
10^{-13}	2.713385072 18388	-0.00489675627516339
10^{-14}	2.664535259 10038	-0.0537465693586703
10^{-15}	2.664535259 10038	-0.0537465693586698
10^{-16}	0 !!	-2.71828182845905
10^{-17}	0	-2.71828182845905
10^{-18}	0	-2.71828182845905
10^{-19}	0	-2.71828182845905

En la segunda columna se muestran en negritas los dígitos correctos que va adquiriendo la aproximación. De la tabla se deduce que $|e| = 0!$ La teoría dice que $e = 2.71828182845905 \dots$, la computadora dice que $e = 0$. ¿A quién le creemos?

3.3 Arquímedes: Aproximación a π

El método de Arquímedes consiste en tomar un círculo de diámetro 1, es decir, con perímetro π , inscribir un cuadrado en este círculo y después construir polígonos con 8, 16, 32, 64, $\dots$, 2^n , $\dots$ lados, de forma tal que el perímetro de estos polígonos va aproximando a π .



Una fórmula recursiva para el perímetro del polígono con 2^n lados es:

$$p_{n+1} = 2^n \sqrt{2 \left(1 - \sqrt{1 - \left(\frac{p_n}{2^n} \right)^2} \right)}, \quad n = 3, 4, \dots$$

donde $p_2 = 2\sqrt{2}$ (Kahaner, Moler, Nash, 1988, p.).

He aquí los resultados:

No. lados		Aproximación	Error
$2^2 =$	4	2.82842712474619	0.313165528843603
$2^3 =$	8	2.82842712474619	0.313165528843603
$2^4 =$	16	3.06146745892072	0.0801251946690744
$2^5 =$	32	3.12144515225805	0.0201475013317403
$2^6 =$	64	3.13654849054594	0.00504416304385247
$2^7 =$	128	3.14033115695474	0.00126149663505393
$2^8 =$	256	3.14127725093276	0.000315402657036223
$2^9 =$	512	3.14151380114415	$7.8852445647648 \times 10^{-5}$
$2^{10} =$	1024	3.14157294036788	$1.97132219104112 \times 10^{-5}$
$2^{11} =$	2048	3.14158772527996	$4.92830983223058 \times 10^{-6}$
$2^{12} =$	4096	3.14159142150464	$1.23208515789841 \times 10^{-6}$
$2^{13} =$	8192	3.14159234561108	$3.07978716307389 \times 10^{-7}$
$2^{14} =$	16384	3.141592576545	$7.70447887710191 \times 10^{-8}$
$2^{15} =$	32768	3.14159263346325	$2.01265448751542 \times 10^{-8}$
$2^{16} =$	65536	3.14159265480759	$-1.21779608619477 \times 10^{-9}$
$2^{17} =$	131072	3.14159264532122	$8.26857782243451 \times 10^{-9}$
$2^{18} =$	262144	3.14159260737572	$4.62140734569516 \times 10^{-8}$
$2^{19} =$	524288	3.14159291093967	$-2.57349879628777 \times 10^{-7}$
$2^{20} =$	1048576	3.14159412519519	$-1.47160539798463 \times 10^{-6}$
$2^{21} =$	2097152	3.14159655370482	$-3.90011502648946 \times 10^{-6}$
$2^{22} =$	4194304	3.14159655370482	$-3.90011502648946 \times 10^{-6}$
$2^{23} =$	8388608	3.14167426502176	$-8.16114319643901 \times 10^{-5}$
$2^{24} =$	16777216	3.1418296818892	-0.000237028299408415
$2^{25} =$	33554432	3.14245127249413	-0.000858618904340691
$2^{26} =$	67108864	3.14245127249413	-0.000858618904340691
$2^{27} =$	134217728	3.16227766016838	-0.0206850065785864
$2^{28} =$	268435456	3.16227766016838	-0.0206850065785864
$2^{29} =$	536870912	3.46410161513775	-0.322508961547961
$2^{30} =$	1073741824	4	-0.858407346410207
$2^{31} =$	2147483648	0	3.14159265358979
$2^{32} =$	4294967296	0	3.14159265358979
$2^{33} =$	8589934592	0	3.14159265358979

La misma historia: ¡La computadora nos dice que $\pi = 0$!. También obsérvese que aunque por construcción los polígonos están inscritos en el círculo, en algunos casos las aproximaciones son mayores que π .

4 Aritmética Computacional

Para poder entender lo que ha ocurrido en los ejemplos precedentes, es conveniente echar un vistazo, en forma simplificada, a la aritmética que emplean las computadoras. Técnicamente es conocida como Aritmética de Punto Flotante. Los números reales son representados dentro de la computadora mediante un conjunto finito de números que se pueden escribir como

$$\pm .d_1d_2\dots d_k \times \beta^e \quad (1)$$

donde k es el número de dígitos, β es la base, y e es un exponente con las siguientes propiedades: $d_1 > 0$, $d_1, d_2, \dots, d_k < \beta$, e es un entero en el intervalo $[-m, M]$.

La base β típicamente es 2 para la mayor parte de las computadoras, pero también son usadas $\beta = 8, 10, 16$. Los lenguajes de programación como FORTRAN, BASIC, PASCAL, C, C++, pueden trabajar con precisión simple o precisión doble. En esta última, $\beta = 2$, pero en términos de aritmética decimal $k \approx 16$, $\min f \approx 2.2251 \times 10^{-308}$, $\max f \approx 1.7977 \times 10^{308}$.

Los elementos de la aritmética de punto flotante, llamémosla F , está formada por los números de la forma (1) más el número cero. Los elementos $f \in F$ distintos de cero, satisfacen $0 < \min f \leq f \leq \max f$. A cada $x \in \mathbb{R}$ se le asocia un elemento $fl(x) \in F$, excepto a $x \in \mathbb{R}$ tales que $|x| > \max f$ en cuyo caso se dice que rebasa el rango de la aritmética de punto flotante, popularmente se le conoce a esta situación como *overflow*. Otra excepción se presenta cuando $|x| < \min f$, se trata del fenómeno de *underflow* y por lo general a esta x se le toma como 0. Una explicación más amplia puede ser encontrada en http://en.wikipedia.org/Category:Computer_arithmetic.

En corto, a (casi) todo número real se le asocia un elemento de la aritmética de punto flotante.

$$x \rightarrow fl(x)$$

Aunque F trata de emular a $\mathbb{R}$, es claro que esto no es posible pues tratamos de reproducir el continuo con un conjunto finito. Sin embargo la aritmética computacional es muy poderosa, aunque por supuesto tiene sus deficiencias. Para empezar las operaciones de suma y producto en F no son cerradas. Pensemos por ejemplo en dos elementos de F , cada uno con k dígitos, claramente el producto de ellos tiene más de k dígitos, por lo que al producto se le tiene que representar por otro elemento de F , es decir, en cada operación aritmética se comete un error pequeño.

Por así decirlo, esto no es grave en sí, con una excepción que se presenta al restar dos números muy parecidos, en este caso el error que se comete causa cálculos erróneos subsecuentes. Vamos a ilustrar esto con un ejemplo.

Una identidad matemática como:

$$\frac{(1+h^2)-1}{h} = 2+h$$

no necesariamente es una identidad en la computadora

h	$2+h$	$\frac{(1+h)^2-1}{h}$
10^{-1}	2.100000000000000	2.100000000000000
10^{-2}	2.010000000000000	2.010000000000000
10^{-3}	2.001000000000000	2.000999999999970
10^{-4}	2.000100000000000	2.000099999999917
10^{-5}	2.000010000000000	2.00001000001393
10^{-6}	2.000001000000000	2.00000099992437
10^{-7}	2.000000100000000	2.00000010108781
10^{-8}	2.000000010000000	1.99999998784506
10^{-9}	2.000000001000000	2.00000016548074
10^{-10}	2.000000000100000	2.00000016548074
10^{-11}	2.000000000010000	2.00000016548074
10^{-12}	2.000000000001000	2.00017780116468
10^{-13}	2.000000000000100	1.99840144432528
10^{-14}	2.000000000000001	1.99840144432528
10^{-15}	2.0	2.22044604925031
10^{-16}	2.0	0.0

La razón de este comportamiento es que para h suficientemente pequeña se presenta el fenómeno de la cancelación catastrófica al evaluar

$$(1+h)^2 - 1$$

y para h todavía más pequeña, pero aún distinta de cero, en la computadora $1+h=1$, por lo cual $(1+h)^2 - 1$ es evaluada como 0.

Esto último lo podemos explicar de esta manera. La expresión $1+h$ va tomando los valores $.11 \times 10$, $.101 \times 10$, $.1001 \times 10$, $\dots$. Llega un momento en el que se agotan los lugares para almacenar los dígitos y el 1 al final queda fuera y la computadora almacena el resultado como 1. Eso ocurre para $h = 10^{-16}$, $10^{-17}, \dots$

Observemos que, por ejemplo, $h = 10^{-16}$, es distinto de cero en F . De hecho, lo mismo ocurre para $h < 2.2204 \times 10^{-16}$. Esto significa que hay una infinidad de números que son distintos de cero, pero actúan como neutros en la suma.

5 Regreso a los ejemplos

5.1 El ejemplo aritmético

$x = 1/3$ es almacenado como $\bar{x}$, es decir $x = \bar{x} + e$. Suponiendo que ya no se presentan más errores, tenemos entonces

$$\begin{aligned}x &= \frac{1}{3} \\ \bar{x} &= x - e \\ \bar{x}_1 &= 4\bar{x} - 1 \\ \bar{x}_1 &= 4(x - e) - 1 \\ &= (4x - 1) - 4e \\ &= x - 4e \\ x - \bar{x}_n &= 4^n e \\ \bar{x}_n &= x - 4^n e\end{aligned}$$

Observemos que $4^n e$ en algún será grande, aunque el valor de e sea pequeño. De hecho $4^n e \rightarrow \infty$ y en algún momento $x - 4^n e$ debe ser negativo. Esto explica el fenómeno observado.

Entonces en este ejemplo, el error que se comete al almacenar $x = 1/3$ va creciendo en cada paso, pero crece implacablemente. Se dice que este proceso es inestable.

Sin embargo, es conveniente observar que ese tipo de error no siempre crece. Veamos, podemos reescribir $4x - 1 = x$ como $x = (x + 1)/4$, así que su solución es también $x = 1/3$. Entonces

$$\begin{aligned}x &= \frac{x+1}{4} = \bar{x} + e ; & \bar{x} &= x - e \\ \bar{x}_1 &= \frac{1}{4}(\bar{x} + 1) = \frac{1}{4}(x - e + 1) = x - \frac{1}{4}e \\ \bar{x}_n &= x - \frac{1}{4^n}e\end{aligned}$$

Es decir, ¡el error disminuye en cada paso! Este proceso se dice que es estable. La moraleja es que los errores no siempre se acumulan.

5.2 Aproximación a la derivada

Recordemos que

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

Si f es continua para h suficientemente pequeña, $x+h \approx x$, así que se presenta el fenómeno de cancelación catastrófica; y para h todavía más pequeña, $f(x+h) - f(x)$ es calculada como 0.

Esto nos indica que para poder ilustrar la aproximación a la derivada, h no debe ser ni muy grande ni demasiado pequeña. ¿Cuál es el valor apropiado para h ? Eso depende del comportamiento de $f(x)$, en concreto, de su segunda derivada (Kahaner, Moler, Nash, 1988, p. 29).

5.3 Aproximación a π

Recordemos que

$$p_{n+1} = 2^n \sqrt{2 \left(1 - \sqrt{1 - \left(\frac{p_n}{2^n} \right)^2} \right)}$$

para $n = 3, 4, \dots$ donde $P_2 = 2\sqrt{2}$.

También se presenta la cancelación catastrófica. La razón es que P_n es una cantidad acotada por π , así que $P_n/2^n$ tiende a 0; en consecuencia, en

$$1 - \sqrt{1 - \left(\frac{P_n}{2^n} \right)^2}$$

se presenta dicha cancelación y eventualmente esta expresión es evaluada como 0.

Un remedio

En (Kahaner, Moler, Nash, 1988, p. 38) se sugiere reescribir la fórmula usada como dos sucesiones

$$r_{n+1} = 2 \left(1 - \sqrt{1 - \left(\frac{P_n}{2^n} \right)^2} \right)$$

$$p_{n+1} = 2^n \sqrt{r_{n+1}}$$

$$r_{n+1} = \frac{r_n}{2} + \sqrt{4 - r_n}$$

De esta manera $P_n \rightarrow \pi$ cuando $n \rightarrow \infty$ sin el problema de la cancelación.

5.4 Conclusiones

La aritmética computacional es desconocida por la mayoría de los usuarios de computadoras, incluso es poco conocida por la mayoría de los programadores. Hemos querido mostrar que este desconocimiento puede resultar en situaciones desafortunadas.

Esto es particularmente delicado cuando se trata de elaborar software educativo. Por supuesto los detalles numéricos finos no se le ocurren a los profesionales promedio y no se trata de que los que generen este software sean especialistas en Análisis Numérico, pero es importante que se recurra a especialistas en esa área.

Referencias

- Dahlquist, G., Björck, A. 2008 *Numerical Methods in Scientific Computing*, Volume 1. SIAM.
- Kahaner, D., Moler, C. Nash, S. 1988 *Numerical Methods and Software*. Prentice Hall.
- Overtone, M.L. *Cómputo numérico con aritmética de punto flotante IEEE. (Con un teorema, una regla empírica y ciento un ejercicios)*. Aportaciones Matemáticas 19. SIAM-SMM. 2002
- Mederos, O.B. y García, I.M. *La representación en punto flotante de los números reales. Flujo ascendente de información*. UCLV. Cuba. 1997