# Capítulo 1

# Aritmética finita y análisis de errores

> "la formación ingenieril que recibí me enseñó a tolerar aproximaciones, y pude ver que aún teorías basadas en aproximaciones podían tener algunas veces una suma considerable de belleza en ellas (...) Pienso que si no hubiera tenido esa formación ingenieril, no habría tenido ningún éxito con el tipo de trabajo que hice después, porque era realmente necesario alejarse del punto de vista de que uno debe ocuparse sólo de resultados que puedan ser deducidos lógicamente de leyes exactas conocidas que uno aceptara, en las que tuviera una fe incondicional".
>
> *P. A. M. Dirac, (1902–1984)*

Todo cálculo numérico introduce errores. Esta afirmación, que puede parecer desalentadora, es en realidad la premisa que hace posible el análisis numérico como disciplina: si los errores son inevitables, el objetivo es entenderlos, controlarlos y, cuando sea posible, explotarlos para obtener resultados útiles con el menor esfuerzo computacional.

**Un ejemplo motivador.** El 25 de febrero de 1991, durante la Guerra del Golfo, una batería de misiles Patriot estadounidense en Dhahran (Arabia Saudí) no logró interceptar un misil Scud iraquí. El Scud impactó en un cuartel, causando 28 muertos y casi cien heridos. La causa del fallo fue un error de

aritmética de coma flotante: el sistema almacenaba el tiempo en décimas de segundo como número entero y lo convertía a segundos multiplicando por $\frac{1}{10}$, que no es exactamente representable en binario. El error por décima de segundo era de 0,000 000 095 segundos, prácticamente cero. Pero el sistema llevaba 100 horas encendido: $100 \times 3600 \times 10 = 3\,600\,000$ décimas de segundo, y el error acumulado era de 0,34 segundos. En ese tiempo, el Scud, que viajaba a 1,7 km/s, había recorrido 578 metros: fuera de la ventana de intercepción del radar.

Este ejemplo ilustra que no hay errores «despreciables» en abstracto. Un error relativo de $9{,}5 \times 10^{-8}$ que se acumula durante $3{,}6 \times 10^{6}$ pasos produce un error absoluto que puede costar vidas. Lo relevante no es la magnitud absoluta del error, sino su tamaño respecto al problema que se está resolviendo.

Este capítulo tiene un doble propósito. El primero es práctico: familiarizarse con la aritmética de coma flotante que emplean los ordenadores y aprender a identificar cuándo un cálculo puede ser numéricamente peligroso. El segundo es conceptual: distinguir con precisión entre el *condicionamiento* de un problema y la *estabilidad* de un algoritmo, dos ideas que a menudo se confunden, lo que lleva a diagnósticos equivocados.

Una advertencia previa. En este capítulo las fórmulas son pocas y las ideas son muchas. La tentación natural es aprenderse las fórmulas de error sin entender de dónde vienen ni cuándo aplican. Esa estrategia es ineficiente: los resultados de este capítulo no se memorizan, se comprenden, y una vez comprendidos resultan casi obvios. El lector que sienta que algo no le cuadra debería detenerse, porque casi siempre hay una razón para esa incomodidad.

## 1.1. Representación de números en el ordenador

Los ordenadores trabajan con conjuntos finitos de números. Esto es una consecuencia inevitable de que la memoria es finita, no un defecto de diseño que pudiera corregirse con más hardware. Para la mayoría de los cálculos científicos se usa el estándar IEEE 754, que define con precisión el conjunto de números representables y las reglas para redondear los demás.

**DEFINICION 1.1 (Sistema de coma flotante).** Un **sistema de coma flotante** $\mathbb{F}(\beta, t, e_{\text{mín}}, e_{\text{máx}})$ es el conjunto de números reales de la forma

$$x \;=\; \pm\,(0.d_1 d_2 \cdots d_t)_\beta \times \beta^e,$$

donde $\beta \geq 2$ es la **base**, $t \geq 1$ es la **longitud de la mantisa**, cada $d_i \in \{0, \ldots, \beta - 1\}$, $d_1 \neq 0$ (condición de **normalización**), y $e \in [e_{\text{mín}}, e_{\text{máx}}]$ es el **exponente** entero. El cero se representa convencionalmente.

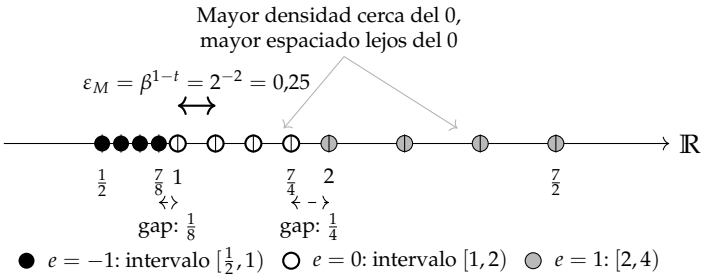



Figura 1.1: Distribución de un sistema de coma flotante simplificado $\mathbb{F}(2, 3, -1, 2)$ (base 2, mantisa de 3 bits, exponente en $\{-1, 0, 1, 2\}$). Los puntos flotantes no están uniformemente distribuidos en $\mathbb{R}$: hay la misma cantidad de números en cada biela $[2^e, 2^{e+1})$, pero con separación que se duplica al pasar a la siguiente. El épsilon de máquina $\varepsilon_M = 2^{1-t} = 2^{-2} = 0{,}25$ es la distancia de 1 a su vecino más próximo por la derecha. En doble precisión IEEE 754 ($t = 53$), $\varepsilon_M = 2^{-52} \approx 2{,}22 \times 10^{-16}$.

En el estándar IEEE 754 de doble precisión, que es el que usa MATLAB y la mayor parte del software científico, tenemos $\beta = 2$, $t = 53$ (52 bits de mantisa explícita más el *bit oculto*), $e_{\text{mín}} = -1022$ y $e_{\text{máx}} = 1023$. Con estos parámetros, los números positivos representables oscilan entre aproximadamente $2{,}2 \times 10^{-308}$ y $1{,}8 \times 10^{308}$.

***El bit oculto.*** *En base binaria, la condición de normalización $d_1 \neq 0$ implica que $d_1 = 1$ siempre. No hace falta almacenar ese dígito: el hardware lo sabe de antemano. Esto equivale a ganar un bit de mantisa de forma gratuita. Eso es exactamente lo que hace IEEE 754. La mantisa tiene 52 bits explícitos pero 53 bits efectivos. Cincuenta y tres*

*bits de mantisa equivalen a unos 15–16 dígitos decimales. Es decir: si los datos de un problema tienen más de 16 cifras significativas, o si el resultado tiene una incertidumbre inherente mayor que $10^{-15}$ en términos relativos, la aritmética de doble precisión es transparente, no introduce error apreciable. El problema surge cuando los cálculos amplían los errores de forma inesperada.*

**DEFINICION 1.2 (Épsilon de máquina).** El **épsilon de máquina** $\varepsilon_M$ es la distancia entre el número 1 y el siguiente número de coma flotante mayor que 1. En doble precisión, $\varepsilon_M = 2^{-52} \approx 2{,}22 \times 10^{-16}$.

**Nota.** En la bibliografía coexisten al menos dos definiciones de épsilon de máquina. La aquí adoptada, que sigue a Higham (2002), coincide con el resultado de `eps` en MATLAB. Otra definición frecuente toma como épsilon el menor número positivo tal que $\mathrm{fl}(1+\varepsilon) > 1$, que con redondeo al más próximo da $\varepsilon_M/2$. La diferencia es un factor dos, suficiente para causar confusión si no se especifica la definición adoptada.

La importancia práctica del épsilon de máquina es que establece el error relativo inevitable al representar un número real en coma flotante.

**PROPOSICION 1.1 (Cota del error de redondeo).** Si $x$ es un número real no nulo representable sin desbordamiento, entonces el número de coma flotante más próximo $\mathrm{fl}(x)$ satisface

$$\frac{|\,\mathrm{fl}(x) - x\,|}{|x|} \;\leq\; \frac{\varepsilon_M}{2}.$$

*Demostración.* Sea $e$ el exponente tal que $|x| \in [\beta^e, \beta^{e+1})$. Los números de coma flotante en ese intervalo están separados uniformemente por $u = \beta^{e-t+1}$ (unidad en el último lugar). El redondeo al más próximo introduce un error absoluto menor o igual a $u/2$. El error relativo es, por tanto,

$$\frac{u/2}{|x|} \;\leq\; \frac{u/2}{\beta^e} \;=\; \frac{\beta^{e-t+1}/2}{\beta^e} \;=\; \frac{\beta^{1-t}}{2} \;=\; \frac{\varepsilon_M}{2},$$

donde en el último paso hemos usado $\varepsilon_M = \beta^{1-t}$ (que en base 2 con $t = 53$ da $2^{-52}$). ⫻

Esta cota no depende de $x$. El error relativo al redondear es siempre menor que $\varepsilon_M/2$, independientemente de la magnitud de $x$. Esto es la gran virtud de la representación en coma flotante frente a la coma fija: el error es *relativo*, no absoluto.

## 1.2. Errores: taxonomía y propagación

En un cálculo real, los errores tienen varias fuentes, y es útil distinguirlas desde el principio porque sus remedios son distintos.

**DEFINICION 1.3 (Tipos de error).**

- **Error de modelización:** la diferencia entre el fenómeno real y el modelo matemático que lo describe.
- **Error de datos:** la imprecisión en los datos de entrada, debida a mediciones o a representación en coma flotante.
- **Error de truncamiento:** la diferencia entre la solución exacta del modelo matemático y la solución del algoritmo, debida a haber sustituido un proceso infinito (suma de serie, límite, integral) por uno finito.
- **Error de redondeo:** la acumulación de errores introducidos por la aritmética de coma flotante en cada operación.

Los dos últimos son los propios del cálculo numérico, pero son de naturaleza muy distinta. El error de truncamiento depende del algoritmo elegido y puede reducirse aumentando el esfuerzo computacional. El error de redondeo depende de la aritmética de la máquina y es, en cierto sentido, inevitable. El análisis numérico estudia cómo gestionar ese equilibrio.

**DEFINICION 1.4 (Error absoluto, relativo y cifras significativas).** Sean $x$ el valor exacto de una cantidad y $\tilde{x}$ una aproximación. El **error absoluto** es $|\tilde{x} - x|$ y el **error relativo** es $|\tilde{x} - x| \,/\, |x|$ (para $x \neq 0$).
Se dice que $\tilde{x}$ tiene $n$ **cifras significativas exactas** si el error relativo es menor que $0{,}5 \times 10^{-n}$.

***¿Por qué el error relativo importa más?*** *Si el error al medir la distancia Tierra-Luna es de 100 metros, eso es extraordinariamente preciso. Si el error al medir la altura de una persona es de 100 metros, es absurdo. El error absoluto solo tiene sentido en relación con la magnitud que se mide, y eso es exactamente lo que captura el error relativo.*

### 1.2.1. Propagación de errores y cancelación catastrófica

Cuando operamos con cantidades que contienen errores, los errores se propagan. El caso crítico es la resta de números cercanos.

**EJEMPLO 1.1 (Cancelación catastrófica).** Consideremos el cálculo de $f(x) = \sqrt{x+1} - \sqrt{x}$ para $x = 10^6$. En aritmética exacta,

$$f(10^6) = \sqrt{1\,000\,001} - 1000 \approx 5 \times 10^{-7}.$$

En coma flotante con 8 dígitos decimales de mantisa:

$$\sqrt{1\,000\,001} \approx 1000{,}0005000 \quad \text{(8 c.s.)}$$
$$\sqrt{1\,000\,000} = 1000{,}0000000 \quad \text{(exacto)}$$

La diferencia es 0,0005000, que tiene solo 4 cifras significativas correctas. Hemos perdido 4 cifras significativas en una resta.
La solución es algebraica: reescribir la expresión multiplicando y dividiendo por el conjugado.

$$f(x) = \frac{\left(\sqrt{x+1} - \sqrt{x}\right)\left(\sqrt{x+1} + \sqrt{x}\right)}{\sqrt{x+1} + \sqrt{x}} = \frac{1}{\sqrt{x+1} + \sqrt{x}}.$$

Con $x = 10^6$: $f(10^6) = 1 \,/\, (\sqrt{1\,000\,001} + 1000) \approx 1 \,/\, 2000{,}0005 \approx 4{,}9999988 \times 10^{-7}$, con todas las cifras significativas preservadas.

El ejemplo ilustra una regla general: la cancelación catastrófica no es un problema de la aritmética de coma flotante (la aritmética funciona correctamente), sino del *algoritmo* que elige restar cantidades casi iguales. El remedio es reformular el cálculo para evitar la resta.

Antes de calcular, examinemos más ejemplos:

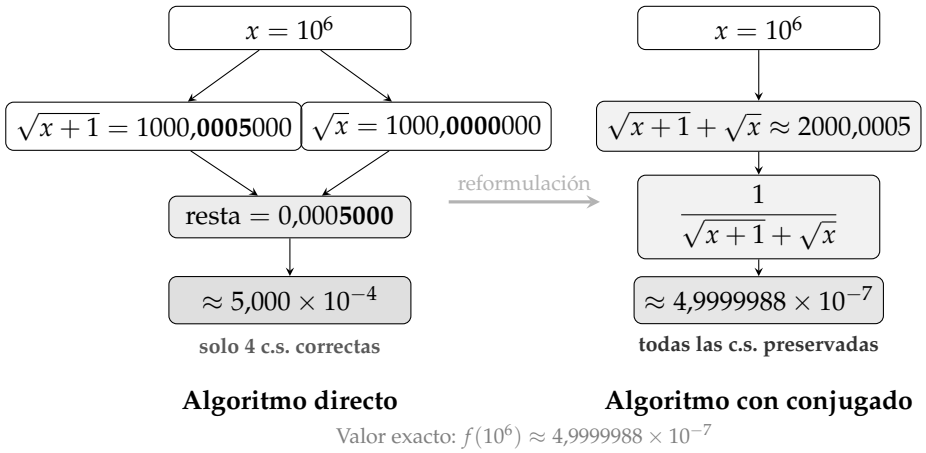



Figura 1.2: Cancelación catastrófica en el cálculo de $f(x) = \sqrt{x+1} - \sqrt{x}$ con $x = 10^6$ (aritmética de 8 dígitos decimales). El algoritmo directo (izquierda) resta dos números casi iguales y pierde 4 cifras significativas; el resultado $5{,}000 \times 10^{-4}$ tiene un error relativo de 1 %, aproximadamente. La reformulación por el conjugado (derecha) evita la resta y recupera toda la precisión (error relativo $\sim 10^{-8}$).

**EJEMPLO 1.2 (Evaluación de un polinomio).** Sea $p(x) = x^3 - 3x^2 + 3x - 1 = (x-1)^3$. Evaluar $p(1{,}00001)$ directamente requiere calcular $1{,}00001^3 - 3 \cdot 1{,}00001^2 + 3 \cdot 1{,}00001 - 1$, que implica restar números casi iguales.
La forma de Horner $p(x) = ((x-3)x+3)x - 1$, aunque algebraicamente equivalente, organiza el cálculo de forma que minimiza las cancelaciones. Pero en este caso la reformulación natural es simplemente evaluar $(x-1)^3$: $p(1{,}00001) = (0{,}00001)^3 = 10^{-15}$. Ningún algoritmo de evaluación directa alcanza esta precisión con aritmética de doble precisión.

La propagación de errores en operaciones aritméticas elementales puede analizarse mediante aritmética de primer orden. Si $\tilde{x} = x(1+\delta_x)$ con $|\delta_x| \leq \varepsilon$ y análogamente para $\tilde{y}$, entonces:

| Operación | Error relativo aproximado |
|---|---|
| $\tilde{x} \cdot \tilde{y}$ (producto) | $\approx \lvert\delta_x\rvert + \lvert\delta_y\rvert \leq 2\varepsilon$ |
| $\tilde{x}/\tilde{y}$ (cociente) | $\approx \lvert\delta_x\rvert + \lvert\delta_y\rvert \leq 2\varepsilon$ |
| $\tilde{x} + \tilde{y}$ (suma, mismo signo) | $\leq \max(\lvert\delta_x\rvert, \lvert\delta_y\rvert) \leq \varepsilon$ |
| $\tilde{x} - \tilde{y}$ ($\tilde{x} \approx \tilde{y}$) | puede ser $\gg \varepsilon$ (cancelación) |

Multiplicar y dividir son operaciones numéricamente benignas; la resta de números de magnitud similar es la operación problemática.

### 1.2.2. Propagación de errores en fórmulas: la regla de la cadena numérica

En la práctica, los ingenieros no solo suman o multiplican dos números: calculan fórmulas del tipo $R = \rho L/A$, $E = \frac{1}{2}mv^2$ o $\theta = \arctan(y/x)$. Cada variable de entrada $x_i$ lleva un error $\Delta x_i$, y queremos saber cuánto error introduce en el resultado final.

Si $f(x_1, x_2, \ldots, x_n)$ es diferenciable y los errores $\Delta x_i$ son pequeños, la propagación de errores en primer orden viene dada por la **regla de la cadena numérica**:

$$\Delta f \;\approx\; \sum_{i=1}^{n} \left| \frac{\partial f}{\partial x_i} \right| \Delta x_i.$$

El error relativo de $f$ se obtiene dividiendo entre $|f|$:

$$\frac{\Delta f}{|f|} \;\approx\; \sum_{i=1}^{n} \left| \frac{\partial f}{\partial x_i} \right| \frac{\Delta x_i}{|f|}.$$

Si todos los errores relativos $|\Delta x_i / x_i|$ son del mismo orden $\varepsilon$, es útil la forma **logarítmica**: tomando logaritmos y diferenciando,

$$\frac{\Delta f}{|f|} \;\approx\; \sum_{i=1}^{n} \left| \frac{\partial \ln f}{\partial \ln x_i} \right| \varepsilon_i,$$

donde $e_i = \partial \ln f / \partial \ln x_i = (x_i/f)\,\partial f/\partial x_i$ es el **exponente de error relativo** de $x_i$ en $f$. Para un monomio $f = x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}$ el exponente de $x_i$ es exactamente $a_i$, por lo que el error relativo de $f$ es $\sum_i |a_i|\,\varepsilon_i$.

La tabla siguiente recoge los casos más frecuentes:

| **Fórmula** $f$ | **Derivada parcial** | **Error absoluto** $\Delta f$ | **Error relativo** |
|---|---|---|---|
| $f = ax + by$ | $\frac{\partial f}{\partial x} = a$ | $\lvert a\rvert\Delta x + \lvert b\rvert\Delta y$ | $\frac{\lvert a\rvert\Delta x+\lvert b\rvert\Delta y}{\lvert ax+by\rvert}$ |
| $f = x^n$ | $\frac{\partial f}{\partial x} = nx^{n-1}$ | $\lvert n\rvert\, \lvert x\rvert^{n-1}\Delta x$ | $\lvert n\rvert \dfrac{\Delta x}{\lvert x\rvert}$ |
| $f = \sqrt{x^2+y^2}$ | $\frac{\partial f}{\partial x} = x/f$ | $\frac{\lvert x\rvert\Delta x+\lvert y\rvert\Delta y}{f}$ | $\dfrac{\lvert x\rvert\Delta x + \lvert y\rvert\Delta y}{x^2+y^2}$ |
| $f = \ln x$ | $\frac{\partial f}{\partial x} = 1/x$ | $\Delta x/\lvert x\rvert$ | $\Delta x \,/\, (\lvert x\rvert\, \lvert \ln x\rvert)$ |
| $f = e^x$ | $\frac{\partial f}{\partial x} = e^x$ | $e^x\Delta x$ | $\lvert x\rvert \dfrac{\Delta x}{\lvert x\rvert} = \lvert x\rvert\, \varepsilon_x$ |
| $f = \arctan(\frac{y}{x})$ | $\frac{\partial f}{\partial y} = \frac{x}{x^2+y^2}$ | $\frac{\lvert x\rvert\Delta y+\lvert y\rvert\Delta x}{x^2+y^2}$ | $\frac{\lvert x\rvert\Delta y+\lvert y\rvert\Delta x}{(x^2+y^2)\,\lvert\arctan(y/x)\rvert}$ |

La regla de la cadena numérica es una *cota superior* (caso de peor signo coordinado). Si los errores son aleatorios e independientes, la estimación estadística más realista usa la norma cuadrática: $\Delta f \approx \sqrt{\sum_i (\partial f/\partial x_i)^2\,(\Delta x_i)^2}$. La suma de valores absolutos corresponde al caso en que todos los errores conspiran en el peor sentido posible.

**EJEMPLO 1.3 (Resistencia equivalente en paralelo).** Dos resistencias $R_1 = 100\,\Omega$ y $R_2 = 200\,\Omega$, medidas con error relativo $\varepsilon = 1\,\%$, se conectan en paralelo. La resistencia equivalente es

$$R = \frac{R_1 R_2}{R_1 + R_2}.$$

Queremos estimar el error relativo de $R$. Tomando logaritmos:

$$\ln R = \ln R_1 + \ln R_2 - \ln(R_1 + R_2).$$

Diferenciando con respecto a $R_1$ (y análogamente para $R_2$):

$$\frac{\partial \ln R}{\partial \ln R_1} = 1 - \frac{R_1}{R_1+R_2} = \frac{R_2}{R_1+R_2}, \qquad \frac{\partial \ln R}{\partial \ln R_2} = \frac{R_1}{R_1+R_2}.$$

Con los valores numéricos: $R_2/(R_1+R_2) = \frac{200}{300} \approx 0{,}667$ y $R_1/(R_1+R_2) = \frac{100}{300} \approx 0{,}333$. El error relativo de $R$ es:

$$\frac{\Delta R}{R} \approx 0{,}667\,\varepsilon_1 + 0{,}333\,\varepsilon_2 = (0{,}667 + 0{,}333) \times 1\,\% = 1\,\%.$$

El error relativo de la resistencia en paralelo no supera el mayor de los errores individuales. Esto se debe a que los exponentes de error suman 1 (la fórmula es homogénea de grado 1 en las resistencias).

**EJEMPLO 1.4 (Energía cinética).** La energía cinética $E = \frac{1}{2}mv^2$ depende de la masa $m$ (error relativo $\varepsilon_m$) y de la velocidad $v$ (error relativo $\varepsilon_v$). Aplicando la regla logarítmica:

$$\frac{\Delta E}{E} \approx \left|\frac{\partial \ln E}{\partial \ln m}\right| \varepsilon_m + \left|\frac{\partial \ln E}{\partial \ln v}\right| \varepsilon_v = 1 \cdot \varepsilon_m + 2 \cdot \varepsilon_v.$$

El exponente de $v$ es 2, así que un error relativo del 1 % en la velocidad produce un 2 % de error en la energía. Los exponentes de la fórmula son los amplificadores de error: en $f = x^a y^b$, los errores se amplifican por $|a|$ y $|b|$.

**Ejercicio resuelto 1.1 (Propagación en el ángulo de un vector).** Un vector tiene componentes $x = 3{,}0 \pm 0{,}1$ e $y = 4{,}0 \pm 0{,}1$ (errores absolutos). Calcular el error absoluto del ángulo $\theta = \arctan(y/x)$.

**Solución.** Las derivadas parciales de $\arctan(y/x)$ son:

$$\frac{\partial \theta}{\partial x} = \frac{-y}{x^2 + y^2}, \qquad \frac{\partial \theta}{\partial y} = \frac{x}{x^2 + y^2}.$$

Con $x^2 + y^2 = 9 + 16 = 25$:

$$\Delta\theta \approx \frac{|-y|\,\Delta x + |x|\,\Delta y}{x^2 + y^2} = \frac{4{,}0 \times 0{,}1 + 3{,}0 \times 0{,}1}{25} = \frac{0{,}7}{25} = 0{,}028\,\text{rad} \approx 1{,}6^\circ.$$

El ángulo exacto es $\theta = \arctan(\frac{4}{3}) \approx 53{,}13^\circ$, así que el error relativo es $1{,}6/53{,}13 \approx 3\,\%$.