Índice general

CAPÍTULO 1

Estabilidad y errores en el cálculo numérico

1.1 Introducción

El saber contar y realizar operaciones aritméticas es el resultado de una larga evolución en el pensamiento humano. En todo este proceso, el hombre ha tratado de ayudarse en la realización de los cálculos mediante artilugios que simplificaran su labor. Primero fueron simples guijarros (calculi en latín), después fueron instrumentos mecánicos simples como el ábaco, instrumentos mecánicos articulados como las máquinas de Pascal o de Schickard y finalmente los computadores electrónicos. También, desde la programación en cilindros rotatorios o tarjetas perforadas hasta los desarrollos actuales en *software* basados en los lenguajes modernos de programación, se ha recorrido un largo camino. Todo este conocimiento y tecnología permite actualmente realizar cálculos de gran complejidad.

Sin embargo, un computador produce resultados en respuesta a cálculos programados que posiblemente difieren ligeramente de los valores exactos esperados. Ello es consecuencia de que trabajan con una aritmética discreta que no coincide plenamente con la aritmética exacta de los números enteros o reales. Todavía el ser humano no alcanza a realizar por medios físicos, todos los cálculos que su mente puede concebir, ni siguiera a representar en la memoria de un ordenador más que un subconjunto finito del conjunto de todos los números que puede manejar.

El propósito de este capítulo es estudiar las representaciones más comunes de números en un computador y las aritméticas que usan en sus cálculos. Una vez establecido que los errores respecto a la aritmética real pueden es-

tar presentes, se introduce el concepto de estabilidad numérica que permite seleccionar los algoritmos que son válidos para esta clase de cálculos.

1.2 Representación de números en un computador

La diferencia (en valor absoluto) entre el valor exacto x y el valor obtenido en un cálculo $\bar{x}$, determina el error absoluto cometido $E(x) = |x - \bar{x}|$. Una indicación más precisa del error cometido en el cálculo la da el error relativo, que se define por la siguiente expresión

$$E_R(x) = \frac{|x - \bar{x}|}{|x|}.$$

Si un determinado cálculo produce como resultado $\bar{x} = 1,2345 \times 10^{12}$ siendo el valor exacto esperado $x = 1.2331 \times 10^{12}$, el error absoluto sería $E(x) = 1.4 \times 10^9$. Este error parece muy alto si no se considera el orden de magnitud de los números con los que se realiza el cálculo. Por el contrario, el error relativo

$$E_R(x) = \frac{1.4 \times 10^9}{1.2331 \times 10^{12}} = 1.1353 \times 10^{-3}$$

da una idea más proporcionada de la imprecisión cometida.

Los computadores representan los números en sus memorias asignándoles una cantidad fija de posiciones a las que puede acceder. Esta cantidad puede variar de un computador a otro, de acuerdo con el estándar de representación que se haya atribuido a ese tipo de número. Es obvio que el modo de representación interna de un número en una memoria no tiene que coincidir necesariamente con el modo en que este número se muestra en una pantalla u otro dispositivo terminal.

Con el fin de comprender cómo usan los ordenadores la aritmética discreta, se recuerda el concepto de representación posicional de los números enteros y reales. Se considera el conjunto de números $\mathcal{M}$ que pueden ser representados en la forma

$$\pm 0.d_1 d_2 \cdots d_n \times b^e$$

donde $0 \leq d_i < b$ para $i = 1, 2, \cdots, n$. El número entero positivo b es fijo y corresponde a la base de la representación. El entero e corresponde al exponente y puede variar en un determinado rango. Finalmente, el entero positivo fijo n controla la precisión de la representación. La parte fraccionaria es frecuentemente llamada mantisa. El valor numérico real asignado a esta representación es

$$x = \pm(d_1 b^{-1} + d_2 b^{-2} + \cdots + d_n b^{-n}) \times b^e.$$

Por ejemplo, a la representación 0.1011×2^3 en base 2 le corresponde el valor decimal

$$x = 4 + 1 + \frac{1}{2} = 5.5$$

Idealmente, se puede ampliar el sistema admitiendo que n sea infinito y de este modo, se podrían incluir todos los números reales en esta representación. Obviamente, para un sistema de representación posicional que se pretenda poner en práctica en un computador, se requiere que n sea un número entero fijo y que se ajuste a la capacidad física de su memoria accesible. Por esta razón, todo sistema de representación que utilice un computador, tendrá un conjunto finito de números-máquina.

Habitualmente, para números enteros se emplea una representación en la que el exponente e es fijo e igual a n. De este modo, a la representación $\pm d_1 d_2 \cdots d_n$ le corresponde el valor entero

$$x = \pm(d_1 b^{n-1} + d_2 b^{n-2} + \cdots + d_n).$$

▪ Ejemplo **1** En un sistema binario ($b = 2$), con un número fijo de dígitos $n = 3$, el conjunto de números (que no son negativos) que pueden ser representados de este modo, son

Representación	000	001	010	011	100	101	110	111
Valor real	0	1	2	3	4	5	6	7

Un aspecto relevante de este sistema de representación es que puede producirse un desbordamiento en las operaciones aritméticas debido a que el conjunto es acotado. Por ejemplo, la suma de 3 y 5 produce un número que no puede ser representado en este sistema. ◇

Si el exponente e es un entero variable en un determinado rango, el conjunto de números que pueden ser representados de este modo (llamado de punto flotante) es más amplio, como muestra el siguiente ejemplo

▪ Ejemplo **2** En un sistema binario $b = 2$, si el número de dígitos es $n = 2$, el conjunto de números (que no son negativos) que pueden ser representados en punto flotante con el rango $-2 \leq e \leq 2$, es el siguiente

Representación	$.00 \times 2^e$	$.01 \times 2^e$	$.10 \times 2^e$	$.11 \times 2^e$
Valor	0	$\frac{1}{4} \times 2^e$	$\frac{1}{2} \times 2^e$	$\frac{3}{4} \times 2^e$

En definitiva, el conjunto de números que no son negativos y pueden ser representados de este modo son

$$\{0, \frac{1}{16}, \frac{1}{8}, \frac{3}{16}, \frac{1}{4}, \frac{3}{8}, \frac{1}{2}, \frac{3}{4}, 1, \frac{3}{2}, 2, 3\}.$$

Los aspectos más relevantes de este sistema de representación son los siguientes:

- Los números representados no son equidistantes y se observa una mayor densidad en las proximidades de 0.
- Un número puede tener representaciones distintas. Por ejemplo, el número $\frac{1}{8}$ se puede representar por $.01 \times 2^{-1}$ o por $.10 \times 2^{-2}$.
- Se puede producir un desbordamiento en las operaciones aritméticas debido a que el conjunto es acotado. Por ejemplo, la suma de 1 y 3 produce un número que no puede ser representado en este sistema.
- El resultado de algunas operaciones aritméticas reales con números de este conjunto está fuera del conjunto aunque no se haya producido un desbordamiento del rango. Por ejemplo, la suma real de 2 y $\frac{1}{8}$ no pertenece al conjunto. $\diamond$

Una representación en punto flotante está normalizada si el primer dígito en la parte fraccionaria es necesariamente distinto de 0. De este modo, se evita que un número pueda tener representaciones distintas en un mismo sistema. Actualmente, la representación que usan la mayoría de computadores es la llamada IEEE Standard 754 en punto flotante. Está basada en los tres elementos mencionados anteriormente, el signo, la mantisa y el exponente.

Si la representación es binaria, el primer dígito es 1 (dígito principal) y en la mayoría de las puestas en práctica de este estándar, no es almacenado en memoria (dígito principal implícito). El estándar de representación en punto flotante de IEEE que corresponde a lo que se conoce como precisión simple, utiliza 4 bytes de memoria (32 bits) de los cuales 8 bits son para el exponente E, 23 para la parte fraccionaria F y uno para el signo S. Cada bit almacena un 0 o un 1. El valor asignado a una representación es

$$(-1)^S \times 2^{E-127} \times 1.F$$

El campo del exponente necesita representar a la vez exponentes positivos y negativos. Para conseguirlo se añade un sesgo al valor positivo almacenado para lograr el exponente deseado. En el estándar IEEE 754 para precisión

simple, este valor es 127. De este modo, un valor almacenado E representa un valor real $E - 127$. Para doble precisión el campo del exponente tiene 11 bits y un sesgo de 1023. El bit del signo es 0 para positivos y 1 para negativos.

Por ejemplo, si en memoria está la siguiente información almacenada de acuerdo al estándar IEEE 754 con dígito principal implícito

$$\underbrace{1}_{-} \quad \underbrace{01010011}_{\text{exponente}} \quad \underbrace{10011110001010000101000}_{\text{mantisa}}$$

el número representado es

$$-\left(1+\frac{1}{2}+\frac{1}{2^4}+\frac{1}{2^5}+\frac{1}{2^6}+\frac{1}{2^7}+\frac{1}{2^{11}}+\frac{1}{2^{13}}+\frac{1}{2^{18}}+\frac{1}{2^{20}}\right)\times 2^{83-127}$$

ya que $01010011_2 = 83$.

1.3 Aritmética en un sistema de representación finito

En general, cuando se utiliza un sistema discreto $\mathcal{M}$, inevitablemente aparecen errores al introducir los números reales convirtiéndolos a números-máquina pero también como el resultado de una operación entre números-máquinas que en general no coincide con el resultado que se obtendría en la aritmética real. Conceptualmente, los números reales pueden aproximarse por números del sistema discreto de dos modos ligeramente distintos que se conocen como truncamiento y redondeo.

La puesta en práctica de métodos de truncamiento está basada en la siguiente idea: Un numero real puede ser aproximado por un número de punto flotante con parte fraccionaria infinita en la base b

$$\pm .d_1 d_2 \cdots \times b^e.$$

Consecuentemente, si se trunca esta serie con n dígitos se obtiene una aproximación en el sistema discreto disponible en el computador. Sin embargo, es más usado el llamado método de redondeo que consiste en aproximar un número real positivo por el número-máquina más próximo. Este procedimiento sería el que nos proporcionaría mayor precisión. Para precisar estas ideas se representa la función parte entera de un número real por un corchete [ ] y se define para un número real positivo representado por $x = 0.m \times b^e$ donde m es una cifra con posiblemente una infinidad de dígitos, las siguientes aproximaciones por números máquina de n dígitos

Truncamiento	Redondeo
$\mathcal{T}(x) = [b^n \times 0.m] \times b^{e-n}$	$\mathcal{R}(x) = [b^n \times 0.m + 0.5] \times b^{e-n}$
Ejemplo (3 dígitos, base 10):	Ejemplo(3 dígitos, base 10):
$\mathcal{T}(1/3) = 10^{-3}[10^3 \times 0.333...] = 0.333$	$\mathcal{R}(1/3) = 10^{-3}[10^3 \times 0.333... + 0.5] = 0.333$
$\mathcal{T}(2/3) = 10^{-3}[10^3 \times 0.666...] = 0.666$	$\mathcal{R}(2/3) = 10^{-3}[10^3 \times 0.666... + 0.5] = 0.667$

En lo que sigue se centrará la atención en el método de redondeo. Si $x < 0$ entonces se define

$$\mathcal{R}(x) = -\mathcal{R}(-x).$$

De la definición de redondeo se deduce que

$$|x - \mathcal{R}(x)| \leq \frac{1}{2} b^{e-n}.$$

El número $\frac{1}{2}b^{e-n}$ se conoce como unidad de redondeo o precisión de la máquina.

Una aritmética para un sistema de representación finita estaría disponible si se consigue asignar como resultado de una operación el que corresponde al redondeo de la operación aritmética exacta.

– Ejercicio 1 *Sea $x > 0$ un número que se representa exacto en una aritmética finita de punto flotante. Analizar el comportamiento del cociente*

$$\frac{\operatorname{sen}(x+h) - \operatorname{sen} x}{h}$$

cuando se consideran valores de h próximos a 0 y los cálculos se realizan en una aritmética de punto flotante.

Solución: Para h suficientemente pequeño el resultado de redondear $x + h$ coincide con x. Por ello, el numerador es nulo y el valor asignado al cociente es 0. ◇

– Ejercicio 2 *Si se usa una aritmética con redondeo, en un sistema de representación decimal y de 3 dígitos de precisión, para realizar la siguiente la operación*

$$\frac{a \times b - c}{b + 2 \times c}, \quad a = 1.34,\ b = 0.712,\ c = -0.355,$$

determinar el error cometido en relación con la aritmética real.

Solución: En la representación finita los números dados son

$$a = 0.134 \times 10^1,\ b = 0,712 \times 10^0,\ c = -0.355 \times 10^0.$$

Consecuentemente, el resultado de las operaciones es el siguiente:

- $a \times b = \mathcal{R}(0.95408) = 0.954 \times 10^0$.
- $a \times b - c = \mathcal{R}(0.954 + 0.355) = 0.131 \times 10^1$.
- $2 \times c = -0.710$.
- $b + 2 \times c = 0.2 \times 10^{-2}$.
- $\frac{a \times b + c}{b + 2 \times c} = 0.655 \times 10^3$.

El resultado exacto es 0.65454×10^3 y consecuentemente el error relativo es

$$E_R = \frac{|655 - 654.54|}{654.54} = 0.00070278\ . \qquad \diamond$$

1.4 Estabilidad en los algoritmos numéricos

Una vez que un entorno de cálculo dispone de una aritmética finita que le permite realizar las operaciones elementales dentro del rango numérico que puede representar, se hace necesario complementarlo con otras funciones elementales que permitan realizar cálculos más complejos. La dificultad está en que estas funciones implican un número elevado de operaciones elementales y puesto que los errores de redondeo respecto a la aritmética exacta son inevitables, el resultado final puede estar muy deteriorado en relación con el exacto.

▪ Ejemplo **3** Si en un entorno que usa el estándar IEEE Double Precision, se realiza el cálculo de e^{-10} mediante el truncamiento de la serie de potencias

$$e^x \approx 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \cdots \frac{x^{30}}{30!}$$

el resultado aproximado es $9.703415796025504 \times 10^{-4}$. Desafortunadamente, este valor se aleja del valor exacto $4.539992976248485 \times 10^{-5}$. El error de truncamiento de la serie está dado

$$E(-10) = \frac{e^\xi}{31!} < \frac{1}{31!}$$

para algún valor $-10 < \xi < 0$. Consecuentemente, si el cálculo se realizase en una aritmética exacta, el truncamiento de la serie no podría producir un error de ese tamaño. El motivo de la inexactitud está en que los pequeños errores causados por la aritmética finita, han sido magnificados a lo largo del cálculo.

Por otra parte, si se calcula $e^{-10} = \frac{1}{e^{10}}$ usando sólo el desarrollo en serie del denominador, el resultado es $4.539993338712231 \times 10^{-5}$ que es muy próximo al valor exacto. $\diamond$

La razón por la que un procedimiento produce resultados más imprecisos que otro se atribuye a su inestabilidad. Es decir, un error en alguna etapa del procedimiento se propaga de un modo creciente en las siguientes. La necesidad de utilizar algoritmos estables para evaluar una función es obligada por la inevitable presencia de errores de redondeo debidos al uso de aritméticas finitas en los entornos de cálculo automático.

1.5 Ejercicios

▪ Ejemplo **4** En este ejemplo se muestra cómo convertir el número decimal $x = 324.65$ a base 2. Se separa la parte entera 324 de la parte decimal 0.65. Para la parte entera se procede de modo sigue:

$$\begin{aligned} 324 &= 162 \times 2 + 0 \\ 162 &= 81 \times 2 + 0 \\ 81 &= 40 \times 2 + 1 \\ 40 &= 20 \times 2 + 0 \\ 20 &= 10 \times 2 + 0 \\ 10 &= 5 \times 2 + 0 \\ 5 &= 2 \times 2 + 1 \\ 2 &= 1 \times 2 + 0 \end{aligned}$$

En consecuencia se tiene que $324 = 101000100_2$. El dígito más significativo siempre es 1 y los siguientes son los restos de la divisiones por 2 comenzando desde la última división.

En la parte decimal, se procede como sigue

$$\begin{array}{rcl} 0.65 & \times & 2 = 1.3 \\ 0.3 & \times & 2 = 0.6 \\ 0.6 & \times & 2 = 1.2 \\ 0.2 & \times & 2 = 0.4 \\ 0.4 & \times & 2 = 0.8 \\ 0.8 & \times & 2 = 1.6 \\ 0.6 & \times & 2 = 1.2 \\ \dots & \dots & \dots \end{array}$$

En consecuencia se tiene que $0.65 = 0.1010011\cdots_2$. Los dígitos son las partes enteras de las sucesivas multiplicaciones por 2 comenzando por la primera. ◇

EJERCICIO 3 *Hallar la representación en punto flotante (sin dígito principal implícito) de*

1. $2/7$ *en un sistema* $b = 10$, $n = 6$.
2. 327 *en un sistema* $b = 2$, $n = 6$.
3. $-\frac{4}{3}$ *en un sistema* $b = 2$, $n = 6$.

por redondeo.

Solución:

1. $\mathcal{R}(\frac{2}{7}) = 0.285714 \times 10^0$.
2. Puesto que $327 = 2^8 + 2^6 + 2^2 + 2 + 1 = 0.101000111_2 \times 2^9$ entonces se tiene que

$$\mathcal{R}(327) = [2^6 \times 0.101000111_2 + 0.5] \times 2^3 = 0.101001_2 \times 2^9 = 328.$$

3. Puesto que $\frac{4}{3} = 0.10101010..._2 \times 2$ entonces se tiene que

$$\mathcal{R}(\frac{4}{3}) = [2^6 \times 0.1010101010_2 + 0.5] \times 2^{-5} = 0.101011_2 \times 2.$$

Consecuentemente $-\frac{4}{3} = -0.101011_2 \times 2$ ◇.

EJERCICIO 4 *Hallar el resultado con la aritmética finita de un sistema de representación $b = 2$, $n = 6$ y $-3 \leq e \leq 3$, sin dígito principal implícito, de las siguientes operaciones*

1. $\frac{1}{3} + \frac{7}{13}$
2. $47 + 347$

Solución:

1. Puesto que $\mathcal{R}(1/3) = 0.101011_2 \times 2^{-1}$ y $\mathcal{R}(7/13) = 0.100010_2$ entonces

$$\begin{aligned} \mathcal{R}(\mathcal{R}(1/3) + \mathcal{R}(7/13)) &= \mathcal{R}(0.1101111_2) = 2^{-6}[110111.1_2 + 0.1_2] \\ &= 2^{-6} \times 111000_2 = 0.111000_2 \ . \end{aligned}$$

2. El valor real máximo que se puede representar en este sistema es

$$0.111111_2 \times 2^3 = 4 + 2 + 1 + 0.5 + 0.25 + 0.125 = 7.875.$$

Consecuentemente, las representaciones de 47 y 347 producen un desbordamiento (overflow) que habitualmente es notificado . ◇

EJERCICIO 5 *Se pretende aproximar el valor de $e^{0.1}$ usando la serie de potencias*

$$e^x = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} + \cdots$$

en una aritmética basada en el redondeo en base 10 y 7 dígitos de precisión. Si se usan cinco sumandos de la serie de potencias, calcular el error relativo que se comete al realizar la aproximación en la aritmética finita respecto de la que corresponde a la aritmética exacta.

Solución: Los cálculos exactos de los cinco primeros términos de la serie dan las siguientes resultados

n	1	2	3	4	5	$p_5(0.1)$
$\frac{x^n}{n!}$	1	0.1	0.005	$0,0001\hat{6}$	$0.0000041\hat{6}$	$1.1051708\hat{3}$
$\mathcal{R}\left(\frac{x^n}{n!}\right)$	1	0.1	0.005	0.0001667	0.0000042	1.105171

donde p_5 representan los valores de la serie truncada, calculados en aritmética exacta y finita. El sombrero representa el decimal periódico en la expresión decimal.

$$\mathcal{R}(x) = [10^7 \times 0.110517083\hat{} + 0.5] \times b^{-6} = 1.105171$$

En este caso, el resultado no depende del orden en el que se realicen las sumas. El error relativo es

$$E_R(0.1) = \frac{1.105171 - 1.10517083\hat{}}{1.10517083\hat{}}. \qquad \diamond$$