Cálculo Numérico I

Introducción. Números y operaciones numéricas

La mayor parte de las funciones que aparecen en matemáticas son difícilmente evaluables. No solamente aquellas que aparecen como soluciones de algunos problemas y no pueden expresarse algebraicamente en términos de funciones elementales, sino que estas mismas funciones elementales —incluyendo entre ellas las funciones trigonométricas, la exponencial, y sus inversas— son difíciles de evaluar. Recordemos que sus valores se obtienen comúnmente de tablas ya calculadas o por medio de máquinas calculadoras. Para calcular estas tablas, para programar estas calculadoras, para estimar valores numéricos relacionados con aquellas funciones, necesitamos técnicas que nos permitan, a partir de la definición de la función, aproximarnos al valor real que queremos estimar controlando en todo momento la magnitud del error que estamos cometiendo. Estas técnicas, reglas, métodos, junto con su justificación teórica, y por tanto control de su error, es lo que constituye el cálculo numérico.

Desde la aparición de las calculadoras electrónicas —incluyendo entre ellas los ordenadores, grandes y pequeños— el panorama del cálculo numérico ha cambiado enormemente. Ya no se trata de buscar métodos y reglas que en unas pocas operaciones nos faciliten el valor buscado; ya no tiene tanta importancia el número de operaciones necesario para realizar un cálculo, es más importante la estabilidad de ese cálculo a pesar del gran número de operaciones, el control del error que produce la aritmética del ordenador.

No significa esto que no haya problemas en los cuales sea necesario

buscar métodos que limiten el número de operaciones. Son por lo general problemas que antes de la aparición del ordenador eran totalmente inabordables debido a su magnitud pero que con la nueva tecnología son fácilmente programables. Estamos pensando en particular en los problemas de inversión de matrices y resolución de sistemas lineales de gran tamaño, aunque no sea este el único caso.

1 Sistemas de numeración

Las reglas aritméticas que hemos aprendido en la escuela están basadas en el sistema numérico de base *diez* (sistema decimal). Hemos aprendido de memoria unas tablas que contienen la suma y el producto de los diez primeros números naturales (que utilizamos como *dígitos*) y a partir de ellas sabemos como operar con números de cualquier magnitud siempre que estén expresados en términos de estos diez dígitos. La expresión de cualquier número natural N en términos de estos diez dígitos viene dada por *la concatenación de los restos que resultan de dividir por diez el número N y sus sucesivos cocientes*. Es decir, la primera cifra —que situamos a la derecha— es el resto R_1 de dividir N entre *diez*, con cociente N_1; la segunda cifra —que situamos inmediatamente a la izquierda de aquella— es el resto R_2 de dividir N_1 entre *diez*, con cociente N_2; iteramos este proceso hasta que obtenemos un último resto R_k —que situamos inmediatamente a la izquierda de la concatenación de restos obtenidos hasta el momento— al obtener por cociente cero. Así la expresión decimal del número N es la concatenación de restos $R_k R_{k-1} \dots R_2 R_1$ todos ellos números entre cero y nueve que representamos usualmente por las cifras $0, 1, \dots, 9$.

Esta es la representación numérica que todos conocemos desde nuestra más tierna infancia descrita en términos un tanto sofisticados con el fin de observar la arbitrariedad contenida en la elección del número *diez* en este proceso —arbitrariedad seguramente debida al hecho de ser éste el número total de dedos, lo que nos permite utilizar las manos como un ábaco natural. Una vez observada esta arbitrariedad nada nos impide cambiar los términos y jugar a reconstruir nuestra aritmética elemental en una base distinta. Puede ser ésta cualquier número positivo b mayor que *uno*. Necesitaremos para esta aritmética un número b de cifras que representen los b primeros números naturales incluido el *cero*. El proceso

será análogo al descrito más arriba para $b =$*diez* y tendremos que construir las correspondientes tablas de operaciones para estos b primeros números.

Estudiemos por un momento qué sucede en base *ocho*. Usaremos los ocho primeros dígitos con su significado habitual: 0, 1, 2, 3, 4, 5, 6, y 7. El número natural siguiente, *ocho*, se representará como 10, que significa, expresado en base *diez*, $1 \cdot 8^1 + 0 \cdot 8^0$. Los números siguientes serán 11, 12, etc., hasta llegar al número 17, cuyo siguiente es el 20.

Ejemplo. Dado el número $341_{(10}$ ¿cuál será su expresión en base *ocho*? Calculamos —operando en base *diez*, que es en la que estamos entrenados a operar— los restos de dividir este número y los sucesivos cocientes por *ocho*: $341_{(10} = 42 \cdot 8 + 5$; $42 = 5 \cdot 8 + 2$; $5 = 0 \cdot 8 + 5$; por tanto $341 = 5 \cdot 8^2 + 2 \cdot 8^1 + 5 \cdot 8^0 = 525_{(8}$.

Ejemplo. Dado el número $1202_{(8}$ hallar su expresión en base *diez*. Simplemente tenemos que escribir en base diez el significado de su expresión en base *ocho* y operar: $1202_{(8} = 2 + 0 \cdot 8 + 2 \cdot 8^2 + 1 \cdot 8^3 = 642_{(10}$.

La base más sencilla de todas —aparte de la base *uno* que es la que utiliza la representación por medio de palotes— es la base *dos* que solamente utiliza dos cifras: 0, 1; y a la que se puede dar por tanto cualquier significado dicotómico: si–no; abierto–cerrado; encendido–apagado. Esto hace que la base *dos* sea la adecuada para las operaciones internas de cualquier operador electrónico o incluso mecánico.

Todo número se representa en base *dos* simplemente señalando con un 1 qué potencias de *dos* aparecen en su representación (y con un 0 las que no aparecen): $1001011101_{(2}$ significa, expresado en base *diez* $2^0 + 2^2 + 2^3 + 2^4 + 2^6 + 2^9$.

Existe un juego bastante conocido que consiste en adivinar un número a partir de seleccionar de entre una serie de tarjetas con listas de números, aquellas en las que el número en cuestión se encuentra. La solución se halla sumando los primeros números que aparecen en estas tarjetas seleccionadas, que son las potencias en base *dos* contenidas en el número dado, es decir *los unos del desarrollo en base dos*.

Algunas botellas de bebidas señalan el mes y el día de llenado de

la botella por medio de marcas en la serie de números 1, 2, 4, 8, 16, y 1, 2, 4, 8. Es particularmente interesante la forma de indicar el día del mes ya que con estos números se cubren exactamente los 31 días posibles.

Ejercicio. Escribir en base *dos* los números en base *diez* siguientes: 521, 473, 1024. Escribir en base *diez* los números $1000101_{(2}$, $121212_{(3}$, $54321_{(6}$.

Otras bases especialmente importantes aparte de *dos* y *diez* son las bases *ocho* y *dieciseis*, que por ser potencias de *dos* tienen una traducción especialmente fácil de unas a otras, siendo las expresiones de cualquier numero más cortas cuanto mayor es la base que se utiliza. Para convertir un número de base *dos* a base *ocho* o *dieciseis* o viceversa basta tener una tabla de los dígitos de éstas últimas en base *dos*:

Base 8		Base 16			
000	0	0000	0	1000	8
001	1	0001	1	1001	9
010	2	0010	2	1010	A
011	3	0011	3	1011	B
100	4	0100	4	1100	C
101	5	0101	5	1101	D
110	6	0110	6	1110	E
111	7	0111	7	1111	F

Cualquier número en base *dos* —100010001110101101— se transforma a base *ocho* o a base *dieciseis* separando desde la derecha bloques de tres o cuatro dígitos respectivamente, y sustituyéndolos por el dígito en la base correspondiente:

$$100\,010\,001\,110\,101\,101_{(2} = 421655_{(8}$$

$$10\,0010\,0011\,1010\,1101_{(2} = 223\mathrm{AD}_{(16}$$

Los números en bases *dos, ocho, diez,* y *dieciseis* se denominan *binarios, octales, decimales,* y *hexagesimales* respectivamente. Muchas calculadoras digitales están preparadas para operar en cualquiera de estas bases.

Hasta ahora hemos hablado de la representación de números enteros. Para representar números fraccionarios podemos utilizar en

cualquier base métodos análogos a los utilizados en base *diez*. ¿Qué significa la expresión decimal $31'27$? Expresado en fracciones quiere decir $31 + \frac{2}{10} + \frac{7}{10^2}$. Esta idea para representar la parte fraccionaria de un número puede utilizarse en cualquier base b: la parte fraccionaria de un número se expresa como suma de fracciones cuyos numeradores son enteros entre 0 y $b-1$ y cuyos denominadores son las potencias de b.

Ejemplo. $0'13265_{(8} = 1{\cdot}8^{-1} + 3{\cdot}8^{-2} + 2{\cdot}8^{-3} + 6{\cdot}8^{-4} + 5{\cdot}8^{-6}$.

Sabido esto, ¿como representaremos un número fraccionario x en una base dada b? Supongamos que $a_1, a_2, a_3, \ldots$ son las cifras del desarrollo: $x = 0'a_1a_2a_3\cdots = a_1{\cdot}b^{-1} + a_2{\cdot}b^{-2} + a_3{\cdot}b^{-3} + \ldots$. Observemos que $b{\cdot}x = a_1 + a_2{\cdot}b^{-1} + a_3{\cdot}b^{-3} + \ldots$ y, por tanto, a_1 es la parte entera de $b{\cdot}x$; $b(b{\cdot}x - a_1) = a_2 + a_3{\cdot}b^{-1} + \ldots$, de forma que a_2 es la parte entera de $b(b{\cdot}x - a_1)$, etc.

Ejemplo. Expresar en base 3 el número $0'31_{(10}$.

0'31	x	3	=	0'93	su parte entera es	0
0'93	x	3	=	2'79	—"—	2
0'79	x	3	=	2'37	—"—	2
0'37	x	3	=	1'11	—"—	1
0'11	x	3	=	0'33	—"—	0
0'33	x	3	=	0'99	—"—	0
etc.						

Obtenemos $0'31_{(10} = 0'022100\ldots_{(3}$. ¿Hasta cuándo debemos continuar el cálculo? Dado que el número cuyo desarrollo fraccionario estamos calculando es racional, las cifras deben repetirse de forma periódica a partir de algún lugar. Esto ocurrirá en cuanto la parte fraccionaria en base 10, que siempre tiene el mismo número de cifras —en nuestro ejemplo 2—, se repita por primera vez; a partir de este punto todos los cálculos se repiten.

2 Representación de números en el ordenador

Al calcular numéricamente con una máquina debemos considerar principalmente dos tipos de números: *enteros* y de *coma flotante* (en inglés,*floating point*). Por medio de los segundos tratamos de

reproducir en el ordenador los números reales. Sin embargo solamente utilizamos un cantidad finita de números racionales con una aritmética particular. La idea es la siguiente.

Todo número real x, expresado en base *diez*, se escribe de la forma

$$x = \sigma \cdot \left(\sum_{k=1}^{\infty} x_k 10^{-k} \right) \cdot 10^N$$

donde σ es el signo, N es un exponente entero y los x_k son los dígitos $0, 1, \ldots, 9$ de su desarrollo decimal, con la única restricción $x_1 \neq 0$.

Por ejemplo, $\pi = (+1) \cdot (3 \cdot 10^{-1} + 1 \cdot 10^{-2} + 4 \cdot 10^{-3} + \cdots) \cdot 10^1$.

La representación de coma flotante consiste en truncar (o redondear) la expresión anterior limitando la suma hasta un número fijo de sumandos t.

$$\texttt{FLO}(x) = \sigma \cdot \left(\sum_{k=1}^{t} x_k \cdot 10^{-k} \cdot 10^N \right) = \sigma \cdot \bar{x} \cdot 10^N$$

FLO

El número $\bar{x}$ recibe el nombre de *mantisa* de $\texttt{FLO}(x)$. En la representación de coma flotante tenemos dos restricciones: la longitud t de la mantisa y el tamaño del exponente N: $-N_0 \leq N \leq N_1$, con N_0 y N_1 enteros positivos (que suelen ser iguales o diferir en una unidad).

MANTISA

Es importante conocer como funciona la aritmética con coma flotante. Curiosamente, no se verifica una propiedad elemental de la aritmética de números reales: la propiedad asociativa, según la cual, por ejemplo, $x(yz) = (xy)z$ lo que nos permite escribir sin ambigüedad xyz. La razón es que el resultado de una multiplicación de números con coma flotante es otro número con coma flotante que se obtiene por medio de un redondeo o de una truncación. Veamos un ejemplo, supongamos que utilizamos representación en coma flotante con una mantisa de longitud 2 y queremos hallar el producto de los tres números 0'24; 0'31; y 0'93. El producto $0'24 \cdot 0'31$ resulta $0'074$ que multiplicado por $0'93$ da $0'069$; el producto $0'31 \cdot 0'93$ resulta $0'29$, que multiplicado por $0'24$ da $0'068$.

Al calcular con coma flotante hay que tener en cuenta dos errores

importantes que pueden producirse. El primero de ellos es el de llegar a un exponente por encima de N_1, se produce entonces un error que la máquina detecta inmediatamente y que anuncia como *overflow*. El segundo es el de bajar del exponente $-N_0$; en este caso la máquina puede no anunciar el error y poner en la variable correspondiente el valor *cero* (el error aparecerá si, por ejemplo, tratamos de dividir por esa cantidad. Sabido lo anterior es fácil experimentar con nuestra máquina (por ejemplo, con nuestra calculadora) para determinar los valores de N_0 y N_1. Este tipo de error se conoce con el nombre de *underflow*.

OVERFLOW

UNDERFLOW

En el entorno MATLAB que vamos a utilizar existe una constante llamada `eps` que es el mayor número que sumado a 1 no produce efecto alguno.

3 Error

En todo cálculo numérico debemos esperar un error: los métodos que utilizamos en nuestros cálculos son generalmente aproximados; los datos de partida proceden de medidas de precisión limitada; las operaciones a que sometemos estos datos producen valores intermedios cuyo número de cifras debe reducirse; los valores numéricos que tratamos de calcular son generalmente reales y por tanto raramente expresable en términos finitos; etc. No debe por tanto preocuparnos el que en nuestros cálculos acarreemos un cierto error sino el controlar en todo momento ese error y mantenerlo por debajo de un extremo prefijado.

Llamaremos error de un resultado x_a a la diferencia entre éste y el valor x que tratamos de determinar. Más concretamente

$$E(x_a) = x_a - x.$$

Con esta definición del error en x_a estamos además estableciendo un signo en éste, criterio que mantendremos en el resto de estas notas.

El error E que acabamos de definir es el que llamaremos *error absoluto*. Más importante a la hora de determinar la validez de nuestros cálculos es el *error relativo*:

ERROR ABSOLUTO

ERROR RELATIVO

$$E_r(x_a) = \frac{E(x_a)}{x} = \frac{x_a - x}{x}$$

que está definido siempre que $x \neq 0$.

El error en un cálculo nunca podrá determinarse con exactitud (ello equivaldría a poder determinar el valor exacto de x) simplemente se acotará.

Una forma de referirse al error relativo en un sistema de numeración de base b es por medio del número de *cifras significativas* en un resultado. Diremos que x_a contiene n cifras significativas de x si la diferencia entre las mantisas de x_a y x es menor de $b/2$ unidades en su cifra $m+1$. Con esta definición la expresión n *cifras significativas* pudiera en algún caso querer decir algo ligeramente distinto de lo que coloquialmente pudiera entenderse. (Ejemplo: 3'1416 tiene 5 cifras significativas del número $\pi = 3'14159\ldots$.)

cifras significativas

4 Procedencia del error

El error en un cálculo puede proceder de diversas causas.

1. Equivocaciones.
2. Desviación del modelo con la realidad.
3. Errores de observación en los datos numéricos.
4. Errores debidos al funcionamiento de la máquina.
5. Errores intrínsecos de los métodos numéricos.

Hemos de señalar en primer lugar el error de cálculo, es decir el error que comete el calculista por equivocación. Incluimos aquí los errores de programación. La forma de evitar este tipo de error es la verificación de los cálculos y la comprobación de los programas. El verificar un programa completo puede resultar una tarea compleja y poco agradable. Tal vez la mejor estrategia sea la de dividir el programa en partes (subrutinas) y comprobar separadamente cada una de ellas experimentalmente con datos sencillos de los que es fácil deducir el resultado esperado.

Al resolver problemas físicos suelen darse dos tipos de errores. Primeramente aquellos que proceden de la medición de los datos. Los sistemas de medida nunca pueden ser exactos —en realidad, según nuestros modelos, tratamos las más de las veces de medir algo que consideramos un número real— y por tanto ya tendremos una ínfima desviación de partida. En segundo lugar debemos tener en cuenta que el modelo físico que utilizamos —las ecuaciones que modelan el experimento— suele obtenerse a través de aproximaciones que simplifican las propiedades físicas —linealización, ...— y que por tanto producirán desviaciones respecto de la realidad. Tanto unas como otras desviaciones no serán objeto de estudio por parte del Cálculo Numérico sino que de ellas se ocupa el estudio de la estabilidad del modelo —ecuaciones diferenciales, álgebra li-

neal....

También debemos considerar los errores debidos al funcionamiento de la máquina, entre los que debemos destacar los errores de redondeo, la pérdida de significación, el ruido al evaluar una función, y los llamados *overflow* y *underflow*.

De los errores de redondeo y del *overflow* y *underflow* ya hemos hablado más arriba. En cuanto a los errores debidos a la *pérdida de significación* en el número de dígitos, suelen aparecer al restar dos cantidades muy próximas. Supongamos que queremos evaluar la función

pérdida de significación

$$f(x) = x[\sqrt{x+1} - \sqrt{x}]$$

y lo hacemos utilizando una aritmética de seis dígitos. Obtenemos los siguientes resultados, donde $\bar{f}$ es el valor calculado directamente con la expresión dada de f.

x	$\bar{f}(x)$	$f(x)$
1	0'41421	0'414121
10	1'5434	1'5434
100	4'99	
1000	15'8	
10 000	50	
100 000	100	158'114
1 000 000	0	499'999

El error procede de la diferencia de los dos valores relativamente muy próximos $\sqrt{x+1}$ y $\sqrt{x}$ entre los que coinciden cada vez más cifras de las seis utilizadas en los cálculos.

Una forma de evitar el error es la de utilizar un número mayor de dígitos en los cálculos. Otra forma es la de evitar la diferencia señalada por medio de una expresión equivalente de la función, por ejemplo

$$f(x) = \frac{x}{\sqrt{x+1} + \sqrt{x}}$$

Ejercicio. Halla una expresión equivalente a

$$\frac{1-\cos x}{x^2}$$

que evite la diferencia entre dos números próximos que aparece en el numerador cuando x está próximo a cero.

ruido al evaluar una función

Por *ruido al evaluar una función* nos referimos a la indeterminación que puede aparecer en los valores de una función cuando se trata de analizar experimentalmente una propiedad cualitativa. Por ejemplo, si tratamos de ver donde está el cero del polinomio $P(x) = x^3 - 3x^2 + 3x - 1$ dando valores y utilizamos aritmética de seis dígitos

$$P(0'999) = 0, \qquad P(1'001) = 0$$

Incluso, con aritmética de siete dígitos

$$P(0'9998) = 0'0000001 > 0$$

cuando en realidad $P(0'9998) < 0$.

error al evaluar una función

Supongamos que queremos evaluar una función f, derivable, con derivada continua, en un punto x y que para ello disponemos de una aproximación $\tilde{x}$ de forma que lo que calcularemos es $f(\tilde{x})$. ¿Cuál es el error? Para evaluarlo podemos utilizar el teorema del punto medio (o, lo que viene a ser lo mismo, la definición de derivada),

$$|f(\tilde{x}) - f(x)| \approx f'(\tilde{x})|\tilde{x} - x|.$$

Es decir, el error se multiplica por el valor de la derivada de f (que será aproximadamente la misma en x que en $\tilde{x}$.

sumas

Al sumar un número grande de términos los errores de redondeo se acumulan. Veamos como proceder para conseguir que este error de redondeo sea lo menor posible. Queremos hallar $S = a_1 + a_2 + \cdots + a_n$ donde cada a_i es un `FLO`. Al realizar la suma debemos operar $n - 1$ veces:

$$\begin{array}{rrrr} S_2 = & \texttt{FLO}(a_1 + a_2) = & (a_1 + a_2)(1 + \varepsilon_2) \\ S_3 = & \texttt{FLO}(S_2 + a_3) = & (S_2 + a_3)(1 + \varepsilon_3) \\ \ldots \quad \ldots & \ldots \quad \ldots & \ldots \quad \ldots \\ S_n = & \texttt{FLO}(S_{n-1} + a_n) = & (S_{n-1} + a_n)(1 + \varepsilon_n) \end{array}$$

donde los ε_i son los errores relativos de redondeo.

Si S es la verdadera suma de $a_1 + a_2 + \cdots + a_n$ entonces

$$\begin{aligned} Sn - S &= S_n - (a_1 + \cdots + a_n) \\ &= S_{n-1}(1 + \varepsilon_n) + a_n\varepsilon_n - (a_1 + \cdots + a_{n-1}) \\ &= \ldots \\ &= a_1(\varepsilon_2 + \cdots + \varepsilon_n) + a_2(\varepsilon_2 + \cdots + \varepsilon_n) \\ &\qquad + \cdots + a_{n-1}(\varepsilon_{n-1} + \varepsilon_n) + a_n\varepsilon_n. \end{aligned}$$

Para que este término sea lo menor posible, la mejor estrategia es la de escribir $a_1 + a_2 + \cdots + a_n$ en orden creciente de los $|a_n|$, es decir sumar *de menor a mayor*.

Debemos mencionar finalmente los errores debidos a los métodos numéricos utilizados. Lo que calculan no es en general lo que se quiere calcular y por tanto el mismo método numérico debe hacer un estudio de cuál es el error que lleva asociado. Este estudio es parte fundamental de las técnicas analizadas en el presente curso.