

Tema 1

Introducción al Cálculo Numérico

1.1 Introducción

El *Cálculo Numérico*, o como también se le denomina, el *Análisis numérico*, es la rama de las Matemáticas que estudia los *métodos numéricos* de resolución de problemas, es decir, los métodos que permiten obtener una solución aproximada (en ocasiones exacta) del problema considerado tras realizar un número finito de operaciones lógicas y algebraicas elementales.

Los problemas que trata el Análisis numérico se pueden clasificar en dos grandes grupos, según tengan naturaleza numérica (o finito–dimensional) o naturaleza funcional (o infinito–dimensional). Pertenecen al primer grupo los problemas relativos a la resolución de sistemas de ecuaciones lineales, cálculo de valores y vectores propios, y resolución de ecuaciones y sistemas de ecuaciones no lineales. Son del segundo tipo, por el contrario, los problemas de interpolación y aproximación de funciones, la derivación e integración numéricas, los problemas de valor inicial y de contorno para ecuaciones diferenciales ordinarias, y los problemas de contorno para ecuaciones en derivadas parciales.

El desarrollo del Análisis numérico como disciplina con entidad propia ha ido indisolublemente ligado a la vertiginosa evolución que los ordenadores han experimentado desde su aparición en la década de los años cuarenta. No en vano, los ordenadores son herramientas imprescindibles para aplicar con eficacia la inmensa mayoría de los métodos que el Análisis numérico propone, dado el considerable volumen de cálculos y manipulaciones de datos que suelen llevar aparejados.

1.2 Noción de Algoritmo

Los diferentes problemas que trata el Cálculo Numérico son en sí procedimientos de aproximación que consisten en sucesiones de cálculos. Estos procedimientos se llaman algoritmos.

Un *algoritmo* es un procedimiento que describe, sin ambigüedad posible, una sucesión finita de pasos que hay que realizar en un orden preciso, desde la introducción de datos hasta la obtención de resultados.

Como vehículo para describir los algoritmos se usará un *pseudocódigo* que especifica tanto los datos de entrada, como el proceso que se debe realizar sobre los mismos para la obtención de los resultados deseados, así como la forma de salida de estos últimos. De esta forma, se puede esquematizar un algoritmo como sigue:

ENTRADA \Rightarrow PROCESO \Rightarrow SALIDA.

Por ejemplo, un algoritmo para el cálculo de la suma de N números reales x_1, \dots, x_N , es decir,

$$\sum_{i=1}^N x_i = x_1 + \dots + x_N,$$

puede ser descrito bajo el siguiente pseudocódigo:

ENTRADA Paso 1. Entrar N, x_1, \dots, x_n .

PROCESO Paso 2. Hacer $SUM = 0$.

Paso 3. Para $i = 1, \dots, N$,

hacer $SUM = SUM + x_i$.

SALIDA Paso 4. Escribir SUM .

PARAR.

Como segundo ejemplo de pseudocódigo para la descripción de un algoritmo, describamos el necesario para el cálculo del producto de las componentes no nulas de un vector de \mathbb{R}^N , es decir,

$$\prod_{\substack{i=1 \\ v_i \neq 0}}^N v_i.$$

Este algoritmo puede describirse mediante el siguiente pseudocódigo:

Paso 1.- Entrar N, v_1, \dots, v_N .

Paso 2.- Hacer $PROD = 1$.

Paso 3.- Para $i = 1 \dots, N$,

Si $v_i \neq 0$ entonces hacer $PROD = PROD * v_i$.

Paso 4.- Escribir $PROD$.

PARAR.

▽ Ejercicios

1.- Escribir un algoritmo que sume la serie finita $\sum_{i=1}^N x_i$ en orden inverso.

2.- Escribir un algoritmo que evalúe el polinomio

$$p(x) = \prod_{i=0}^N (x - x_i),$$

para un cierto número real x , conocidos el número de raíces $N + 1$ y el valor de éstas, x_0, \dots, x_N .

3.- Construir un algoritmo para el estudio de la existencia y el cálculo, en su caso, de las raíces de la ecuación $ax^2 + bx + c = 0$.

1.3 Fuentes de error: errores de redondeo y de discretización

La elaboración de algoritmos no sólo exige un buen conocimiento de los entes matemáticos y los métodos numéricos involucrados, sino también un cierto cuidado en la disposición de los cálculos para minimizar, en la medida de lo posible, los efectos indeseados e inevitables de los errores de redondeo que provienen del hecho de que el ordenador no puede utilizar más que una cantidad limitada de números racionales para representar todos los números reales.

El número máximo de decimales que considera el ordenador en la representación interna de un número real se llama *precisión*. Por ejemplo, si la precisión es $k = 9$, el número 0.123456789×10^2 representa, indistintamente, todos los números del intervalo $[0.1234567885 \times 10^2, 0.1234567895 \times 10^2)$.

A la diferencia entre el número considerado y su representación interna se le llama error de redondeo.

Definición 1.3.1 Si r^* es una aproximación del número real r , se llama error absoluto a $|r - r^*|$ y error relativo a $\frac{|r - r^*|}{|r|}$, siempre que sea $r \neq 0$.

Por ejemplo, el número $\pi = 3.14159265 \dots$ se representa, con una precisión $k = 5$, mediante el número 0.31416×10^1 , es decir, 3.1416. Por tanto, el error absoluto es

$$|3.14159265 \dots - 3.1416| = 0.00000735 \dots < 0.8 \times 10^{-5}.$$

Resulta, por tanto, que dos formulaciones equivalentes de un mismo problema pueden arrojar distintos resultados al efectuar los cálculos en un ordenador. Asimismo, al realizar largas cadenas de cómputos, los errores de redondeo pueden acumularse de tal modo que resten validez a los resultados, dando lugar a fenómenos llamados de inestabilidad numérica. Para controlar la propagación de errores de redondeo, es necesaria, por tanto, la búsqueda de algoritmos con un número mínimo de operaciones convenientemente dispuestas, manteniendo, de paso, el tiempo de cálculo dentro de unos límites aceptables.

Se dice que un proceso es *inestable* cuando los pequeños errores que se producen en una de sus etapas se agrandan en etapas posteriores y degradan seriamente la exactitud del cálculo en su conjunto.

Por ejemplo, la sucesión de término general $x_n = \left(\frac{1}{2}\right)^n$ puede ser también definida mediante la inducción:

$$(I) \begin{cases} x_0 = 1, x_1 = \frac{1}{2}, \\ x_n = \frac{23}{2}x_{n-1} - \frac{11}{2}x_{n-2}, n \geq 2, \end{cases}$$

ya que si, por hipótesis, $x_k = \left(\frac{1}{2}\right)^k$, para $k \leq n$, entonces

$$x_{n+1} = \frac{23}{2}x_n - \frac{11}{2}x_{n-1} = \left(\frac{23}{2} - 11\right)\left(\frac{1}{2}\right)^n = \left(\frac{1}{2}\right)^{n+1}.$$

Si consideramos una precisión $k = 6$, obtenemos que

$$x_{15} = 0.000031,$$

y mediante la inducción (I), se tiene:

$$\begin{aligned} x_0 &= 1, \\ x_1 &= 0.500000, \\ x_2 &= 0.250000, \\ &\vdots \\ x_{13} &= 0.936518, \\ x_{14} &= 10.300417, \\ x_{15} &= 113.303947, \end{aligned}$$

por lo que el error de redondeo es bastante significativo y desvirtúa la definición inductiva (I) de la sucesión.

Los errores de redondeo no son los únicos que aparecen en el tratamiento numérico de problemas matemáticos. Aparte de las posibles equivocaciones en el diseño de un método, en la programación de un algoritmo o en la entrada de datos, hay que considerar, por un lado, los errores inherentes al planteamiento del problema (al imponer, por ejemplo, hipótesis que simplifiquen un modelo complejo) y, por otro lado, a la obtención de datos iniciales (por ejemplo, errores de medida de datos físicos).

Un primer tipo de estos errores son los *errores de discretización o de truncatura*, originados al aproximar un proceso infinito por un proceso finito. En esencia, supongamos que la obtención de la solución s de un problema está dada por un algoritmo con un número no finito de pasos y que, si consideramos, para cada $n \in \mathbb{N}$, la solución x_n del paso n -ésimo, se verifica que

$$\lim x_n = s.$$

Definición 1.3.2 *Se llama discretización de un problema, dado por un proceso infinito a la sustitución del mismo por la solución de un problema con un número finito de pasos. El método numérico correspondiente se dice convergente si es capaz de proporcionar una solución aproximada del problema considerado para cualquier grado prefijado de exactitud, es decir, si x_n es la solución obtenida en n pasos y s es la solución exacta, dado $\varepsilon > 0$, existe $n_0 \in \mathbb{N}$ tal que*

$$|x_{n_0} - s| < \varepsilon.$$

Por ejemplo, el método numérico dado por la inducción:

$$(P_N) \begin{cases} x_0 = f(a), \\ x_n = x_{n-1} + \frac{f^{(n)}(a)}{n!}(x-a)^n, \quad 1 \leq n \leq N, \end{cases}$$

es una discretización del cálculo de $f(x)$ mediante el desarrollo limitado de Taylor de orden N de f centrado en a . Se sabe que, dado $\varepsilon > 0$, existen n_0 tal que, bajo ciertas condiciones,

$$|R_{n_0}(f, x)| = \left| \frac{f^{(n_0+1)}(\alpha)}{(n_0+1)!}(x-a)^{n_0+1} \right| < \varepsilon, \quad \alpha \in (a, x).$$

Por tanto,

$$|f(x) - x_{n_0}| = \left| f(x) - \sum_{n=0}^{n_0} \frac{f^{(n)}(a)}{n!}(x-a)^n \right| < \varepsilon,$$

de donde, se tiene que el método numérico (P_N) , para un cierto $N \in \mathbb{N}$, es una discretización consistente en calcular $f(x)$, mediante el desarrollo limitado de Taylor de orden N , que, bajo ciertas condiciones, es convergente.

Por último, siempre puede darse el caso de que el propio problema en sí sea bastante sensible a posibles perturbaciones de los datos iniciales, de modo que los resultados obtenidos registren grandes variaciones; en cuyo caso los errores de redondeo se acumulan a lo largo de la cadena de cálculos de cualquier método numérico empleado, sobre todo de aquellos poco estables, dando lugar a fuertes desviaciones de los resultados con respecto a la solución exacta. Por ello, habrá que ser especialmente cuidadoso a la hora de plantearse resolver numéricamente uno de estos problemas.

Definición 1.3.3 *Un problema matemático general se dice bien condicionado (mal condicionado) si pequeños cambios relativos en los datos de entrada, condiciones iniciales y/o parámetros, producen pequeños cambios (grandes cambios) relativos en la solución.*

Es decir, la palabra condicionamiento se usa para indicar la sensibilidad de la solución ante pequeños cambios relativos de los datos iniciales y posibles parámetros. A veces un exhaustivo estudio de dicha sensibilidad de la solución respecto de estos

datos iniciales y parámetros puede resultar de vital importancia de cara a una posterior resolución numérica e implementación práctica de un determinado algoritmo, para que efectivamente los resultados puedan ser fiables.

Un ejemplo sencillo de problema mal condicionado es la obtención de las raíces de una ecuación polinómica de segundo grado mediante fórmulas usuales. Por ejemplo, la mayor raíz de la ecuación

$$0.1x^2 - 20x + 20 = 0$$

es $x = 198.9949$ y, realizando un pequeño aumento de los coeficientes de 0.01, es decir, considerando la ecuación

$$0.11x^2 - 19.99x + 20.01 = 0$$

se obtiene un valor para la mayor raíz de $x = 180.7206$.

▽ Ejercicios

4.- Calcular el valor de $f(x) = \sqrt{x^2 + 1} - 1$ y de $g(x) = \frac{x^2}{\sqrt{x^2 + 1} + 1}$, con un ordenador o una calculadora, para una sucesión de valores $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots$. Aunque $f = g$, los resultados serán diferentes a partir de un cierto momento. ¿Cuáles de los resultados serán más de fiar y cuáles no?

5.- Existe una función f de la forma

$$f(x) = ax^7 + bx^8$$

para la cual $f(0.1) = 0.3 \times 10^{-5}$ y $f(0.9) = 0.012$. Determine a y b y evalúe la sensibilidad de estos parámetros respecto de pequeños cambios de valores de f .

1.4 Rapidez y coste computacional

Una vez establecida la convergencia de un método, surge el problema, más delicado, de estimar la rapidez con que las soluciones aproximadas tienden a la solución exacta. Si el método converge lentamente, el número de cálculos necesarios para alcanzar

una precisión aceptable puede llegar a ser enorme, haciendo prohibitiva la aplicación del método.

Aun cuando la alta rapidez de convergencia sea un mérito, a la hora de comparar métodos numéricos hay que tener en cuenta otros aspectos, como pueda ser la complejidad de los cálculos que el método requiere. En ocasiones, se prefiere elegir un método que converja más lentamente si con él los cálculos son más sencillos de efectuar: puede que a posteriori necesite realizar menos operaciones elementales para alcanzar un mismo grado de exactitud (técnicamente se hablará más bien de orden de aproximación o de convergencia). En este sentido, es aleccionadora la comparación de los métodos de la secante y de Newton para la resolución de ecuaciones no lineales.

El *coste computacional* de un método numérico puede ser medido también en función de la cantidad de memoria del ordenador que se precisa para almacenar los datos iniciales, resultados y cómputos intermedios. En equipos informáticos de dimensiones pequeñas o medias, las limitaciones de memoria pueden ser un factor determinante en la elección del método.

▽ Ejercicios

Calcular el número de operaciones necesarias para resolver cada uno de los siguientes problemas:

6.- Hallar $\sum_{i=1}^N x_i$, con $x_i \in \mathbb{R}$, para $i = 1, \dots, N$.

7.- Resolver el sistema de ecuaciones lineales con coeficientes reales:

$$\begin{aligned}ax + by &= d, \\cx + dy &= f,\end{aligned}$$

por el método de sustitución, si $ad - bc \neq 0$.

8.- Resolver el sistema anterior por el método de reducción.

9.- Hallar el producto escalar de dos vectores de \mathbb{R}^N .