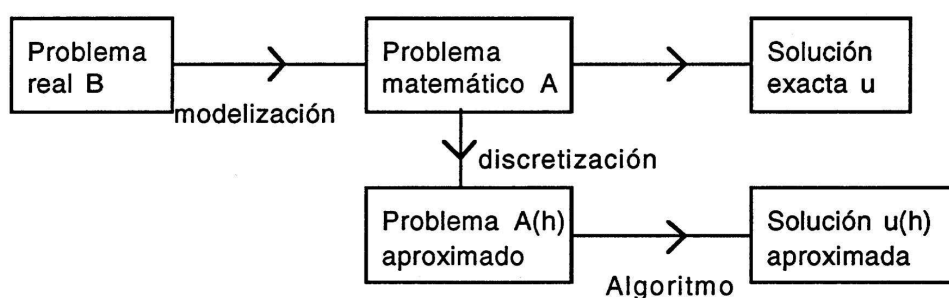


CAPITULO II. ANALISIS DE LOS ERRORES

1. ESQUEMA DE RESOLUCION NUMERICA DE UN PROBLEMA

Si se desea resolver un problema físico B , lo primero que se suele hacer es traducirlo al lenguaje matemático para dar un problema matemático A . Se estudia la existencia y unicidad de la solución u de este problema, pero en la mayor parte de los casos y después de probado esto, no se sabe cómo determinar la solución de forma efectiva. Por ello, se sustituye el problema matemático A por un problema proximo a él, A_h , en el que aparecerá algún parámetro h que se va a hacer tender hacia un cierto valor (normalmente 0). Se exige que este problema tenga solución única, u_h , y se espera que al tender h hacia el valor elegido, u_h converja hacia u . Esquemáticamente este tratamiento típico (pero no único), es el siguiente:



De este planteamiento surgen algunos problemas interesantes:

- ¿Cuál es la **velocidad de convergencia** de u_h hacia u ?
- Problemas de estabilidad**; es inevitable cometer errores en el cálculo, debido a los redondeos que efectúan los computadores. Interesa que pequeños errores cometidos en los cálculos que conducen a u_h hagan que el resultado no difiera mucho de u ; (de eso hablaremos más en el siguiente párrafo).
- Coste del proceso**. ¿Cuántas operaciones deben realizarse? ¿Cuánto tiempo se precisará para realizarlas?

Veamos ahora unos ejemplos que muestran la importancia de esas últimas cuestiones:

Ejemplo 1.

Supongamos que se necesita evaluar el polinomio

$$p(x) = 3x^4 + 4x^3 + 5x^2 + 2x + 1$$

que es equivalente a:

$$p(x) = (((3x + 4)x + 5)x + 2)x + 1$$

El número de operaciones para evaluarlo en el primer caso es de:

$$4 + 3 + 2 + 1 = 10 \text{ Multiplicaciones}$$

$$4 \text{ Sumas,}$$

mientras que en el segundo se requieren solamente:

4 Multiplicaciones

4 Sumas.

En el caso general de un polinomio de orden n :

$$p(x) = a_0 x^n + a_1 x^{n-1} + \dots + a_{n-1} x + a_n ,$$

$$p(x) = ((\dots((a_0 x + a_1) x + a_2) x + \dots + a_{n-1}) x + a_n) ,$$

sólo hay diferencia en el número de multiplicaciones (el de sumas es n en ambos casos), que es de

$$n + (n-1) + \dots + 1 = \frac{(n+1)n}{2} \approx \frac{n^2}{2} \text{ en el primer caso y } n \text{ en el segundo .}$$

Se comprende pues, que es preferible usar el segundo método porque exige menos operaciones (y por lo tanto existen menos posibilidades de que se propaguen los errores de redondeo, lo que dará lugar a una solución más exacta). El algoritmo que lleva a evaluar el polinomio con el segundo método se denomina **algoritmo de Horner** y es:

$$b_0 = a_0$$

$$b_i = a_i + b_{i-1} x, i = 1, \dots, n .$$

Ejemplo 2.

Para resolver sistemas de ecuaciones de orden n con el método de Cramer se precisa un total de $(n+1)!(n-1)$ operaciones (multiplicaciones) (cada determinante exige $n!$ $(n-1)$ multiplicaciones $a_{p(1)} a_{p(2)} \dots a_{p(n)}$ y hay $n+1$ determinantes a calcular).

El método de Gauss (que explicaremos en un capítulo posterior) exige, sin embargo, sólo $\frac{n^3}{3}$ operaciones. Así, una tabla comparativa de estos métodos sería:

n	5	10	50	100
Cramer	2880	3.99*10 ⁶	7*10 ⁶⁷	16010
Gauss	42	334	42000	334000

Por ejemplo, para $n = 5$, haciendo una operación cada medio minuto (manualmente) se tardarían 24 horas en resolver el sistema por el método de Cramer, mientras que por el de Gauss se tardarían sólo 21 minutos.

Si se intentase utilizar el método de Cramer para resolver un sistema de orden 15 en un ordenador que efectuase 10^6 operaciones por segundo, tardaría más de 9 años en obtener la solución, que además posiblemente no se parecería en nada a la solución verdadera debido a los errores de redondeo que se hubieran producido. ¡Con el método de Gauss, el mismo ordenador tardaría centésimas de segundo!

Este último ejemplo justifica suficientemente la necesidad de buscar algoritmos que sean prácticos.

2. DISTINTOS TIPOS DE ERRORES

Generalmente el resultado de un cálculo numérico es aproximado (sólo en casos excepcionales es exacto), y por eso necesitamos conocer la precisión.

Si p y p^* son dos números reales y p^* se considera como aproximación de p , una medida de la precisión de p^* es

$$E = |p - p^*|.$$

De costumbre el conocimiento de E no basta para establecer si p^* es una aproximación buena de p . Por ejemplo:

$$p_1 = 5.1346, p_1^* = 5.1345$$

$$E = |p_1 - p_1^*| = 10^{-4},$$

y

$$p_2 = 0.0005, p_2^* = 0.0004$$

$$E = |p_2 - p_2^*| = 10^{-4}.$$

En los dos casos E es igual a 10^{-4} , pero sólo en el primer caso pensamos que p_1^* es una buena aproximación de p_1 . En el segundo caso, p_2 y E son del mismo orden de magnitud, y entonces nos parece mejor considerar su razón.

Damos entonces la siguiente definición: si p^* es una aproximación de p , el **error absoluto** está dado por $E_a = |p - p^*|$, y el **error relativo** está dado por $E_r = \frac{|p - p^*|}{|p|}$, siempre y cuando $p \neq 0$.

En el ejemplo previo, obtenemos:

$$E_r = \frac{|p_1 - p_1^*|}{|p_1|} = 0.000019476,$$

$$E_r = \frac{|p_2 - p_2^*|}{|p_2|} = 0.2.$$

Muchas son las causas que pueden interferir en la precisión de un cálculo, y generar errores. Esos errores se pueden clasificar en:

- a) errores iniciales;
- b) errores de redondeo;
- c) errores de truncamiento;
- d) errores de propagación.

Los **errores iniciales** no se pueden evitar si, por ejemplo, son el resultado de medidas de precisión limitada. Supongamos que debemos calcular $f(x)$ en un cierto punto x . Puede ocurrir que estemos obligados a sustituir x por x' , con lo cual se calculará $f(x')$ en vez de $f(x)$. Se llama error inicial al valor $f(x') - f(x) = \varepsilon_i$.

Los **errores de redondeo** son debidos a redondeos en los cálculos porque están hechos con un número finito de cifras significativas. Entonces, y continuando con el ejemplo previo, no calcularemos $f(x')$ sino $f_1(x')$. El valor $f_1(x') - f(x') = \varepsilon_r$ se llama error de redondeo.

Los **errores de truncamiento** generalmente corresponden a truncamientos de procedimientos infinitos (desarrollos en serie, etc.). En el ejemplo previo puede ocurrir que

f (y f_1) sea poco manejable y estamos obligados a sustituirla por otra función próxima a ella, f_2 . El valor $f_2(x') - f_1(x') = \varepsilon_t$ es llamado error de truncamiento o de discretización.

Aquí es útil, por ejemplo, recordar el **Teorema de Taylor**: supongamos que $f \in C^n[a, b]$ y $f^{(n+1)}$ existe en $[a, b]$. Sea $x_0 \in [a, b]$. Para toda $x \in [a, b]$, existe $\xi(x)$ entre x_0 y x tal que

$$f(x) = P_n(x) + R_n(x)$$

donde

$$\begin{aligned} P_n(x) &= f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \dots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n \\ &= \sum_{k=0}^n \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k \end{aligned}$$

y

$$R_n(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x - x_0)^{(n+1)}.$$

A $P_n(x)$ se le llama el polinomio de Taylor de grado n para f alrededor de x_0 y a $R_n(x)$ se le llama el residuo (o error de truncamiento) asociado con $P_n(x)$. La serie infinita que se obtiene tomando el límite de $P_n(x)$ cuando $n \rightarrow \infty$ se denomina Serie de Taylor para f alrededor de x_0 . En el caso de que $x_0 = 0$, el polinomio de Taylor se conoce frecuentemente como polinomio de MacLaurin, y la serie de Taylor se denomina serie de MacLaurin.

Los **errores de propagación** son debidos a la propagación de errores previos en el algoritmo.

Ejemplo 3.

Supongamos que se desea calcular $e^{\sqrt{2}/8}$, y que disponemos de una calculadora de seis dígitos significativos que no dispone de la función exponencial. Es sabido que

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2} + \dots$$

- El primer paso consistirá en calcular $\sqrt{2} \approx 1.41421$
- A continuación el punto en el que se evalúa la función será $x' = 1.41421/8$ en vez del verdadero $\sqrt{2}/8$. El error que estamos cometiendo $f(x') - f(x) = \varepsilon_i$ es el error inicial.
- Pasamos a utilizar una función aproximada a e^x , como puede ser los tres primeros términos de la serie. Se comete así el error de truncamiento $\varepsilon_t = 1 + x' + \frac{x'^2}{2} - e^x$.
- La necesidad de utilizar 6 dígitos significativos hace que las operaciones se hagan sólo con esos 6 dígitos, con lo cual se pierden partes decimales: se está produciendo el error de redondeo.

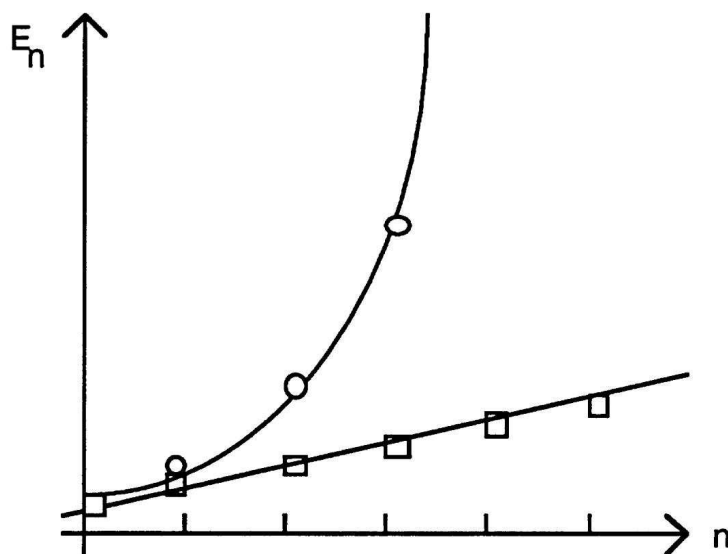
3. CONVERGENCIA

Hemos dicho ya que los cálculos que involucran aproximaciones en la máquina pueden resultar en el crecimiento de los errores de redondeo. Por supuesto, estamos interesados en escoger métodos que produzcan resultados fiables en su precisión. Un criterio que impondremos en un algoritmo, cuando sea posible, es que cambios pequeños en los datos iniciales produzcan correspondientemente cambios pequeños en los resultados finales. Un algoritmo que satisface esta propiedad se llama **estable**. Es **inestable** cuando este criterio no se cumple. Algunos algoritmos serán estables para ciertos grupos de datos iniciales pero no para todos. Se tratará, siempre que se pueda, de caracterizar las propiedades de estabilidad de los algoritmos.

Para considerar un poco más el tema del crecimiento del error de redondeo y su conexión con la estabilidad de los algoritmos, supongamos que se introduce un error ε en alguna etapa de los cálculos y que el error después de n operaciones subsecuentes se denota por E_n . Los dos casos que se presentan más frecuentemente en la práctica se definen a continuación.

Definición. Supongamos que E_n representa el crecimiento del error después de n operaciones subsecuentes. Si $|E_n| \approx C n \varepsilon$, donde C es una constante independiente de n , se dice que el crecimiento del error es **lineal**. Si $|E_n| \approx k^n \varepsilon$, para algún $k > 1$, el crecimiento del error es **exponencial**.

El crecimiento lineal del error es usualmente inevitable, y cuando C y ε son pequeños los resultados son generalmente aceptables. El crecimiento exponencial del error debe ser evitado, ya que el término k^n será grande aún para valores pequeños de n . Esto lleva a imprecisiones inaceptables, no importando la magnitud de ε . Como consecuencia, un algoritmo que exhibe crecimiento lineal del error es estable, mientras que un algoritmo en el que el crecimiento del error es exponencial es inestable.



Ejemplo 4.

La sucesión $p_n = (\frac{1}{3})^n$, $n > 0$, puede generarse recursivamente tomando $p_0 = 1$ y definiendo $p_n = (\frac{1}{3}) p_{n-1}$, para $n > 1$. Si obtenemos la sucesión de esta manera, usando aritmética de redondeo a cinco dígitos, los resultados vienen dados en la tabla 1.

El error de redondeo introducido en reemplazar $\frac{1}{3}$ por 0.33333 produce un error de sólo $(0.33333)^n \times 10^{-5}$ en el n -ésimo término de la sucesión. Este método de generar la sucesión es claramente estable.

Tabla 1

n	p_n
0	0.10000×10^1
1	0.33333×10^0
2	0.11111×10^0
3	0.37036×10^{-1}
4	0.12345×10^{-1}

Otra manera de generar la sucesión es definiendo $p_0 = 1$, $p_1 = \frac{1}{3}$, y calculando para cada $n \geq 2$,

$$p_n = \left(\frac{10}{3}\right) p_{n-1} - p_{n-2} .$$

La tabla 2 muestra los resultados tanto exactos como redondeados a cinco dígitos usando esta fórmula.

Tabla 2

n	p_n calculado	p_n exacto
0	0.10000×10^1	0.10000×10^1
1	0.33333×10^0	0.33333×10^0
2	0.11111×10^0	0.11111×10^0
3	0.37000×10^{-1}	0.37037×10^{-1}
4	0.12230×10^{-1}	0.12346×10^{-1}
5	0.37660×10^{-2}	0.41152×10^{-2}
6	0.32300×10^{-3}	0.13717×10^{-2}
7	-0.26893×10^{-2}	0.45725×10^{-3}
8	-0.92872×10^{-2}	0.15242×10^{-3}

Este método es obviamente inestable.

Nótese que la fórmula dada, $p_n = (\frac{10}{3}) p_{n-1} - p_{n-2}$, se satisface si p_n es de la forma

$$p_n = C_1 \left(\frac{1}{3}\right)^n + C_2 3^n$$

para cualquier par de constantes C_1 y C_2 . Para verificar esto, notemos que

$$\begin{aligned}
 \frac{10}{3} p_{n-1} - p_{n-2} &= \frac{10}{3} [C_1 \left(\frac{1}{3}\right)^{n-1} + C_2 3^{n-1}] - [C_1 \left(\frac{1}{3}\right)^{n-2} + C_2 3^{n-2}] \\
 &= C_1 \left[\frac{10}{3} \left(\frac{1}{3}\right)^{n-1} - \left(\frac{1}{3}\right)^{n-2} \right] + C_2 \left[\frac{10}{3} 3^{n-1} - 3^{n-2} \right] \\
 &= C_1 \left(\frac{1}{3}\right)^n + C_2 3^n = p_n .
 \end{aligned}$$

Para tener $p_0 = 1$ y $p_1 = \frac{1}{3}$, las constantes C_1 y C_2 deben escogerse como $C_1 = 1$ y $C_2 = 0$. Sin embargo, en la aproximación de cinco dígitos, los dos primeros términos son $p_0 = 0.10000 \times 10^1$ y $p_1 = 0.33333 \times 10^0$, los cuales requieren una modificación de estas constantes a $C_1 = 0.10000 \times 10^1$ y $C_2 = -0.12500 \times 10^{-5}$. Este pequeño cambio en C_2 da lugar a un error de redondeo de $3^n(-0.12500 \times 10^{-5})$ al producir p_n . Como consecuencia resulta un crecimiento exponencial del error, lo cual se refleja en la pérdida extrema de exactitud encontrada en la tabla 2.

Para reducir los efectos del error de redondeo, podemos usar una aritmética de un orden grande de dígitos, como las opciones de doble o múltiple precisión, disponibles en la mayoría de las computadoras digitales. Una desventaja del uso de la aritmética de doble precisión es que toma mucho más tiempo de computadora. Por otro lado, no se elimina completamente el crecimiento serio del error de redondeo, sino que sólo se postpone si es que se realizan un gran número de cálculos posteriores. Hay también otros métodos para estimar el error de redondeo (aritmética de intervalo, métodos estadísticos, etc.) que no estudiaremos.

EJERCICIOS.

1. Considerar los siguientes valores de p y p^* . ¿Cuál es el (i) error absoluto, (ii) error relativo al aproximar p^* por p ?
 - a) $p = \pi$, $p^* = 3.1$;
 - b) $p = \frac{1}{3}$, $p^* = 0.333$;
 - c) $p = \frac{\pi}{1000}$, $p^* = 0.0031$;
 - d) $p = \frac{100}{3}$, $p^* = 33.3$;
 - e) $p = \pi$, $p^* = 3.141$;
 - f) $p = \frac{1}{3}$, $p^* = 0.33333$;
 - g) $p = \frac{\pi}{1000}$, $p^* = 0.00314$;
 - h) $p = \frac{100}{3}$, $p^* = 33.33$.
2. Considerar los siguientes valores de p y p^* . ¿Cuál es el (i) error absoluto, (ii) error relativo al aproximar p^* por p ?
 - a) $p = e$, $p^* = 2.5$;
 - b) $p = \frac{10}{3}$, $p^* = 3.33333$;
 - c) $p = \frac{e}{1000}$, $p^* = 0.0025$;
 - d) $p = \frac{10000}{3}$, $p^* = 3333.333$.
3. Calcular los siguientes valores en el caso que se disponga de una calculadora de seis dígitos significativos que no dispone de la función exponencial (usar el desarrollo en serie de Taylor hasta el cuarto orden):
 - a) e^2 ;
 - b) $e^{-1} + e^3$.