1

Introducción y errores

Introducción

De una forma sencilla, el Cálculo Numérico se puede definir como la rama del Análisis Matemático que estudia y desarrolla procedimientos matemáticos para resolver problemas con ayuda del ordenador.

Se pueden resolver de forma aproximada problemas que no tienen solución en el Análisis Matemático tradicional. Las únicas operaciones que se realizan son +, -, *,/ y comparaciones y los resultados son siempre numéricos y aproximados de la solución exacta del problema.

Los problemas que se estudian abarcan un amplio rango de campos como son la resolución de ecuaciones no lineales, grandes sistemas de ecuaciones lineales, interpolación y aproximación numérica, derivación e integración de funciones, resolución de ecuaciones diferenciales ordinarias y en derivadas parciales y optimización entre otros.

Para implementar los métodos numéricos se pueden utilizar distintos softwares como por ejemplo los de uso general: FORTRAN, C, C++,.. También se puede utilizar un software matemático como MATLAB, MAPLE, Mathematica o Derive, que permiten el cálculo numérico y simbólico, trabajar con valores exactos o hacer representaciones gráficas de forma muy sencilla, a la vez que incorporan comandos como Do, For o While que permiten repetir muchas veces un conjunto de operaciones.

Aritmética del ordenador

La aritmética de las calculadoras u ordenadores es distinta de la tradicional que se utiliza en cursos de cálculo o álgebra.

En la aritmética tradicional se pueden manejar números con infinitas cifras decimales que no se repiten, como por ejemplo al representar $\sqrt{3}$. Esto es lo que se denomina *aritmética exacta*. Sin embargo, en los ordenadores o calculadoras, los números que se pueden representar y los resultados que se obtienen en las distintas operaciones solo pueden tener un número finito de dígitos.

Notación Científica

La notación *científica* consiste en representar un número utilizando potencias de base diez : $a \times 10^n$, siendo *a* un número entero o con coma decimal mayor o igual que 1 y menor que 10, y *n* un número entero que se denomina *exponente* u *orden de magnitud*. Con esta notación resulta muy cómodo representar números muy grandes en los que aparecerá una potencia de diez de exponente positivo, o números muy pequeños con una potencia de diez de exponente negativo.

Por ejemplo:

$$0.000351 = 3.51\times10^{-4} = 3.51E-4$$
$$27.1828 = 2.71828\times10 = 2.71828E+1$$
$$58000000 = 5.8\times10^{7} = 5.8E+7$$

La notación decimal en punto flotante *normalizada* consiste en representar cualquier número en la forma:

$$\pm 0.d_1d_2\cdots d_k\times10^{n} \quad , \quad 1\le d_1\le 9 \;\; y \;\; 0\le d_i\le 9 \;\; i=2,3\cdots k$$

Representación de números en el ordenador

Los ordenadores usan para los números reales una representación binaria en coma flotante normalizada. Esto significa que lo que almacena el ordenador no es una cantidad numérica *x*, sino una aproximación binaria a *x*:

$$x \approx \pm M \times 2^{n}$$

El número *M* se llama mantisa. El número entero *n* se llama exponente.

Precisión de un ordenador

Como se ha dicho anteriormente, las máquinas no pueden guardar tantos dígitos como se quiera de una determinada cantidad. Por esto tampoco se pueden almacenar cantidades tan grandes o tan pequeñas como se quiera. La precisión del ordenador viene determinada por el menor y mayor valor con los que puede trabajar. Si el resultado de una operación es menor que el menor valor que puede representar el resultado se hace igual a cero (*underflow*) y si es mayor que el mayor valor que puede representar el resultado es una parada de los cálculos (*overflow*).

Dígitos significativos

Los dígitos significativos de un número, son aquellos que pueden ser empleados en forma fiable para describir una cantidad.

Es importante establecer que los ceros, no son siempre dígitos significativos, ya que pueden emplearse para ubicar el punto decimal, por ejemplo:

a) 0.00001845

b) 0.0001845

c) 0.001845

d) 0.0000180

Los apartados a, b y c, tienen cuatro dígitos significativos, donde el número 1 es el primer dígito significativo (dígito significativo principal o dígito más significativo), el 8 es el segundo dígito significativo, el 4 es el tercer dígito significativo y el 5 es el cuarto. El apartado d tiene tres dígitos significativos: 1, 8 y 0.

Por otro lado el número 45300 puede tener 3, 4 ó 5 dígitos significativos, dependiendo los ceros que se conocen con exactitud. Podemos representar esta cantidad utilizando la notación científica normalizada:

a) 0.453 x 10^5, tres dígitos significativos.

b) 0.4530 x 10^5, cuatro dígitos significativos.

c) 0.45300 x 10^5, cinco dígitos significativos.

Definición: Si $\hat{p}$ es un valor aproximado de un valor exacto p, se dice que aproxima a p hasta el t^o dígito significativo si:

$$\frac{|p-\hat{p}|}{|p|} \leq 5\times 10^{-t}$$

Al desarrollar métodos numéricos utilizando un software matemático como Mathematica es importante que quede clara la diferencia entre dígitos significativos y cifras decimales de una cantidad que no tienen por qué coincidir.

Incertidumbre en los datos

Los datos de los problemas que se presentan en la realidad contienen incertidumbre o error. Este tipo de error se conoce como ruido y afectará a la exactitud de cualquier cálculo numérico que se base en dichos datos. No podemos mejorar la precisión de los cálculos si realizamos operaciones con datos afectados por ruido. Así, si empezamos con datos que contienen d cifras significativas, el resultado de un cálculo con ellos debería mostrarse con d cifras significativas; p.ej., supongamos que los datos $p_1 = 4.152$ y $p_2 = 0.07931$ tienen ambos una precisión de cuatro cifras, entonces sería tentador indicar todas las cifras que aparecen en la pantalla de una calculadora al hacer, digamos su suma: $p_1 + p_2 = 4.23131$. Esto no es correcto, no deberíamos obtener conclusiones que tengan más cifras significativas que los datos de partida. Así el resultado obtenido de la suma será $p_1 + p_2 = 4.231$.

Distintos tipos de errores

Error absoluto y error relativo

En la práctica del cálculo numérico es importante tener en cuenta que las soluciones calculadas no son soluciones matemáticas exactas en la mayoría de los casos. La precisión de una solución numérica puede verse disminuida por diversos factores, y la comprensión de estas dificultades es importante para desarrollar o a construir algoritmos numéricos adecuados.

Definición. Supongamos que $\hat{p}$ es una aproximación a p. El error absoluto de la aproximación es $E_p = |p-\hat{p}|$; y el error relativo es $R_p = \frac{|p-\hat{p}|}{|p|}$, supuesto que $p \neq 0$.

El error absoluto no es más que la distancia entre el valor exacto y el valor aproximado, mientras que el error relativo mide el error entendido como una porción del valor exacto.

Ejemplo

Calcular el error absoluto y el error relativo siendo el valor exacto $x = 3.141592$ y el valor aproximado, $\hat{x} = 3.14$.

El error absoluto es:

$$E_x = |x - \hat{x}| = |3.141592 - 3.14| = 0.001592$$

y el error relativo:

$$R_x = \frac{|x - \hat{x}|}{|x|} = \frac{0.001592}{3.141592} \approx 0.00507$$

Cuando se implementa un método numérico mediante un algoritmo iterativo, en general, no se conoce el valor exacto. En este caso, en cada etapa de iteración se utiliza lo que se puede denominar como "error aproximado" o "error relativo aproximado" que se pueden definir de la siguiente forma:

$$e_a = x_i - x_{i-1} \qquad r_a = \frac{x_i - x_{i-1}}{x_i}$$

donde x_i es el valor aproximado de la solución exacta del problema que resulta en la iteración iª y x_{i-1} en la iteración anterior.

En métodos numéricos suele establecerse una tolerancia porcentual como criterio de parada. En cada iteración se calculará el "error relativo aproximado" que se comparará con la tolerancia establecida de forma que el proceso iterativo finaliza cunado $r_a < t$, siendo *t* la tolerancia fijada de antemano. Cuanto menor sea la tolerancia mayor será la precisión del método aunque esto evidentemente supone un mayor número de iteraciones.

Error de truncamiento

La noción de error de truncamiento se refiere normalmente a los errores que se producen cuando una expresión matemática complicada se "reemplaza" por una fórmula más simple. Esta terminología se originó en la sustitución de una función por uno de sus polinomios de Taylor. Por ejemplo, podríamos reemplazar la serie de Taylor

$$sen(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \frac{x^{11}}{11!} + \cdots$$

por los cinco primeros términos $x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!}$ a la hora de calcular una integral numéricamente.

Ejemplo.

Sabiendo que: $\int_0^1 sen(x)dx = 0.459697694132 = p$

vamos a determinar la precisión de la aproximación obtenida al reemplazar el integrando $f(x) = sen(x)$ por los 5 primeros términos de su desarrollo en serie de Taylor.

Integrando término a término este polinomio, obtenemos

$$\int_0^1 x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!}\,dx = \left(\frac{x^2}{2!} - \frac{x^4}{4!} + \frac{x^6}{6!} - \frac{x^8}{8!} \right)_0^1 =$$

$$\frac{1}{2!} - \frac{1}{4!} + \frac{1}{6!} - \frac{1}{8!} = \frac{3707}{8064} = 0.459697420635 = \hat{p}$$

Puesto que los valores de p y $\hat{p}$ coinciden hasta la 6ª cifra decimal diremos que el error cometido al sustituir p por $\hat{p}$ es menor que 10^{-6}.

Error de redondeo

La representación de los números reales en un ordenador está limitada por el número de cifras de la mantisa, de manera que algunos números no coinciden exactamente con su representación en la máquina. Esto es lo que se conoce como **error de redondeo**. El número que, de hecho, se guarda en la memoria del ordenador puede haber sufrido el redondeo de su última cifra; en consecuencia, y puesto que el ordenador trabaja con números que tienen una cantidad limitada de dígitos, los errores de redondeo se introducen y propagan a través de operaciones sucesivas.

El redondeo se puede hacer de dos formas distintas.

Truncado: Cuando no se modifica o altera el último dígito que no se descarta.

Redondeo (simétrico): Si el primer dígito que se va a descartar es menor que 5 no se modifica el anterior, mientras que si es mayor o igual que 5, el último dígito no descartado aumenta en una unidad.

Por ejemplo, consideremos el número real:

$$p = 22/7 = 3.14285714285142857\ldots$$

La representación en coma flotante normalizada con redondeo a seis cifras significativas, tiene los dos resultados siguientes:

$$fl_{trun}(p) = 0.314285 \times 10^1 \; ; \quad fl_{\text{red}}(p) = 0.314286 \times 10^1 .$$

En las aplicaciones de ingeniería, en general, se utiliza el redondeo simétrico ya que el redondeo por truncado supone una pérdida de información.

Pérdida de cifras significativas

Consideremos los números p = 3.1415926536 y q = 3.1415957341, que son casi iguales y están ambos expresados con una precisión de 11 cifras significativas. Si calculamos su diferencia $p-q$ = -0.0000030805 vemos que, como las seis primeras cifras de p y de q coinciden, su diferencia $p-q$ sólo contiene cinco cifras decimales; este fenómeno se conoce como **pérdida de cifras significativas** o **cancelación**, y hay que tener cierto cuidado con él porque puede producir sin que nos demos cuenta, una reducción en la precisión de la respuesta final calculada.

Ejemplo.

Evaluemos la función *f(x)=1-Cos(x)*.

Para valores próximos a cero, se produce un efecto de cancelación.

Calculemos el valor de *f*(0.05359) utilizando cinco cifras significativas en las operaciones.

El resultado es: *f*(0.05359)=1-0.99856=0.00144

Al restar dos cantidades prácticamente idénticas se produce el efecto de cancelación por lo que en lugar de disponer de las cinco cifras significativas iniciales el resultado queda tan solo con tres.

Para evitar este problema, se puede utilizar una función en la que no se produzca este efecto de cancelación. La función $g(x)=\frac{Sen^2(x)}{1+Cos(x)}$ evita este problema

$$f(x)=1-Cos(x)=1-Cos(x)*\frac{1+Cos(x)}{1+Cos(x)}=\frac{(1-Cos(x))*(1+Cos(x))}{1+Cos(x)}=\frac{1-Cos^2(x)}{1+Cos(x)}=\frac{Sen^2(x)}{1+Cos(x)}$$

Evaluando *g*(0.05359) resulta:

$$g(0.05359)=\frac{0.0028691}{1+0.99856}=0.0014356\, g(0.05359)=$$

De este modo se dispone de las cinco cifras significativas y el resultado es mas exacto que el que se obtiene a partir de la función *f(x)*.

Propagación del error

Los cálculos que realiza el ordenador tampoco son exactos ya que los operandos no son valores exactos en general, y, además, los errores cometidos se van propagando en operaciones sucesivas. Así, si un resultado está afectado de cierto error y posteriormente se multiplica por un número grande o se divide por uno pequeño, el error se amplifica.

Vamos a ver ahora cómo pueden propagarse los errores en una cadena de operaciones sucesivas. Consideremos la suma de dos números *p* y *q* (que son valores exactos) con valores aproximados y $\hat{q}$ cuyos errores son ε_p y ε_q, respectivamente. A partir de $p=\hat{p}+\varepsilon_p$ y de $q=\hat{q}+\varepsilon_q$, la suma es

$$p+q=(\hat{p}+\varepsilon_p)+(\hat{q}+\varepsilon_q)=(\hat{p}+\hat{q})+(\varepsilon_p+\varepsilon_q)$$

Por tanto, el error en una suma es la suma de los errores de los sumandos.

La propagación del error en una multiplicación es más complicada. El producto es

$$pq=(\hat{p}+\varepsilon_p)(\hat{q}+\varepsilon_q)=\hat{p}\hat{q}+\hat{p}\varepsilon_q+\hat{q}\varepsilon_p+\varepsilon_p\varepsilon_q$$

Por tanto, si $\hat{p}$ y $\hat{q}$ son mayores que 1 en valor absoluto, los términos $\hat{p}\varepsilon_q$ y $\hat{q}\varepsilon_p$ indican que hay una posibilidad de que los errores originales ε_p y ε_q sean magnificados. Si se analiza los errores relativos, se tiene una percepción más clara de la situación. Reordenando los términos :

$$pq-\hat{p}\hat{q}=\hat{p}\varepsilon_q+\hat{q}\varepsilon_p+\varepsilon_p\varepsilon_q$$

Supongamos que $p\neq 0$ y que $q\neq 0$; entonces podemos dividir entre $p\cdot q$ para obtener el error relativo del producto pq :

$$R_{pq} = \frac{pq - \hat{p}\hat{q}}{pq} = \frac{\hat{p}\varepsilon_q + \hat{q}\varepsilon_p + \varepsilon_p\varepsilon_q}{pq} = \frac{\hat{p}\varepsilon_q}{pq} + \frac{\hat{q}\varepsilon_p}{pq} + \frac{\varepsilon_p\varepsilon_q}{pq}$$

Es más, suponiendo que $\hat{p}$ y $\hat{q}$ son buenas aproximaciones de p y q; entonces $\hat{p}/p \approx 1$, $\hat{q}/q \approx 1$ y $R_pR_q = \left(\varepsilon_p/p\right)\left(\varepsilon_q/q\right) \approx 0$ (R_p y R_q son los errores relativos de las aproximaciones $\hat{p}$ y $\hat{q}$). Sustituyendo estas aproximaciones en R_{pq} se obtiene una relación más simple:

$$R_{pq} = \frac{pq - \hat{p}\hat{q}}{pq} \approx \frac{\varepsilon_q}{q} + \frac{\varepsilon_p}{p} + 0 = R_q + R_p$$

Esto prueba que el error relativo del producto pq es aproximadamente la suma de los errores relativos de las aproximaciones $\hat{p}$ y $\hat{q}$ a los factores.

Algoritmos y Convergencia

Definición. Un algoritmo es un procedimiento que describe de forma precisa una sucesión de pasos que deben ser ejecutados en un orden especificado para resolver un problema o para obtener una aproximación a dicha solución.

Los algoritmos implementan los métodos numéricos para la resolución de problemas. Estos métodos pueden ser:

Iterativos si el método va construyendo una sucesión que en determinadas condiciones converge a la solución exacta del problema, es decir, si el algoritmo es re**iterativo**, en el sentido de que hay pasos de él que se repiten un número arbitrario de veces hasta que se cumpla cierto criterio de parada. En este tipo de métodos el error de redondeo no suele afectar a la solución obtenida tanto como en los métodos directos.

Directos si no son iterativos. En estos métodos los errores de redondeo suelen ser preponderantes sobre los de truncamiento.

En el caso de los algoritmos iterativos, se entiende que un método numérico converge si la sucesión formada por las aproximaciones obtenidas en cada iteración: $\{x_n\}_{n=1}^{\infty}$ converge, es decir, tiene como límite la solución exacta *x*. En términos más sencillos esto también se puede expresar diciendo que las aproximaciones obtenidas en cada iteración, x_n se van aproximando cada vez más al valor exacto solución del problema.

Cuanto menor sea el número de iteraciones necesarias para obtener la solución del problema con una tolerancia fijada de antemano, mayor será la velocidad de convergencia del método.

Es normal que los errores iniciales en los datos se propaguen a lo largo de una cadena de operaciones. Una cualidad deseable de cualquier proceso numérico es que un error pequeño en las condiciones iniciales produzca errores pequeños en el resultado final. Un algoritmo con esta cualidad se llama **estable**; en otro caso, se llama **inestable**. Siempre que sea posible, elegiremos métodos que sean estables.

Un algoritmo iterativo estable garantiza la convergencia. Un método numérico no siempre converge. Se dice que un método numérico iterativo diverge si los resultados obtenidos en cada iteración se van alejando cada vez más de la solución exacta. Por este motivo, al implementar un

método numérico mediante el correspondiente algoritmo suele ser una buena técnica que el criterio de parada contemple un número máximo de iteraciones a realizar.

Existen métodos numéricos de convergencia rápida pero inestables y otros estables pero de convergencia lenta.

La siguiente definición describe el fenómeno de la propagación de los errores.

Definición. Supongamos que ε representa un error inicial y que $\varepsilon(n)$ representa el crecimiento de dicho error después de n operaciones. Si se verifica que $|\varepsilon(n)| \approx n\varepsilon$, entonces se dice que el crecimiento es lineal. Si $|\varepsilon(n)| \approx K^n\varepsilon$, entonces se dice que el crecimiento es exponencial. Si $K>1$, entonces un error exponencial crece cuando $n\to\infty$ sin que podamos acotarlo; pero si $0<K<1$, entonces un error exponencial disminuye a cero cuando $n\to\infty$. Un error inicial puede propagarse de manera estable o inestable.

Ejemplo

Algoritmo de Horner para la evaluación de polinomios.

Consiste en una forma eficiente de evaluar un polinomio en la que se reduce el número de multiplicaciones con respecto a la forma tradicional. Sea el polinomio:

$$P(x)=a_0\cdot x^n+a_1\cdot x^{n-1}+\cdots+a_{n-1}\cdot x+a_n$$

Para obtener el valor de $P(x_0)$ se considera una expresión equivalente de *P(x):*

$$P(x)=\underbrace{\left(\cdots\underbrace{(a_0\cdot x+a_1)}_{b_1}\cdot x+\cdots+a_{n-1}\right)}_{b_{n-1}}\cdot x+a_n$$

De esta forma sustituyendo x por x_0, $P(x_0)=b_n$. Los pasos del algoritmo se pueden describir de la siguiente forma:

1º.- $b_0=a_0$.

2º.- Desde k = 1 a n

$$b_k=b_{k-1}\cdot x_0+a_k$$

3º.- $P(x_0)=b_n$

Ejercicios

1.— Sean: $f(x)=\dfrac{e^x-1-x}{x^2}$ y $P(x)=\dfrac{1}{2}+\dfrac{x}{6}+\dfrac{x^2}{24}$

Calcular $f(0.01)$ y $P(0.01)$ con 6 cifras significativas. Teniendo en cuenta que $P(x)$ es el polinomio de grado 2 de $f(x)$, ¿cuál de los dos resultados es más correcto?

Sol: f(0.01)=0.5, P(0.01)=0.501671; El 2º.

2.— Sea $P(x)=x^3-3x^2+3x-1$. Calcular $P(2.19)$ directamente y utilizando el algoritmo de Horner con 3 cifras significativas, y comparar con el valor exacto 1.685159.

Sol.: a)1.67; b)1.69

3.— Sean $p_1 = 1.414$ y $p_2 = 0.09125$, que están dados con 4 cifras significativas. Hallar el resultado más adecuado para $p_1 + p_2$ y $p_1 \cdot p_2$.

Sol.: Suma: 1.505; Producto: 0.1290

4.— Realizar los siguientes cálculos directamente indicando qué fenómeno se presenta. Obtener después un valor más preciso.

a) $\dfrac{\sin\left(\frac{\pi}{4}+0.00001\right)-\sin\left(\frac{\pi}{4}\right)}{0.00001}$ b) $\dfrac{\ln(2+0.00005)-\ln(2)}{0.00005}$

Sol: a) a1)0.7; a2)0.707103; b)b1)0.5; b2)0.499994

5.— La pérdida de cifras significativas se puede evitar reordenando los cálculos. Hallar en los siguientes casos una forma equivalente que evite la pérdida de cifras significativas para valores grandes de x:

a) $\ln(x+1)-\ln(x)$ b) $\sqrt{x^2+1}-x$

6.— Sean $P(x)=x^3-3x^2+3x-1$; $Q(x)=((x-3)x+3)x-1$; $R(x)=(x-1)^3$

Calcular con redondeo a 4 cifras significativas:

a) $P(2.72)$; $Q(2.72)$; $R(2.72)$

b) $P(0.975)$; $Q(0.975)$; $R(0.975)$

Sol.:a)P(2.72)=5.08;Q(2.72)=5.087;R(2.72)=5.088;

b)P(0.975)=0.00035; Q(0.975)=0.00035; R(0.975)=-0.1562 10^{-4}

7.— Justificar que para evitar el efecto de cancelación en la resolución de la ecuación de segundo grado se pueden utilizar las expresiones:

$$x_1 = \frac{-2c}{b+\sqrt{b^2-4ac}} \quad \text{y} \quad x_2 = \frac{-2c}{b-\sqrt{b^2-4ac}}$$

¿Qué expresiones habría que utilizar para x_1 y x_2 si $b>0$? ¿Y si $b<0$?

8.— Utilizar las expresiones más adecuadas para resolver la ecuación: $x^2-1000.001x+1=0$