75.12/95.04 ANÁLISIS NUMÉRICO I
95.10 MODELACIÓN NUMÉRICA
95.13 MÉTODOS MATEMÁTICOS Y NUMÉRICOS

ERRORES Y REPRESENTACIÓN NUMÉRICA

Ing. Rodolfo A. Schwarz

Años 2020/2021

Indice

Indice

II

Conceptos

Principales conceptos asociados al error.

- Asociado al concepto de aproximación.
- No necesariamente tiene que ver con una equivocación o una falla en los cálculos o procedimientos.
- También está asociado a la incertidumbre, principalmente en el caso de los resultados de un modelo numérico.
- Los datos y los modelos numéricos siempre tienen errores.
- Las operaciones como suma, resta, multiplicación y división, hechas con ayuda de computadoras, tienen errores.

Fuentes de error

- **Error inherente**: error de los datos de entrada.
- **Error de redondeo**: error asociado a la representación numérica.
- **Error de truncamiento/discretización**: error debido al uso de procedimientos (algoritmos) discretos y/o finitos.
- **Error del modelo matemático**: error en la formulación matemática del modelo. Fuera del alcance del Análisis Numérico.
- **Error humano (incluye el error de máquina)**: errores causados por la intervención humana, como la carga de datos, desarrollo de un programa, fallas en el diseño del procesador, etc. Ídem anterior.

Error inherente

- **Error absoluto**: diferencia entre el valor «exacto» (m) y el aproximado ($\tilde{m}$).

$$E_A = \tilde{m} - m$$

- **Error relativo**: relación entre el error absoluto y el valor «exacto».

$$e_r = \frac{\tilde{m} - m}{m} = \frac{E_A}{m}$$

- El error relativo es una mejor «aproximación» que el error absoluto.

Error inherente

- En realidad, interesa:
  1. Cota del error absoluto (módulo):

$$|E_A| = |\tilde{m} - m|$$

  2. Cota del error relativo (módulo):

$$|e_r| = \left|\frac{\tilde{m} - m}{m}\right| = \frac{|\tilde{m} - m|}{|m|} = \frac{|E_A|}{|m|}$$

- Normalmente no conocemos m, por lo tanto:

$$|e_r| = \frac{|E_A|}{|\tilde{m} \pm E_A|} \approx \frac{|E_A|}{|\tilde{m}|}$$

Error inherente

Condición: asociada al error inherente.
Supongamos la siguiente función:

$$y = f(x)$$

En realidad tenemos una aproximación:

$$\tilde{y} = f(x + \Delta x)$$

El error absoluto de esta aproximación está dado por:

$$E_y = \tilde{y} - y$$

Error inherente

- Si aproximamos por Taylor queda:

$$E_y = \tilde{y} - y = f(x + \Delta x) - f(x) = f'(x)\Delta x + \frac{f''(x)}{2!}\Delta x^2 + \dots$$
$$E_y = f'(x)\Delta x + \frac{f''(\xi)}{2!}\Delta x^2$$

- El error relativo es:

$$e_y = \frac{E_y}{y} = \frac{\tilde{y} - y}{y} = \frac{f'(x)\ \Delta x}{f(x)} + O(\Delta x^2)$$
$$e_y \approx \frac{f'(x)\ x}{f(x)}\frac{\Delta x}{x} = \frac{f'(x)\ x}{f(x)}e_x$$

Error inherente

- Al coeficiente que multiplica al error relativo de x (e_x) lo llamaremos *Número de Condición*:

$$C_p = \frac{f'(x)\,x}{f(x)}$$

- Si la función es de dos o más variables, entonces tendremos varios C_p:

$$C_{p_i} = \frac{\partial f(\bar{x})}{\partial x_i} \frac{x_i}{f(\bar{x})}$$

Error inherente

- Si suponemos que:

$$e_{rx_i} = \left| \frac{\Delta x_i}{x_i} \right| = \frac{|\Delta x_i|}{|x_i|} \leq r$$

- Tendremos la siguiente fórmula para el Número de Condición:

$$C_p = \sum_i \left| \frac{\partial f(x)}{\partial x_i} \right| \frac{|x_i|}{|f(x)|}$$

Error inherente

El coeficiente C_p:

- Nos da una idea de la sensibilidad del problema a pequeños cambios en los datos (Δx).
- Nos muestra cómo se propagan los errores inherentes.
- Si $C_p >> 1$, se dice que el problema está **mal condicionado**.
- La condición suele ser **inherente** al modelo matemático.

Error de redondeo

- Está asociado a la representación numérica.
- Es fundamental al trabajar con calculadoras y computadoras.
- Ejemplo: Representar el número $\frac{4}{3}$ en el sistema decimal:

$$\frac{4}{3} = 1{,}33333\ldots = 1{,}\hat{3}$$

- En una calculadora o computadora esto es impracticable.

Error de redondeo

- Si usamos solamente tres dígitos para representar dicho número, tenemos:

$$\frac{4}{3} = 1{,}33$$

- Podemos escribirlo así:

$$\frac{4}{3} \approx 1 \cdot 10^0 + \frac{3}{10} + \frac{3}{100} = 1{,}33 \cdot 10^0$$

Error de redondeo

- Otro ejemplo es $\frac{1}{7}$:

$$\frac{1}{7} = 0{,}142857142857\ldots$$

- Si tomamos solamente tres dígitos nos queda:

$$\frac{1}{7} \approx 0{,}142 \quad \text{o} \quad \frac{1}{7} \approx 0{,}143$$

$$\frac{1}{7} \approx 1{,}42 \cdot 10^{-1} \quad \text{o} \quad \frac{1}{7} \approx 1{,}43 \cdot 10^{-1}$$

- Hay dos formas de representarlo. ¿Cual elegimos?

Representación numérica

- Un número se puede representar con la siguiente notación:

$$N \approx d,d_1 d_2 d_3 \ldots d_t \cdot \beta^p$$

- La representación numérica queda definida por t, p y β.
- Se conoce como representación por *punto o coma flotante*.
- Las calculadoras y las computadoras suelen usar $\beta = 2$ (sistema binario).

Representación numérica

- Podemos decir que 1/7 se puede representar así:

$$N \approx d{,}d_1 d_2 d_3 d_4 \dots d_k \cdot 10^p$$

- Si usamos solamente tres dígitos despreciamos:

$$d_3 d_4 \dots d_k$$

- Pues la representación elegida es:

$$N \approx d{,}d_1 d_2 \dots d_k \cdot 10^p$$

Representación numérica

Dos casos:
El primero es:

$$\frac{1}{7} \approx 1{,}42 \cdot 10^{-1}$$

Se conoce como *corte* (en algunos libros también como truncado o truncamiento).

Simplemente «borra» los restante dígitos. No los toma en cuenta.

Representación numérica

El segundo es:

$$\frac{1}{7} \approx 1{,}43 \cdot 10^{-1}$$

Se conoce como *redondeo*.

El último dígito (d_2) se modifica según el siguiente criterio:

$$d_2 = \begin{cases} d_2 & \text{si } 0 \leq d_3 \leq 4 \\ d_2 + 1 & \text{si } 5 \leq d_3 \leq 9 \end{cases}$$

Representación numérica

- Podemos definir nuestra unidad de redondeo como:

$$\mu = \begin{cases} 10^{-2} & \text{para corte,} \\ 0{,}5 \cdot 10^{-2} & \text{para redondeo.} \end{cases}$$

- Si aplicamos esto en forma genérica tenemos:

$$\mu = \begin{cases} 10^{-t} & \text{para corte,} \\ 0{,}5 \cdot 10^{-t} & \text{para redondeo.} \end{cases}$$

Representación numérica

- Notación de punto flotante:

$$0, d_1 d_2 d_3 \ldots d_t$$

- La unidad de redondeo es:

$$\mu = \begin{cases} 10^{-t} & \text{para corte,} \\ 0{,}5 \cdot 10^{-t} & \text{para redondeo.} \end{cases}$$

Representación numérica

- La unidad de redondeo se conoce también como *unidad de máquina* o *precisión*.
- Por ejemplo, en las computadores personales tenemos al menos dos tipos:
  1. Simple precisión: 10^{-7} o 10^{-8}.
  2. Doble precisión: 10^{-15} o 10^{-16}.
- Las computadoras comunes suelen utilizan una precisión denominada «real» que está dada por una precisión de 10^{-12} o 10^{-13}.

Representación numérica

Estabilidad: asociada al error de redondeo y al algoritmo.

Ejemplo

Supongamos la siguiente operación usando la representación con tres dígitos:

$$z = \frac{4}{3} + \frac{1}{7}$$

Al redondear el resultado es:

$$\tilde{z} = 1{,}33 + 1{,}43 \cdot 10^{-1} = 1{,}33 + 0{,}143 = 1{,}47$$

Sin ninguna limitación, tendríamos:

$$\tilde{z} = 1{,}3333333\ldots + 0{,}1428571\ldots = 1{,}4761904\ldots$$

Representación numérica

Si redondeamos este último resultado obtenemos:

$$\tilde{z} = 1{,}48.$$

Al analizar el resultado aproximado, tenemos:

- El error absoluto está dado por:

$$|\tilde{z} - z| = |1{,}47 - 1{,}4761904\ldots \approx |0{,}0061| = 6{,}1 \cdot 10^{-3}$$

- El error relativo es:

$$\frac{|\tilde{z} - z|}{|z|} \approx \frac{6{,}1 \cdot 10^{-3}}{1{,}4761904\ldots} \approx 0{,}00419$$

- Muy parecido la unidad de redondeo: $\mu = 0{,}5 \cdot 10^{-2}$

Representación numérica

- Podemos decir que por el error de redondeo el error relativo es:

$$e_{rz} = \mu.$$

- Por lo tanto, para k operaciones podríamos decir que el error relativo es:

$$e_{rz} = \sum_{i=1}^{k} \mu_i \approx |k|\mu$$

- En general, podemos expresarlo como:

$$e_{rz} = \sum_{i=1}^{k} T(x)\mu_i = T_e\, \mu$$

Representación numérica

Finalmente

- Coeficiente T_e: **Término de Estabilidad**.
- Representa: **Influencia del error de redondeo**.
- Si $T_e >> 1$ se considera que el algoritmo es *inestable*.

Error total

- Si sumamos los errores relativos debidos a los errores inherentes y de redondeo tenemos:

$$e_{r_T} = C_p\, r + T_e\, \mu.$$

- Podemos decir que un buen algoritmo es aquel que cumple lo siguiente:

$$r \approx \mu \Rightarrow e_{r_T} = C_p\, r + T_e\, \mu = (C_p + T_e)\, \mu = C_p \left(1 + \frac{T_e}{C_p}\right) \mu$$

$$1 + \frac{T_e}{C_p} = \frac{C_p + T_e}{C_p} > / > 1$$

.

Error de truncamiento/discretización

- Otra fuente de error es el **error de truncamiento y/o discretización**.
- Está dado por el uso de modelos numéricos discretos para aproximar modelos matemáticos continuos.
- Tomemos el caso de una ecuación diferencial ordinaria con valores iniciales:

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(t, y) \quad \text{con} \quad a \leq t \leq b \quad \text{y} \quad y(a) = \alpha.$$

Error de truncamiento/discretización

- Un algoritmo (sencillo) para resolver esta ecuación es el siguiente:

$$y_{i+1} = y_i + h\, f(t_i, y_i),$$

algoritmo basado en la aproximación discreta de la derivada:

$$\left.\frac{\mathrm{d}y}{\mathrm{d}t}\right|_{t_i} = f(t_i, y_i) = \frac{y(t_i + h) - y(t_i)}{h} = \frac{y_{i+1} - y_i}{h}.$$

- Se trata de un algoritmo discreto que aproxima un modelo continuo.

Error de truncamiento/discretización

- El truncamiento está asociado al uso de series infinitas.
- Tenemos el caso de la serie para obtener el número e:

$$e = \sum_{j=0}^{\infty} \frac{x^j}{j!}.$$

- Es evidente que no podemos ejecutar con una computadora una sumatoria con infinitos términos, por lo que la serie se modifica así:

$$e = \sum_{j=0}^{n} \frac{x^j}{j!}.$$

donde n es un valor determinado.

Error de truncamiento/discretización

- En consecuencia, la aproximación del valor de e depende de la cantidad de términos que integran la sumatoria, es decir, el valor de n.
- El error de la aproximación está dado por los términos despreciados ($j \geq n+1$).
- Veremos más adelante que el primer término despreciado es el que «mejor» representa el error de la aproximación.
- Eso nos permite identificar la «calidad» del algoritmo y su capacidad para aproximar el valor buscado.

Ejemplos

Problema mal condicionado

- Supongamos la siguiente función:

$$y(x) = \ln(x).$$

- Calculemos el C_p:

$$C_p = \frac{\partial y(x)}{\partial x}\frac{x}{y(x)} = \frac{1}{x}\frac{x}{\ln x} = \frac{1}{\ln x}$$

- Cuando $x \approx 1$, entonces:

$$C_p = \frac{1}{\ln 1} \to \infty$$

Ejemplos

Algoritmo inestable

- Supongamos la siguiente ecuación:

$$y_n = \int_0^1 \frac{x^n}{x+10}\, dx.$$

- Obtener el y_1 es sencillo:

$$y_1 = \int_0^1 \frac{x}{x+10}\, dx = x\big|_0^1 - 10\ \ln(x+10)\big|_0^1$$
$$y_1 = 1 - 10\ \ln(1{,}1) = 0{,}0468982019570$$

Ejemplos

Algoritmo inestable

Obtener otros valores puede no ser tan sencillo:

- El y_5:

$$y_5 = \int_0^1 \frac{x^5}{x+10}\, dx;$$

- El y_{20}:

$$y_{20} = \int_0^1 \frac{x^{20}}{x+10}\, dx;$$

- O el y_{34}:

$$y_{34} = \int_0^1 \frac{x^{34}}{x+10}\, dx.$$

Ejemplos

Algoritmo inestable

Propongamos el siguiente algoritmo:

$$y_n + 10\,y_{n-1} = \int_0^1 \frac{x^n + 10\,x^{n-1}}{x+10}\mathrm{d}x$$

$$y_n + 10\,y_{n-1} = \int_0^1 \frac{x+10}{x+10}x^{n-1}\mathrm{d}x = \int_0^1 x^{n-1}\mathrm{d}x$$

$$y_n + 10\,y_{n-1} = \left.\frac{x^n}{n}\right|_0^1 = \frac{1}{n}(1^n - 0) = \frac{1}{n}$$

$$y_n = \frac{1}{n} - 10\,y_{n-1}$$

Ejemplos

Algoritmo inestable

Con el algoritmo obtenido calculemos:

$$y_1 = \frac{1}{1} - 10\, y_0$$

Necesitamos el y_0, que lo obtenemos así:

$$y_0 = \int_0^1 \frac{x^0}{x+10}\mathrm{d}x = \ln(x+10)\big|_0^1 = \ln\left(\frac{10}{11}\right) = 0{,}0953101798043$$

Finalmente, el y_1 resulta ser:

$$y_1 = \frac{1}{1} - 10 \cdot 0{,}0953101798043 = 0{,}0468982019570$$

Ejemplos

Algoritmo inestable

Con el algoritmo obtenido calculemos otros valores de y_n:

1. Para $n = 5$:
$$y_5 = 0{,}0153529008398454 74$$
2. Para $n = 12$:
$$y_{12} = 0{,}00711381076779595$$
3. Para $n = 20$:
$$y_{20} = 7483{,}468021084803$$
4. Analicemos este último resultado.

Ejemplos

Algoritmo inestable

Para eso representemos las curvas de algunas funciones:

- Con $n = 5$, $f_5(x)$:

$$f_5(x) = \frac{x^5}{x + 10};$$

- Con $n = 10$, $f_{10}(x)$:

$$f_{10}(x) = \frac{x^{10}}{x + 10};$$

Ejemplos

Algoritmo inestable

- Con $n = 15$, $f_{15}(x)$:

$$f_{15}(x) = \frac{x^{15}}{x + 10};$$

- Con $n = 20$, $f_{20}(x)$:

$$f_{20}(x) = \frac{x^{20}}{x + 10}.$$

Ejemplos

Algoritmo inestable

Veamos las curvas de las funciones a integrar para algunos n:

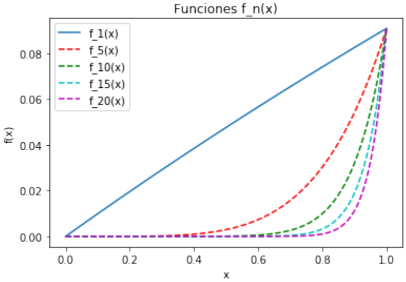


Figura: Curvas de las funciones a integrar.

Ejemplos

Algoritmo inestable

- El gráfico anterior nos muestra que los valores y_n son las áreas bajo las curvas $f_n(x)$ entre .
- El valor y_1 es mayor que y_5, puesto que el área bajo $f_1(x)$ es mayor que el área bajo $f_5(x)$, entre 0 y 1.
- Por esa razón, es fácil advertir que siempre se debe cumplir que $y_n < y_{n-1}$.

Ejemplos

Algoritmo inestable

Como conclusión de lo anterior

- El resultado obtenido no es correcto. El valor correcto es: $y_{20} = 0{,}004347035818028110 0$.
- El valor obtenido con el algoritmo desarrollado es: $y_{20} = 7483{,}468021084803$.
- La diferencia es muy grande. La causa de esta diferencia tan notable es que a la computadora le «cuesta» procesar la operación $\frac{1}{n} - 10\, y_{n-1}$, a medida que n aumenta, pues $y_n < y_{n-1}$ y $\frac{1}{n} < \frac{1}{n-1}$ pero no disminuyen de la misma forma. Como los valores y_{n-1} y y_n son muy pequeños, la computadora no los puede representar correctamente.

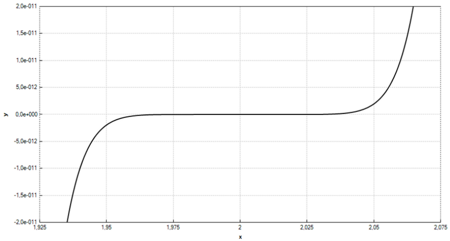
Ejemplos

Veamos el caso de algoritmos inestables.

Algoritmos inestables

Tomemos la siguiente función polinómica:

$$P_1(x) = (x - 2)^9.$$

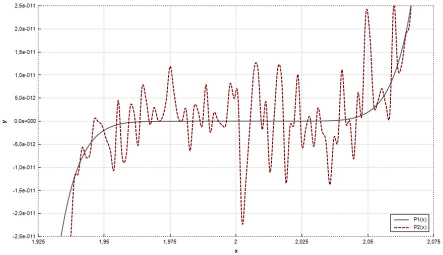


Ejemplos

Algoritmos inestables

Desarrollemos la función polinómica:

$$P_2(x) = x^9 - 18x^8 + 144x^7 - 672x^6 + 2016x^5 - 4032x^4 + 5376x^3 - 4608x^2 + 2304x - 512.$$

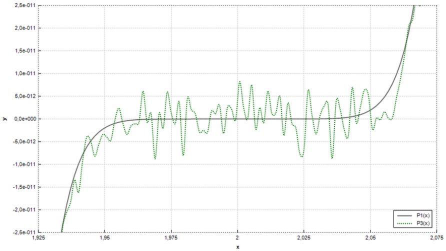


Ejemplos

Algoritmos inestables

Otro algoritmo posible:

$$P_3(x) = \{\{\{\{\{\{[(x-18)x+144]x-672\}x+2016\}x-4032\}x+5376\}x-4608\}x+ \\ +2304\}x-512.$$

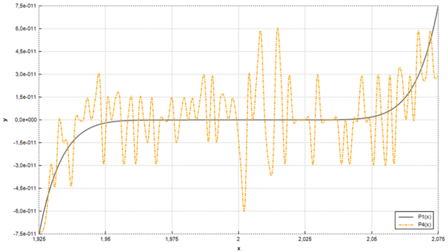


Ejemplos

Algoritmos inestables

Agrupando términos obtenemoos otro algoritmo:

$$P_4(x) = (x^8+144x^6+2016x^4+5376x^2+2304)x-(9x^8+336x^6+2016x^4+2304x^2+256)2.$$

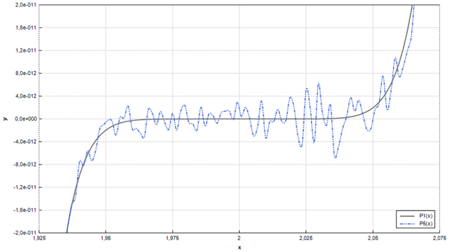


Ejemplos

Algoritmos inestables

Agrupando términos de otra forma obtenemoos un último algoritmo:

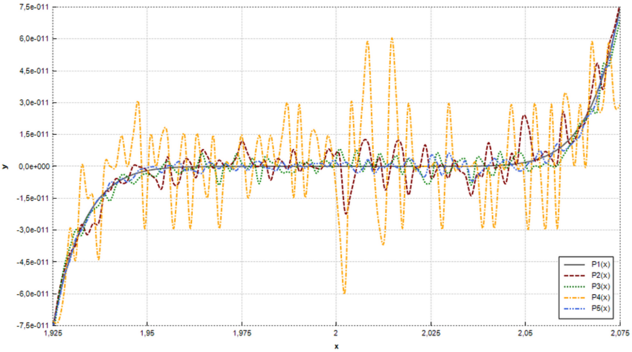
$$P_5(x) = x^9 - 512 - (x^7 - 128)18x + (x^5 - 32)144x^2 - (x^3 - 8)672x^3 + (x - 2)2016x^4.$$



Ejemplos

Algoritmos inestables

Comparamos todas las representaciones gráficas.



Gráfica de proceso

- Determinar analíticamente los coeficientes C_p y T_e puede ser muy engorroso, dependiendo de las características del modelo matemático y del algoritmo.
- Una forma de obtenerlos es la llamada Gráfica de Proceso, que no es otra cosa que un diagrama de flujo que representa la propagación de los errores relativos y de redondeo. Este método ayuda a ordenar el proceso.
- Dos ejemplos:
  1. Suma y/o resta;
  2. Producto y/o cociente.

Gráfica de proceso

Gráfica de proceso de $z = x \pm y$.

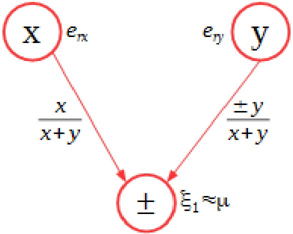


Figura: Gráfica de proceso de la suma y/o resta.

Desarrollemos la propagación de los errores inherentes e_{rx} y e_{ry} para hallar e_{rz}.

Gráfica de proceso

Desarrollo de la propagación del error para obtener e_{rz}.

$$e_{rz} = \frac{|x|}{|x \pm y|} e_{rx} + \frac{|\pm y|}{|x \pm y|} e_{ry} + \xi_1 = \frac{|x|}{|x \pm y|} e_{rx} + \frac{|\pm y|}{|x \pm y|} e_{ry} + \mu$$

$$C_{px} = \frac{|x|}{|x \pm y|}, \quad C_{py} = \frac{|\pm y|}{|x \pm y|} = \frac{|y|}{|x \pm y|}, \quad T_e = 1$$

Si $e_{rx}; e_{ry} \leq r$, entonces

$$C_p = \frac{|x| + |y|}{|x \pm y|}, \quad T_e = 1$$

Gráfica de proceso

Gráficas de proceso de $z = x \cdot y$ y $z = \frac{x}{y}$.

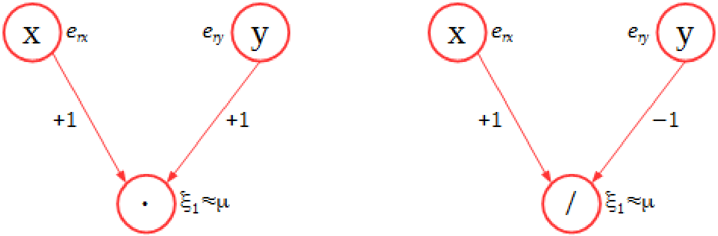


Figura: Gráfica de proceso de la multiplicación y división.

Desarrollemos la propagación de los errores inherentes e_{rx} y e_{ry} para hallar e_{rz}.

Gráfica de proceso

Desarrollo de la propagación del error para obtener e_{rz}.

$$e_{rz} = |1|e_{rx} + |\pm 1|e_{ry} + \xi_1 = |1|e_{rx} + |\pm 1|e_{ry} + \mu$$

$$C_{px} = |1|, \quad C_{py} = |\pm 1|, \quad T_e = 1$$

Si $e_{rx}; e_{ry} \leq r$, entonces

$$C_p = |1| + |\pm 1| = 2, \quad T_e = 1$$

Gráfica de proceso

Ejemplo

Tomemos el caso de algoritmo ya analizado con $n = 1; 2; 3$.

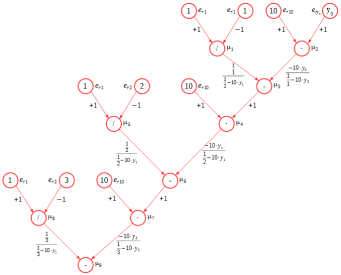


Figura: Gráfica de proceso del algoritmo $y_n = \frac{1}{n} - 10 \cdot y_{n-1}$.

Perturbaciones experimentales

- El ejemplo anterior sirve para darnos cuenta que la gráfica de proceso no siempre es útil y práctica.
- Si debemos analizar un algoritmo con miles de pasos, evidentemente no nos alcanza el papel para dibujar la gráfica.
- Y si nuestro algoritmo tiene una cantidad de pasos no determinada previamente, como muchos modelos numéricos iterativos, entonces no hay forma de obtener la gráfica de proceso.
- ¿Cómo podemos estimar el Cp y el Te?

Perturbaciones experimentales

Número de condición

- Partamos de la definición de cómo obtener el error relativo total de una operación.

$$e_{rT} = C_p \cdot r + T_e \cdot \mu.$$

- Supongamos por un momento despreciables los errores por redondeo, es decir:

$$T_e \approx 0.$$

- Entonces, el error relativo total queda:

$$e_{rT} = C_p \cdot r \Rightarrow C_p = \frac{e_{rT}}{r}.$$

Perturbaciones experimentales

Número de condición

Estimación del C_p:

- Calcular varios valores, introduciendo errores en los datos de entrada.
- Tomar como real al valor calculado sin errores.
- Obtener el error relativo de cada uno de los otros.
- Resultado: varios coeficientes C_p.
- ¿Qué debería pasar?

Perturbaciones experimentales

Número de condición

Ejemplo

Analicemos si el siguiente algoritmo está mal condicionado:

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

Al calcular

$$y\left(\frac{\pi}{4}\right) = y(0{,}785398)$$

obtenemos

$$y(0{,}785398) = 0{,}707107$$

Perturbaciones experimentales

Número de condición

Ejemplo

Perturbemos levemente el valor de x y calculemos nuevamente la función:

$$y_1(0{,}784613) = 0{,}706551 \quad \text{con} \quad e_{rx_1} = -0{,}001$$

Con otra perturbación:

$$y_2(0{,}786184) = 0{,}707662 \quad \text{con} \quad e_{rx_2} = 0{,}001$$

Y finalmente con una última perturbación:

$$y_3(0{,}788540) = 0{,}709325 \quad \text{con} \quad e_{rx_3} = 0{,}004$$

Perturbaciones experimentales

Número de condición

Ejemplo

Calculemos ahora los C_p de cada perturbación tomando y_0 como base de comparación:

$$C_{p_1} = \frac{y_1 - y_0}{y_0 \cdot e_{rx_1}} = \frac{0{,}706551 - 0{,}707107}{0{,}707107 \cdot (-0{,}001)} = 0{,}785707$$

$$C_{p_2} = \frac{y_2 - y_0}{y_0 \cdot e_{rx_2}} = \frac{0{,}707662 - 0{,}707107}{0{,}707107 \cdot 0{,}001} = 0{,}785090$$

$$C_{p_3} = \frac{y_3 - y_0}{y_0 \cdot e_{rx_3}} = \frac{0{,}709325 - 0{,}707107}{0{,}707107 \cdot 0{,}004} = 0{,}784163$$

Podemos ver que los valores de los diferentes C_p son muy parecidos y muy cercanos a 1. Por ello la función está *bien condicionada*.

Perturbaciones experimentales

Término de estabilidad

- Para obtener el T_e, partamos de la misma expresión:

$$e_{rT} = C_p \cdot r + T_e \cdot \mu$$

- Supongamos $r \approx 0$, entonces:

$$e_{r\mu} = \frac{y_\mu - y}{y} = T_e \cdot \mu$$

- Para otra representación numérica tendremos:

$$e_{r\xi} = \frac{y_\xi - y}{y} = T_e \cdot \xi$$

Perturbaciones experimentales

Término de estabilidad

- Si hacemos:

$$e_{r\mu} - e_{r\xi} = \frac{y_\mu - y}{y} - \frac{y_\xi - y}{y} = \frac{y_\mu - y_\xi}{y} = T_e \cdot (\mu - \xi)$$

- Supongamos también que:

$$\mu >> \xi \to y_\xi \approx y$$

- Entonces:

$$T_e = \frac{y_\mu - y_\xi}{y_\xi \cdot (\mu - \xi)}$$

Perturbaciones experimentales

Término de estabilidad

Ejemplo

Analicemos la estabilidad del algoritmo visto antes:

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

Con la representación numérica $\mu = 10^{-4}$, $x_\mu = 0{,}785400$, calculemos y_μ:

$$y_\mu = y(x_\mu = 0{,}785400) = 0{,}707100.$$

Ahora con $\xi = 10^{-12}$, $x_\xi = 0{,}785398163397$, calculemos y_ξ:

$$y_\xi = y(x_\xi = 0{,}785398163397) = 0{,}707106782937.$$

Perturbaciones experimentales

Término de estabilidad

Ejemplo

Con valores calculados y las representaciones numéricas aplicadas, obtengamos una aproximación del término de estabilidad:

$$T_e = \frac{y_\mu - y_\xi}{y_\xi(\mu - \xi)} = \frac{0{,}707100 - 0{,}707106782937}{0{,}707106782937\,(10^{-4} - 10^{-12})} = -0{,}0959$$

Si tomamos el valor absoluto tenemos que:

$$|\,T_e| = 0{,}0959 << 1,$$

por lo tanto, podemos afirmar que el algoritmo es estable.

Exactitud y precisión

Son dos términos que se usan en forma indistinta pero en rigor no los son.

- **Exactitud (*accuracy*)**: se refiere al error absoluto o relativo de un determinado valor o cantidad.
- **Precisión (*precision*)**: se refiere a la exactitud con que se llevan a cabo las operaciones aritméticas básicas.
- *La Exactitud no está limitada por la Precisión*.

Cancelación

- Se da cuando se restan dos cantidades similares.
- Caso típico: solución de la ecuación de segundo grado:

$$a\,x^2 + b\,x + c = 0$$

$$x_{1;2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2\,a}$$

Si $\sqrt{b^2 - 4\,a\,c} \approx |b|$, queda:

$$x_{1;2} = \frac{-b \pm |b|}{2a} \Rightarrow \begin{cases} x_1 \approx -\frac{b}{a} \\ x_2 \approx 0 \end{cases}$$

Cancelación

- Podemos evitar el último resultado no correcto haciendo

$$x_{1;2} = \frac{-b \pm |b|}{2a} \Rightarrow \begin{cases} x_1 = -\frac{b}{a} \\ x_2 = \frac{c}{x_1\ a} \end{cases}$$

- En cambio, si $b^2 \approx 4ac$ no hay solución algebraica o modificación del algoritmo para evitar el problema de que $\sqrt{b^2 - 4ac} \approx 0$ y el resultado sea una raíz doble ($x_{1;2} \approx -\frac{b}{2a}$). La única solución es utilizar una unidad de máquina menor que pueda representar la diferencia.

Cancelación

- No siempre es perjudicial. Por ejemplo, si tenemos:

$$z = y + (x - t)$$

- si $y >> x; t$ siempre tendremos que:

$$z \approx y,$$

- Por lo que la resta $(x - t)$ no influye en el resultado final.

Precisión y aproximación

- Una mayor precisión no siempre mejora la aproximación.
- Tomemos el caso de aproximar la derivada de la siguiente función:

$$f(x) = \text{sen}(2\pi x)$$

- Conocemos la derivada «analítica»:

$$f'(x) = 2\pi \cos(2\pi x)$$

- Para $x = 0{,}45$ tenemos:

$$f'(0{,}45) = 2\pi \cos(2\pi 0{,}45) = -5{,}975664\ldots.$$

Precisión y aproximación

- Tomemos esta aproximación de la derivada:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

- Para $x = 0{,}45$ y $h = 0{,}1$ tenemos la siguiente aproximación:

$$f'(x) \approx \frac{f(0{,}45 + 0{,}1) - f(0{,}45)}{0{,}1} = -6{,}180340$$

- Para $x = 0{,}45$ y $h = 0{,}01$ tenemos la siguiente aproximación:

$$f'(0{,}45) \approx \frac{f(0{,}45 + 0{,}01) - f(0{,}45)}{0{,}01} = -6{,}032711$$

- A continuación, una tabla con aproximaciones de la derivada para $x = 0{,}45$ y diferentes valores de h.

Precisión y aproximación

Tabla con aproximaciones de la derivada para $x = 0{,}45$.

h	f(x+h)	f(x)	f'(x)	$e_P(x)$
1,0E-01	-0,3090169943749480	0,3090169943749480	-6,1803398874989500	2,04676E-01
1,0E-02	0,2486898871648550	0,3090169943749480	-6,0327107210092700	5,70464E-02
1,0E-03	0,3030352696327740	0,3090169943749480	-5,9817247421734500	6,06041E-03
...	...	...	...	...
1,0E-06	0,3090110187045180	0,3090169943749480	-5,9756704291480400	6,09966E-06
1,0E-07	0,3090163968084530	0,3090169943749480	-5,9756649417597200	6,12277E-07
1,0E-08	0,3090169346183040	0,3090169943749480	-5,9756643855379800	5,60549E-08
1,0E-09	0,3090169883992830	0,3090169943749480	-5,9756645187647400	1,89282E-07
1,0E-10	0,3090169937773810	0,3090169943749480	-5,9756660730769800	1,74359E-06
...	...	...	...	...
1,0E-13	0,3090169943743500	0,3090169943749480	-5,9763305415572200	6,66212E-04
1,0E-14	0,3090169943748880	0,3090169943749480	-5,9952043329758400	1,95400E-02
1,0E-15	0,3090169943749420	0,3090169943749480	-5,8841820305133300	9,14823E-02
h	f(x+h)	f(x)	f'(x)	$e_P(x)$

Precisión y aproximación

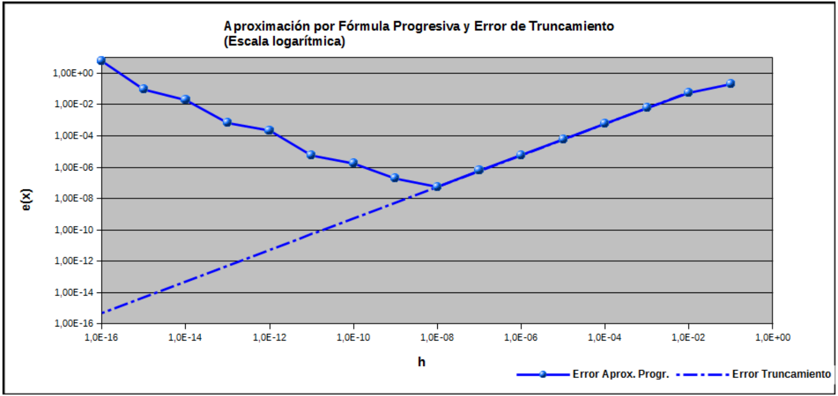


Figura: Errores de las aproximaciones.

Precisión y aproximación

Explicación del gráfico anterior

- La curva con línea continua y puntos representa el error «real» de la aproximación;
- La curva con línea no continua representa el error «teórico» dado por el error de truncamiento (modelo numérico discreto/modelo matemático continuo);
- En el tramo $h = 10^{-8}$–$h = 10^{-1}$, los valores prácticamente coinciden, no así en el tramo $h = 10^{-16}$–$h = 10^{-8}$, donde el error «teórico» (de truncamiento) es notablemente superado por el error «real».
- Un algoritmo puede ser estable en algunas circunstancias e inestable en otras.

Precisión y aproximación

Conclusiones a partir del gráfico:

- Disminuir el h mejora la aproximación hasta un cierto punto.
- La mejor aproximación se obtiene con $h = 10^{-8}$.
- Cuando $h < 10^{-8}$, la aproximación empeora; particularmente cuando $h = 10^{-16}$, la situación es peor que para $h = 10^{-1}$.
- Por lo tanto no siempre trabajar con más precisión en los cálculos (más decimales en este caso) es garantía de obtener mayor «exactitud».

Conclusiones

Algunas reglas para obtener algoritmos estables:

- Evitar la resta de cantidades con errores.
- Minimizar el tamaño de las cantidades intermedias relativas al resultado final.
- Es más ventajoso escribir una expresión que actualice el resultado, como por ejemplo:

$$\text{valor}_{\text{nuevo}} = \text{valor}_{\text{viejo}} + \text{corrección}_{\text{pequeña}}$$

- Usar transformaciones bien condicionadas.

Bibliografía

- González, H. *Análisis numérico. Primer curso.* Nueva Librería. 2002.
- Higham, N.J. *Accuracy and stability of numerical algorithms.* SIAM. 1996.
- Burden, R. L., Faires, J. D. & Burden, A. M. *Análisis Numérico*. Décima Edición, CENGAGE Learning, 2016.
- Samarski, A. A. *Introducción a los métodos numéricos*. Editorial Mir, 1986.
- Gavurin, M.K. *Conferencia sobre los métodos de cálculo.* Mir. 1973.

¡MUCHAS GRACIAS POR LA ATENCIÓN!