

# Introducción a errores

1. Motivación del estudio del tema
2. Introducción a la aritmética de punto flotante
  - a. Definición de base, mantisa, exponente
  - b. Definición de error relativo y absoluto, emach
3. Reglas de propagación de errores
4. Números de condición para problemas y algoritmos
5. Algunas técnicas usadas en calculo numérico
  - a. Noción de iteración
  - b. Linealización local
  - c. Extrapolación de Richardson

# Errores

## Introducción

En el curso se verán métodos para hallar la solución numérica de ecuaciones diferenciales.

Como criterio general de trabajo, el objetivo será llegar a la solución **correcta** (esto es: con un nivel de error aceptable) en forma **eficiente** (en un tiempo razonable usando los recursos disponibles).

Lo último nos lleva naturalmente al estudio del número de operaciones necesarias en los algoritmos utilizados, al menos una estimación de su orden de magnitud, mientras que lo primero nos conduce naturalmente al estudio del origen y posterior propagación de los errores de cálculo.

En la aproximación de un fenómeno físico mediante un modelo matemático y su posterior solución numérica, se cometen diferentes clases de errores.

Ellos son de distintos tipos, y pueden estar:

- en el modelo usado
- en los datos
- en la aproximación matemática del modelo
- en las cuentas

Sobre el último tipo de errores se va a centrar la atención ahora, observando hechos comunes que pueden ocurrir. Para entender esto mejor, se debe primero tener una visión clara de la aritmética usual de las computadoras digitales, y de cómo se representan los números reales en ellas.

## Representación de punto flotante

Cualquiera sea la máquina usada esta no trabaja con todos los números reales, sino sólo con un conjunto finito de números racionales, que típicamente es de la forma:

$$FP = \{ \sigma \times 0.d_1 d_2 \dots d_t \times \beta^e \mid d_i \in N \text{ con } 0 \leq d_i < \beta, d_1 \neq 0, e \in Z, L \leq e \leq U \}$$

Y al que se designa por su sigla en inglés FP (floating point)

$\sigma \rightarrow$  signo

$t \rightarrow$  Cantidad de cifras en la mantisa

$\beta \rightarrow$  Base (habitualmente es 2)

$e \rightarrow$  Exponente

Notaciones:

$$1) \quad FP_i = \{x \in FP : \text{exponente}(x) = i\}$$

$$2) \quad 0.d_1 d_2 \dots d_t = \sum_{i=1}^t d_i \beta^{-i} \quad (d_i \text{ son las cifras del número representado})$$

Ejemplo 1

$$t = 3, \beta = 10, L = -10, U = 10$$

Nuestros números serán de la forma  $\{\pm 0.d_1 d_2 d_3 \times 10^e, \text{ con } 0 \leq d_i \leq 9, -10 \leq e \leq 10\}$

Cómo se distribuyen nuestros números de  $FP_e$  en este caso sobre los reales?

Si  $e = 1$

Un número cualquiera del conjunto puede expresarse como suma:

$$\eta_1 = \left(\frac{d_1}{10} + \frac{d_2}{100} + \frac{d_3}{1000}\right) \times 10^1 = d_1 + \frac{d_2}{10} + \frac{d_3}{100}$$

La separación entre números es  $10^{-2}$ , por lo que en FP se tendrá  $9 \times 10 \times 10 = 900$  números.

Si  $e=2$

$$\eta_2 = \left(\frac{d_1}{10} + \frac{d_2}{100} + \frac{d_3}{1000}\right) \times 10^2 = 10 \times \left(d_1 + \frac{d_2}{10} + \frac{d_3}{100}\right)$$

La separación entre números es  $10^{-1} \Rightarrow$  FP<sub>2</sub> es una homotecia de razón 10 de FP<sub>1</sub>, por lo que también tiene 900 números.

Esto ocurre en general para cualquier FP<sub>e</sub>, y la separación allí será:  $10^{e-3}$  (se define, para  $x \in \text{FP}$ ,  $\text{separación}(x) = \min\{s > x\} - x$ ) (es la distancia al próximo número en FP)

Esto muestra que la separación relativa ( $\text{separación}(x) / x$ ) varía muy poco, por lo que los números de FP estarán muy juntos cerca del cero y comenzarán a distanciarse para valores crecientes de  $|x|$ .

Ejemplo

Según el estándar de IEEE para aritmética de punto flotante, la base es 2,  $L=-126$ ,  $U=127$ ,  $t=23$ , para simple precisión y  $t=52$  para doble precisión.

$$\text{FP} = \{\sigma \times 1.d_1 d_2 \dots d_t \times 2^e \text{ con } d_i = 1 \vee d_i = 0, e \in Z, L \leq e \leq U\}$$

Ahora que han sido definidos los FP<sub>e</sub>, se analizarán los errores de representación de números reales. Para fijar ideas se comenzará con un ejemplo simple.

Sea  $x = 1/3 \rightarrow \bar{x} = 0,333$  (representación de  $x$  en la aritmética del ejemplo 1)

Esta representación aparece como muy intuitiva. Sin embargo, la decisión sobre cómo aproximar el racional  $1/3$  a un elemento de FP<sub>0</sub> no es única.

Lo anterior es conocido como “tablemaker’s dilemma”.

Las alternativas se denominan “por truncamiento” o “por redondeo”.

En el primer caso, dado  $x$ , se le representa por  $\bar{x} / \bar{x} \in \text{FP}, \bar{x} = \max(y \in \text{FP}, y \leq x)$

A modo de ejemplo:

$X = 2/3 \rightarrow \bar{x} = 0,667$  si la máquina redondea (esto es, si  $d_4 \geq 5 \Rightarrow$  suma 0,001;

si  $d_4 < 5 \Rightarrow$  solo "corta" el número)

↓

$\bar{x} = 0,666$  si trunca nada más.

Entonces, si se define  $E_x = x - \bar{x}$  (**error absoluto**) se cumplirá  $|E_x| \leq 0,0005 = 0,5 \times 10^{-3}$  si la máquina redondea.

En general, si se trabaja con base  $\beta$  y mantisa de  $t$  números

$$|E_x| \leq \frac{1}{2} \beta^{e-t} \quad \text{para una máquina que redondea}$$

$$|E_x| \leq \beta^{e-t} \quad \text{para una máquina que trunca}$$

¡Verifíquelo! Sugerencia: considere la diferencia entre 2 números consecutivos de  $FP_e$ .

### DEFINICIÓN DEL $\epsilon$ DE LA MÁQUINA ( $\epsilon_{MACH}$ )

$$\epsilon_{MACH} = \min\{x : FP(1+x) > 1\}$$

En general ocurrirá que:

$$\epsilon_{MACH} = \begin{cases} \frac{1}{2}\beta^{e-t} & \text{(para máquina que redondea)} \\ \beta^{e-t} & \text{(para máquina que trunca)} \end{cases}$$

Lo anterior surge directamente de aplicar las acotaciones anteriores para  $E_x$ .

En MATLAB  $\epsilon$  se define como la distancia entre 1 y el siguiente número de FP. Mostrar que, según la anterior definición, en la hipótesis del redondeo en la suma, ese valor es igual a  $2 \times \epsilon_{MACH}$ .

Nótese que  $\epsilon_{MACH}$  depende de la arquitectura de la máquina y a la vez, de cómo está implementada la suma (truncamiento o redondeo).

Ejercicio:

Estimar  $\epsilon$  mediante algún algoritmo iterativo.

También resulta conveniente estudiar el error relativo, que se define como  $e_x = \frac{x - \bar{x}}{x}$  para  $x \neq 0$ . Para la representación en Punto Flotante de un determinado número, vale el siguiente resultado:

$$FP(x) = x(1 + \delta_x) \text{ con } |\delta_x| \leq \epsilon_{MACH}$$

En general como regla práctica se puede decir que

$$E_x \sim 10^{-\# \text{ decimales OK}}$$

$$e_x \sim 10^{-\# \text{ cifras significativas OK}}$$

OBS: # cifras significativas  $\equiv$  # dígitos significativos

### PROPAGACIÓN DE ERRORES

Como en general el error será consecuencia de un cúmulo de errores ocurridos en pasos sucesivos, se debe estudiar la mecánica de “propagación” de los mismos a lo largo del cálculo.

Un mito común es que las computadoras modernas trabajan con tal grado de precisión que los usuarios no necesitan contemplar la posibilidad de resultados inexactos. Esto se ve reforzado cuando vemos en la pantalla los resultados con gran cantidad de cifras. Sin embargo, veremos a lo largo del curso que la falta de cuidado en cálculos aparentemente directos y triviales puede conducir a resultados catastróficos.

#### REGLAS DE PROPAGACIÓN

1) Suma

$$|E_{x+y}| \approx |E_x| + |E_y|$$

2) Producto y cociente (si  $e_x$  y  $e_y$  son pequeños)

$$i) \quad |e_{xy}| \cong |e_x| + |e_y|$$

$$\text{ii)} \quad |e_{x/y}| \cong |e_x| + |e_y|$$

3) Caso General (para errores pequeños)

$$|E_x| \cong \sum_{t=1}^n \left| \frac{\partial f}{\partial x_t} \right| |E_{x_t}|$$

Por último observamos que en máquinas con propiedades numéricas razonables, el resultado almacenado de una operación de punto flotante entre 2 números pertenecientes a FP (o sea números representables exactamente) a y b satisface

$$\text{FP}(a \text{ op } b) = (a \text{ op } b) (1 + \delta_{\text{op}})$$

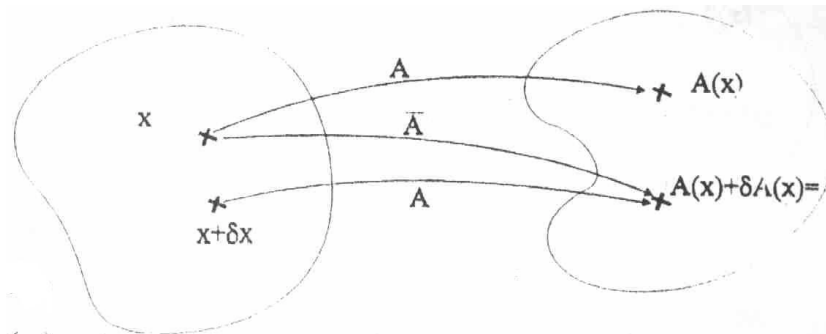
Donde “op” es una de las operaciones básicas (suma, resta, producto o división). El valor de  $\delta_{\text{op}}$  satisface típicamente  $|\delta_{\text{op}}| \leq \epsilon_{\text{MACH}}$

## NUMEROS DE CONDICION PARA PROBLEMAS Y ALGORITMOS

Pueden existir varias razones para obtener pobres resultados en la solución numérica de un determinado problema. Puede ser debido a que el algoritmo utilizado para la resolución sea inadecuada, pero puede darse el caso en que los resultados de nuestro problema sea muy sensible a perturbaciones en los datos (independientes de la elección de nuestro algoritmo). En el primer caso se dice que el algoritmo esta mal condicionado, mientras que en el segundo se dice que el problema esta mal condicionado.

Siguiendo esta línea de análisis, se definen:

El n° de condición de un Algoritmo A, que denominaremos  $C_A$ :



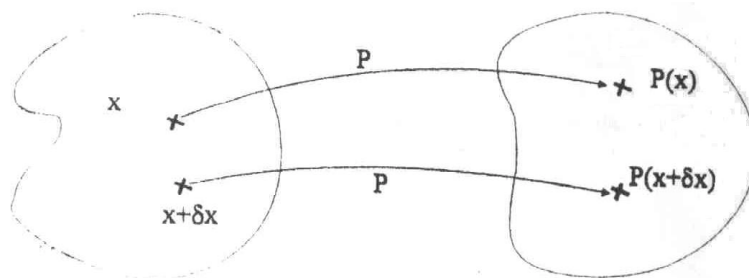
### Notación:

- $A(x)$  es el valor exacto (sin redondeo en las operaciones intermedias) calculado con el algoritmo A. Asumiremos que la función A es invertible.
- $\bar{A}(x)$  es el valor calculado con una maquina que tiene una unidad de redondeo asumiremos tambien que existe  $A^{-1}(\bar{A}(x))$ .

### Definición:

$$C_{A(x)} = \frac{\|\delta x\|}{\|x\| \varepsilon_{\text{MACH}}} = \frac{\|A^{-1}(\bar{A}(x)) - x\|}{\|x\| \varepsilon_{\text{MACH}}}$$

El n° de condición de un Problema P, que denominaremos  $C_P$ :



**Notación:**

$P(x)$  es la solución exacta del problema con datos  $x$

$P(x + \delta x)$  es la solución exacta del problema con datos  $x + \delta x$

Definición de número de condición del problema:

$$C_{P(x)} = \max_{\frac{\|\delta x\|}{\|x\|} \leq \varepsilon} \frac{\|P(x + \delta x) - P(x)\|}{\|P(x)\|}$$

Con esto estimamos la sensibilidad relativa de los resultados respecto a los datos.

Si el número de condición es de la forma  $10^n$  en la operación se pierden  $n$  cifras. Normalmente  $C_P$  es difícil de estimar, y a lo sumo solo se puede acotar. Si  $C_P$  es muy alto, puede ocurrir que el resultado no tenga ninguna cifra correcta.

## ALGUNAS TÉCNICAS USADAS EN CÁLCULO NUMÉRICO

### Noción de iteración

Una de las ideas más frecuentes en diferentes contextos es la iteración del latín “iteratio” que significa repetición. Visto de una manera general, la iteración implica la repetición de una acción o proceso. En este sentido, se trata de aplicar en muchos casos un método numérico en repetidas ocasiones para ir mejorando la estimación del resultado buscado.

Para ilustrar este punto, se considera el problema de resolver la ecuación  $x = F(x)$ . (se asume  $F \in C^n$ )

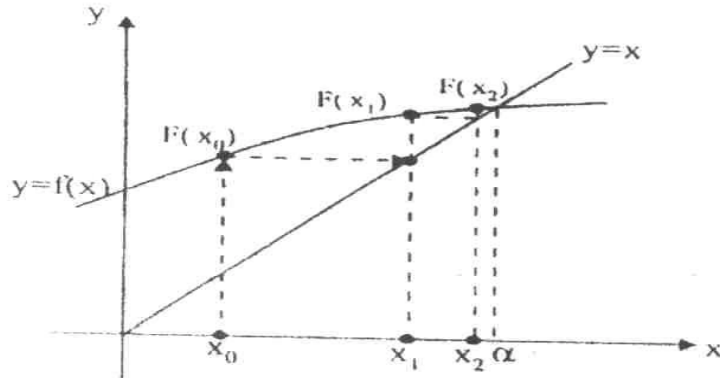
Usando el método de la iteración, se comienza con un valor inicial supuesto  $x_0$  y se calcula la secuencia:

$$x_1 = F(x_0), \quad x_2 = F(x_1) \text{ y en general } x_{n+1} = F(x_n).$$

Cada paso del cálculo se llama también iteración. Si  $\{x_n\}$  tiene límite  $\alpha$  ocurrirá:

Y  $\alpha$  es la solución de la ecuación.

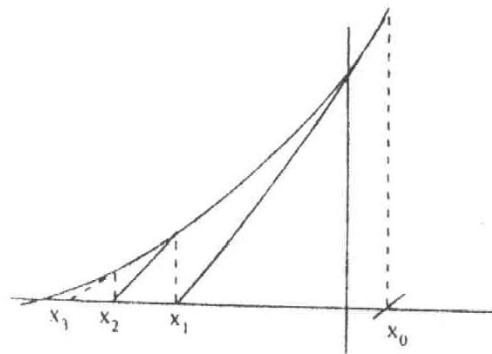
Por lo tanto, con  $n$  suficientemente grande se puede alcanzar la precisión deseada en el caso en que  $x_n \rightarrow \alpha$ . (Siempre que esa precisión deseada este dentro de lo realizable por la aritmética de FP)



### Linealización local

Otra idea es la aproximación local (esto es en un pequeño entorno de una función complicada por otra función mas simple lineal).

Un ejemplo de esta idea es el método de Newton-Raphson para resolver el problema  $F(x)=0$ . Si  $x_0$  es próximo a la raíz  $\alpha$  se aproxima  $F(x)$  por su tangente en  $x_n$ , halla el  $x_{n+1}$  como el cero de dicha tangente y se itera hasta la convergencia (que es segura solo bajo ciertas hipótesis)



No debe confundirse el concepto de iteración con el de recursión . Esta ultima es una identidad matemática, valida para todo un rango.

### Extrapolación de Richardson

Esta técnica es utilizada cuando se quiere calcular el valor limite de una determinada función  $f(x,h)$  cuando el parámetro  $h$  tiende a cero.

En muchos casos, tomar el limite para  $h \rightarrow 0$  se hace prácticamente imposible, ya sea por los enormes esfuerzos requeridos, o porque simplemente el efecto de los errores de redondeo se vuelve inadmisibles, estableciendo una cota inferior en los valores de  $h$  que pueden ser usados.

Consideremos pues que estamos tratando de calcular  $f(x)$ ..... y que el error de truncamiento tiene el siguiente comportamiento:

$$\tilde{f}(x,h) - f(x) = C_r h^r + O(h^r) \quad r > p$$

La idea de la E.R. consiste en obtener una mejor aproximación de  $f(x)$  a partir del calculo de  $f(x,h)$  para 2 valores diferentes de  $h$ :

$$\tilde{f}(x, h) = f(x) + \mathcal{O}_p(h^p) + o(h^r)$$

$$\tilde{f}\left(x, \frac{h}{q}\right) = f(x) + \mathcal{O}_p\left(\frac{h}{q}\right)^p + o(h^r)$$

Obs:  $q > 1$

$$\Rightarrow \boxed{\tilde{f}\left(x, \frac{h}{q}\right) + \frac{\tilde{f}\left(x, \frac{h}{q}\right) - \tilde{f}(x, h)}{q^p - 1} = f(x) + o(h^r)}$$

Se ha obtenido una formula de mayor orden para el error el truncamiento.