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, ε_{MACH}.
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. Linealizacion local.
    c. Extrapolación de Richardson.

Errores

Introducción

En el curso se estudiarán 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).

Mientras la eficiencia computacional conduce al estudio del número de operaciones necesarias en los algoritmos utilizados, tratando de obtener al menos una estimación de su orden de magnitud, verificar la correctitud de los resultados obtenidos implica el 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.

Los errores son de distintos tipos, y pueden estar:

- en el modelo físico/matemático usado.
- en los datos.
- en la aproximación matemática del modelo.
- en las operaciones realizadas al resolver el problema en una computadora.

Sobre el último tipo de errores se va a centrar la atención en este documento, observando hechos comunes que pueden ocurrir al resolver problemas utilizando métodos numéricos. Para comprenderlos, se debe primero tener una visión clara de la aritmética utilizada por las computadoras digitales, y de cómo se representan los números reales en ellas.

Representación de punto flotante

Los computadores digitales no trabajan con todos los números reales, sino sólo con un conjunto finito de números racionales, que típicamente es de la forma:

$$\mathrm{FP} = \{ s \times 0.d_1 d_2 ....d_t \times b^{\,e} \; d_i \in N \text{ con } 0 \le d_i < b \,,\; d_1 \neq 0, e \in Z, L \le e \le U \}$$

Y al que se designa por su sigla en inglés FP (Floating Point), siendo:

$s \rightarrow$ Signo

$t \rightarrow$ Cantidad de cifras en la mantisa

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

$e \rightarrow$ Exponente

Notaciones:

1) $\mathrm{FP}_i = \{ x \in \mathrm{FP} : \text{exponente}(x) = i \}$

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

Ejemplo 1

$$t = 3,\ b = 10,\ \mathrm{L} = -10,\ \mathrm{U} = 10$$

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

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

Para e = 1, se tiene que:

Un número cualquiera del conjunto FP_1 puede expresarse mediante la suma:

$$h_1 = (\frac{d_1}{10} + \frac{d_2}{100} + \frac{d_3}{1000}) \times 10^1 = d_1 + \frac{d_2}{10} + \frac{d_3}{100}$$

Definiendo para cada $x \in \mathrm{FP}$, $\text{separación}(x) = min\{s > x\} - x)$ (la distancia entre x y el siguiente número en FP), se tiene que la separación entre números consecutivos en FP_1 es 10^{-2}, por lo que en FP_1 existen 9x10x10=900 números.

Para e = 2, se tiene que

$$h_2 = (\frac{d_1}{10} + \frac{d_2}{100} + \frac{d_3}{1000}) \times 10^2 = 10 \times (d_1 + \frac{d_2}{10} + \frac{d_3}{100})$$

La separación entre números es 10^{-1} => FP_2 es una homotecia de razón 10 de FP_1, por lo que también tiene 900 números.

Esto ocurre en general para cualquier FP_e, donde la separación allí será de 10^{e-3}.

Esto muestra que la separación relativa (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 2

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

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

Representación de números

Ahora que han sido definidos los FP_e, 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 FP 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_0 no es única.

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

Al utilizar truncamiento, dado x, se le representa por $\bar{x} / \bar{x} \in \mathrm{FP}, \bar{x} = \max\{y \in \mathrm{FP}, y \le x\}$.

Por su parte, utilizando redondeo dado x, se le representa por $\bar{x} / \bar{x} \in \mathrm{FP}$, tal que $|x - \bar{x}|$ es mínimo, $\forall \bar{x} \in \mathrm{FP}$.

A modo de ejemplo:

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

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

$\downarrow$

$\bar{x} = 0{,}666$ si la máquina trunca.

Definiendo el **error absoluto** como $E_x = x - \bar{x}$, se cumplirá $|E_x| \leq 0{,}0005 = 0{,}5 \times 10^{-3}$ si la máquina redondea y $|E_x| \leq 0{,}0005 = 0{,}5 \times 10^{-3}$ si la máquina trunca (otra notación usual para el error absoluto en x es Δx).

En general, si se trabaja con base b y mantisa de t números

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

$|E_x| \leq \beta^{1-t}$ para una máquina que trunca

Este resultado se verifica de modo sencillo, considerando la diferencia entre dos números consecutivos de FP_e

Otra magnitud relevante que resulta conveniente estudiar es el **error relativo**, que se define como $e_x = \frac{x - \bar{x}}{x}$ para $x \neq 0$. (Otra notación usual para el error relativo en x es dx)

DEFINICIÓN DEL e DE LA MÁQUINA (e_{MACH})

El e de la máquina es una magnitud dependiente del sistema de representación FP que juega un rol importante como cota de los errores relativos de representación. Se define como

$$\varepsilon_{\text{MACH}} = min(x : \text{FP}(1 + x) > 1\} .$$

En general ocurrirá que:

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

La expresión anterior surge directamente de aplicar las acotaciones presentadas para E_x .

En MATLAB $\varepsilon_{\text{MACH}}$ 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 $2.\varepsilon_{\text{MACH}}$.

Nótese que $\varepsilon_{\text{MACH}}$ depende de la arquitectura de la máquina y a la vez, de cómo está implementada la representación del resultado y la operación de suma (si se utiliza truncamiento o redondeo).

Ejercicio:

Estimar $\varepsilon_{\text{MACH}}$ mediante algún algoritmo iterativo.

$\varepsilon_{\text{MACH}}$ juega un rol importante como cota del error relativo de representación de punto flotante, expresado por el siguiente resultado:

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

En general, como regla práctica se puede decir que los errores de representación cumplen que:

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

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

PROPAGACIÓN DE ERRORES

Al resolver un problema utilizando métodos numéricos, 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

Los errores se propagan de acuerdo a las reglas que se presentan a continuación:

1) Suma: propaga los errores *absolutos.*

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

2) Producto y cociente (si e_x y e_y son pequeños): propaga los errores *relativos.*

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

ii) $$\left|e_{x/y}\right| \cong \left|e_x\right| + \left|e_y\right|$$

3) En caso general, al calcular ***f(x)***, siendo $x = (x_1, x_2, \mathbf{K}, x_N)$ y considerando errores pequeños que permitan desprec iar términos de mayor orden, se tiene que

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

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

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

Donde "*op*" es una de las operaciones básicas (suma, resta, producto o división). El valor de δ_{op} satisface típicamente $\left|\delta_{op}\right| \leq \varepsilon_{MACH}$

NÚMEROS 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 inadecuado, pero puede darse el caso en que los resultados de nuestro problema sean muy sensible a perturbaciones en los datos (independientes de la elección del algoritmo utilizado para la resolución). 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:

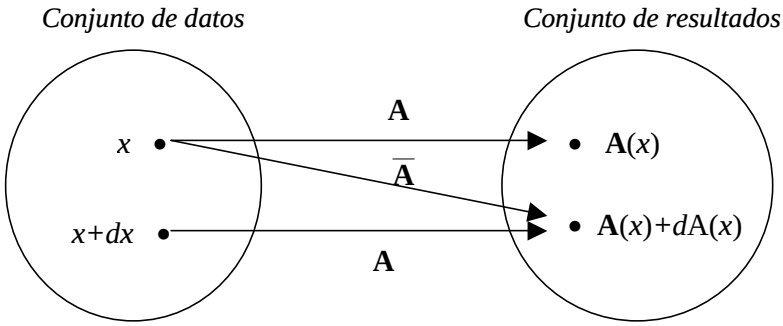
1) El número de condición de un Algoritmo A, que denominaremos C_A, definido por

$$\mathbf{C}_{\mathbf{A}(x)} = \frac{\|dx\|}{\|x\|.e_{MACH}} = \frac{\left\|\mathbf{A}^{-1}(\overline{\mathbf{A}}(x)) - x\right\|}{\|x\|.e_{MACH}}$$

En la formulación anterior, se han introducido como notación los siguientes elementos:

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

La idea de definición consiste en que la magnitud *número de condición de un algoritmo* refleja los errores "hacia atrás" en el proceso algorítmico, es decir hacia los datos del problema. Una representación gráfica se presenta a continuación, para un problema y algoritmo genérico.



Interpretación del número de condición de un algoritmo

1) El número de condición de un Problema P, que denominaremos C_P, definido por

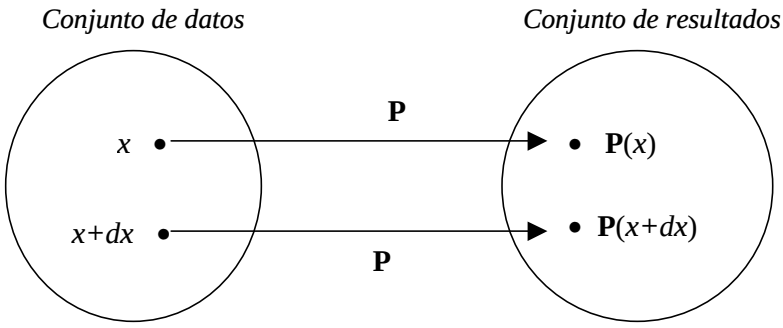
$$\mathbf{C}_{\mathbf{P}(x)} = \max_{\frac{\|dx\|}{\|x\|} < e_{MACH}} \frac{\frac{\|\mathbf{P}(x+dx) - \mathbf{P}(x)\|}{\|\mathbf{P}(x)\|}}{\frac{\|dx\|}{\|x\|}}$$

En la formulación anterior, se han introducido como notación los siguientes elementos:

- P(x) es la solución exacta del problema con datos x
- P(x + δx) es la solución exacta del problema con datos x + δx

La idea de definición consiste en que la magnitud *número de condición de un problema* permite estimar la sensibilidad relativa de los resultados respecto a perturbaciones en los datos del problema. Esta magnitud es una propiedad matemática, y por lo tanto es independiente del algoritmo utilizado para la resolución del problema, así como de cálculos intermedios y errores de representación del sistema de punto flotante.

Una representación gráfica se presenta a continuación, para un problema genérico.



Interpretación del número de condición de un problema.

Si el número de condición de un problema es de magnitud 10^n, en la resolución del problema se pierden ***n*** cifras de precisión. Normalmente C_p es difícil de estimar, y a lo sumo solo se puede acotar. Si C_p es muy alto (problema *mal condicionado*), puede ocurrir que el resultado no tenga ninguna cifra correcta.

ALGUNAS TÉCNICAS USADAS EN CÁLCULO NUMERICO

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 implícita 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 que $F \in C^n$)

Usando el método de la iteración, se comienza con un valor inicial supuesto x0 y se calcula la secuencia:

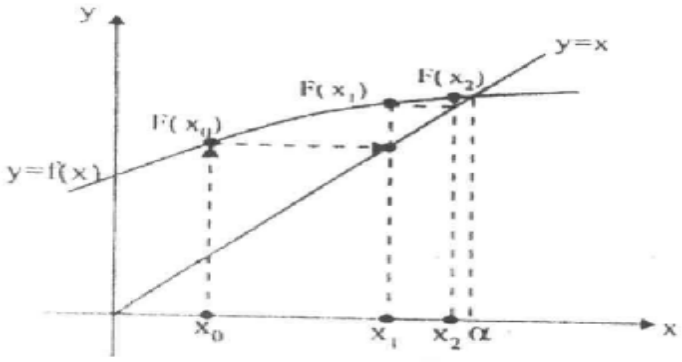
$$x_1 = F(x_0),\ x_2 = F(x_1) \text{ y en general } x_{n+1} = F(x_n).$$

Cada paso del calculo se llama también iteración . Si $\{x_n\}$tiene limite α ocurrirá:

$$a = \lim x_n = \lim F(x_n) = F(\lim x_n) = F(a) \Rightarrow a = F(a)$$

Siendo α, por lo tanto, la solución de la ecuación.

Por lo tanto, con *n* suficientemente grande se puede alcanzar la precisión deseada en la solución, en el caso en que la sucesión $x_n \rightarrow \alpha$. (Siempre que esa precisión deseada este dentro de lo realizable por la aritmética de FP). A continuación se presenta un esquema gráfico del funcionamiento de la técnica de iteración.



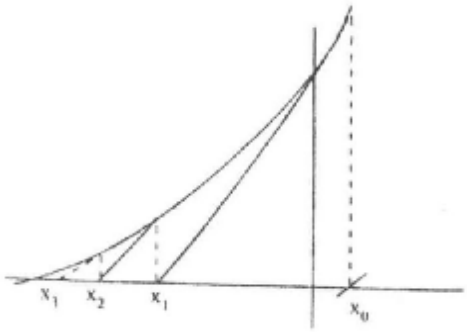
Ejemplo de iteración

No debe confundirse el concepto de iteración con el de recursión. Este último concepto denota a una identidad matemática, valida para todo un rango, mientras que la iteración utiliza una sucesión para aproximar una identidad, y toma su límite para la obtención del resultado.

Linealizacion local

Otra idea comúnmente utilizada es la aproximación local, que consiste en aproximar una función complicada por otra función más simple en un pequeño entorno. Cuando se asume una aproximación lineal, la técnica se denomina *linealizacion local*.

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 alfa 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). A continuación se presenta un esquema gráfico del funcionamiento de la técnica de linearización local.



Ejemplo de linearización local: método de Newton-Raphson

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 límite para $h \to 0$ se hace prácticamente imposible, ya sea por los enormes esfuerzos requeridos, o porque simplemente el efecto de los errores de redondeo se vuelve inadmisible, estableciendo una cota inferior en los valores de h que pueden ser usados.

Considerando que se trata de calcular $f(x) = \lim_{h \to 0} \bar{f}(x,h)$ y que el error de truncamiento tiene el siguiente comportamiento:

$$\bar{f}(x,h) - f(x) = C_p.h^p + \mathrm{O}(h^r) \text{ con } r > p$$

La idea de la extrapolación de Richardson consiste en obtener una mejor aproximación de $f(x)$ a partir del calculo de $\bar{f}(x,h)$ para 2 valores diferentes de *h* (tomando *q* > 1).

$$\bar{f}(x,h) = f(x) + C_p.h^p + \mathrm{O}(h^r)$$

$$\bar{f}\left(x,\frac{h}{q}\right) = f(x) + C_p.\left(\frac{h}{q}\right)^p + \mathrm{O}(h^r)$$

$$\bar{f}\left(x,\frac{h}{q}\right) + \frac{\bar{f}\left(x,\frac{h}{q}\right) - \bar{f}(x,h)}{q^P - 1} = f(x) + \mathrm{O}(h^r)$$

De este modo se obtiene una formula de mayor orden para el error el truncamiento.