Introducción al Análisis Numérico. Errores

María González Taboada

Departamento de Matemáticas

22 de febrero de 2007

Esquema:

1. Estudio matemático de un problema real
2. Análisis numérico y métodos constructivos
3. Tipos de problemas en análisis numérico y errores
4. Representación en coma flotante
   - Formato en coma flotante para números decimales
   - Representación en coma flotante de números binarios
   - El estándar IEEE 754
   - Exactitud de la representación en coma flotante

Estudio matemático de un problema real

- El proceso de representar la realidad mediante ecuaciones matemáticas se denomina modelado matemático.
- Etapas para resolver un problema real:
  1. Establecer un modelo matemático.
  2. Estudio teórico del modelo:
     existencia, unicidad, propiedades de la solución.
  3. Cálculo de la solución exacta (métodos clásicos) o de una aproximación (métodos numéricos o constructivos).
     Si se usan métodos numéricos:
     - Estudio del error.
     - Implementación de los métodos en el ordenador.
  4. Interpretación o visualización de la solución.

Análisis numérico y métodos constructivos

- **Análisis numérico**:
  Teoría de los métodos constructivos en el análisis matemático.

- **Método constructivo**:
  Procedimiento que permite obtener la solución de un problema con una precisión determinada en un número finitos de pasos.

Tipos de problemas en análisis numérico

- **Problemas de dimensión finita**:
  En el planteamiento del problema interviene un conjunto finito de números.

- **Problemas de dimensión infinita:**
  En el planteamiento del problema interviene un conjunto infinito de números.

Resolución de un problema de dimensión infinita

- Para resolver numéricamente un problema de dimensión infinita, el problema se aproxima por uno de dimensión finita. Este proceso se llama discretización y lleva asociado un error.

- El error de discretización mide la diferencia entre la solución del problema original y la solución del problema de dimensión finita que lo aproxima.

Resolución de un problema de dimensión finita

- **Métodos directos**:
  - Permiten calcular la solución del problema en un número finito de pasos, conocido *a priori*.
  - En la práctica, se cometen errores de redondeo, debido al empleo de sistemas de cálculo que usan airtmética finita.

- **Métodos iterativos:**
  - Construyen una sucesión diseñada para converger a la solución exacta del problema.
  - En la práctica, además de los errores de redondeo, se produce un error de truncamiento, al tomar un término de la sucesión como aproximación de la solución.

Representación en coma flotante

- **Formato entero:**
  Permite almacenar de forma exacta cierto subconjunto de los números enteros.

- **Formato en coma flotante:**
  Permite almacenar números no enteros y enteros fuera de ese subconjunto.
  El formato en coma flotante más utilizado en la actualidad es el formato IEEE 754.

Formato en coma flotante para números decimales

- Sea $x \neq 0$ un número real.
- Notación científica normalizada:

$$x = \sigma\, \bar{x}\, 10^{e}$$

donde

- $\sigma = \pm 1$ es el *signo*
- $e \in \mathbb{Z}$ es el *exponente*
- $\bar{x} \in [1/10, 1)$ es la *mantisa*, *significante* o *fracción*.

Ejemplo:

$x = 123{,}45 = (0{,}12345)\, 10^{3} \Rightarrow \sigma = +1,\ e = 3,\ \bar{x} = 0{,}12345$

Formato en coma flotante para números decimales

- Sea $x \neq 0$ un número real.
- Representación en coma flotante de un número del sistema decimal:

$$x = \sigma\, \bar{x}\, 10^{e}$$

con limitaciones en el número de dígitos en $\bar{x} \in [1, 10)$ y en el tamaño del exponente e.

Ejemplo:

Si se usan al menos 5 *dígitos para la mantisa:*

$x = 123{,}45 = (1{,}2345)\, 10^{2} \Rightarrow \sigma = +1,\ e = 2,\ \bar{x} = 1{,}2345$

Formato en coma flotante para números decimales

Nota:

Algunas calculadoras usan una aritmética decimal de coma flotante de diez dígitos:

- 10 *dígitos para* $\bar{x}$
- $e \in \mathbb{Z} \cap [-99, 99]$

En este caso, no podemos garantizar la representación exacta de más de diez dígitos de un número. Incluso puede ocurrir que algunos de los dígitos almacenados no sean exactos (por efecto del redondeo).

Formato en coma flotante para números binarios

- Sea $x \neq 0$ un número real.
- En el sistema binario:

$$x = \sigma\, \bar{x}\, 2^e$$

donde

- $\sigma = \pm 1$ es el *signo*
- $e \in \mathbb{Z}$ es el *exponente*
- $(1)_2 \leq \bar{x} < (10)_2$ es la *mantisa*, *significante* o *fracción*. (En el sistema decimal, $1 \leq \bar{x} < 2$).

Ejemplo:

$$x = (10101{,}11001)_2 \Rightarrow \begin{cases} \sigma = +1,\ e = (100)_2, \\ \\ \bar{x} = (1{,}010111001)_2 \end{cases}$$

Formato en coma flotante para números binarios

- Sea $x \neq 0$ un número real.
- Representación en coma flotante de un número binario:

$$x = \sigma\, \bar{x}\, 2^e$$

con restricciones en el número de dígitos binarios en $\bar{x}$ y en el tamaño del exponente e.

- El número de dígitos binarios permitido en la mantisa se denomina precisión de la representación binaria en coma flotante.

Ejemplo:

En el ejemplo anterior, para representar x de forma exacta se necesitan 10 dígitos para la mantisa y 3 para el exponente:
$x = (10101{,}11001)_2 = (+1)(1{,}010111001)_2\, 2^{(100)_2}$

El estándar IEEE 754

- Es el formato de representación de números binarios en coma flotante utilizado por la gran mayoría de los ordenadores actuales (p. ej., los procesadores Intel).
- Precisión simple:

$$x = \sigma(1.a_1 a_2 \ldots a_{23})2^e$$

  - Tiene una precisión de 24 dígitos binarios.
  - El exponente $e \in \mathbb{Z} \cap [-126, 127]$.

- Precisión doble:

$$x = \sigma(1.a_1 a_2 \ldots a_{52})2^e$$

  - Tiene una precisión de 53 dígitos binarios.
  - El exponente $e \in \mathbb{Z} \cap [-1022, 1023]$.

El estándar IEEE 754: precisión simple

$$x = \sigma(1.a_1 a_2 \ldots a_{23})2^e, \quad e \in \mathbb{Z} \cap [-126, 127]$$

- Se usan 4 bytes (es decir, 32 bits):

$$b_1 b_2 \ldots b_9 b_{10} \ldots b_{32}$$

donde

- $b_1 = 0$ si $\sigma = +1$ y $b_1 = 1$ si $\sigma = -1$,
- $b_2 \ldots b_9$ se usan para almacenar $E = e + 127$,
- $b_{10} \ldots b_{32}$ se usan para almacenar la mantisa (el primer dígito de la mantisa no se almacena en memoria).

El estándar IEEE 754: precisión simple

$$b_1 b_2 \ldots b_9 b_{10} \ldots b_{32}$$

- El cero se almacena de forma especial, con todos los bits cero: $b_i = 0$, $i = 1, \ldots, 32$.
- Si $E = (255)_{10} = (11111111)_2$ y $b_i = 0$, para $i = 10, \ldots, 32$, x representa los valores $\pm\infty$ ($\frac{1}{0}$).
- Si $E = (255)_{10} = (11111111)_2$ y algún $b_i \neq 0$, para $i = 10, \ldots, 32$, x representa *NaN* ($\frac{0}{0}$).

El estándar IEEE 754: precisión doble

$$x = \sigma(1.a_1 a_2 \ldots a_{52})2^e, \quad e \in \mathbb{Z} \cap [-1022, 1023]$$

- Se usan 8 bytes (es decir, 64 bits):

$$b_1 b_2 \ldots b_{12} b_{13} \ldots b_{64}$$

donde

- $b_1 = 0$ si $\sigma = +1$ y $b_1 = 1$ si $\sigma = -1$,
- $b_2 \ldots b_{12}$ se usan para almacenar $E = e + 1023$,
- $b_{13} \ldots b_{64}$ se usan para almacenar la mantisa (el primer dígito de la mantisa no se almacena en memoria).

El estándar IEEE 754: precisión doble

$$b_1 b_2 \ldots b_{12} b_{13} \ldots b_{64}$$

- El cero se almacena de forma especial, con todos los bits cero: $b_i = 0$, $i = 1, \ldots, 64$.
- Si $E = (2047)_{10} = (11111111111)_2$ y $b_i = 0$, para $i = 13, \ldots, 64$, x representa los valores $\pm\infty$ $(\frac{1}{0})$.
- Si $E = (2047)_{10} = (11111111111)_2$ y algún $b_i \neq 0$, para $i = 13, \ldots, 64$, x representa *NaN* $(\frac{0}{0})$.

Exactitud de la representación: el epsilon de máquina

- El epsilon de máquina es la diferencia entre 1 y el siguiente número mayor que 1 que puede ser almacenado.

- **Precisión simple:**

  El siguiente número binario mayor que 1 es $1{,}0\overbrace{\ldots}^{22}01$.
  Por tanto, el epsilon de máquina en este caso es

  $$2^{-23} \approx 1{,}19 \cdot 10^{-7}$$

  El formato IEEE de simple precisión puede usarse para almacenar aproximadamente 7 dígitos de un número decimal.

Exactitud de la representación: el epsilon de máquina

- El epsilon de máquina es la diferencia entre 1 y el siguiente número mayor que 1 que puede ser almacenado.

- **Precisión doble:**

  El epsilon de máquina es

  $$2^{-52} \approx 2{,}22 \cdot 10^{-16}$$

  Este formato puede usarse para almacenar aproximadamente 16 dígitos de un número decimal.

Exactitud de la representación: el mayor entero

- Otra forma de medir la exactitud de un formato en coma flotante es considerar el mayor entero M tal que pueden representarse (y almacenarse) de forma exacta todos los enteros positivos menores o iguales que M.

- Sea n es el número de dígitos binarios en la mantisa.

Exactitud de la representación: el mayor entero

- Todos los números naturales menores o iguales que

$$(1,1\overset{n-1}{\ldots}1)_2 \cdot 2^{n-1} = 2^n - 1$$

  pueden almacenarse de forma exacta.
- El número

$$2^n = (1,0\overset{n-1}{\ldots}0)_2 \cdot 2^n$$

  también se almacena de forma exacta.
- Para almacenar $2^n + 1$ números, serían necesarios $n + 1$ dígitos en la mantisa.
- Por tanto, $M = 2^n$.

Exactitud de la representación: el mayor entero

- **Precisión simple:**

$$M = 2^{24} = 16\,777\,216$$

Todos los enteros de 7 dígitos se almacenan de forma exacta.

- **Precisión doble:**

$$M = 2^{53} \approx 9{,}0 \cdot 10^{15}$$

Todos los enteros de 15 dígitos y la mayor parte de los enteros de 16 dígitos se almacenan de forma exacta.

Errores de *overflow* y *underflow*

- **Errores de *overflow* o desbordamiento**:
  - Ocurren cuando se trata de usar números demasiado grandes para el formato de coma flotante correspondiente.
  - En la mayoría de los ordenadores, son errores fatales y la ejecución del programa correspondiente se detiene.
  - El formato IEEE soporta estos errores, asignándoles el valor $\pm\infty$ ó NaN, según corresponda.
  - Normalmente hay errores en el programa que deben corregirse.
- **Errores de *underflow***:
  - Ocurren cuando se trata de crear números demasiado pequeños en magnitud.
  - La mayoría de los ordenadores toman el número como cero y continúan con las operaciones.