Aproximaciones y errores

Lección 02

Dr. Pablo Alvarado Moya

CE3102 Análisis Numérico para Ingeniería
Área de Ingeniería en Computadores
Tecnológico de Costa Rica

I Semestre 2018

Contenido

Aproximaciones y errores

Aproximación

- Modelo matemático *aproxima* comportamiento real
- Método numérico *aproxima* solución analítica
- Discrepancia entre objeto y su aproximación $\equiv$ **error**
- Si se desconoce objeto $\Rightarrow$ aproximar error

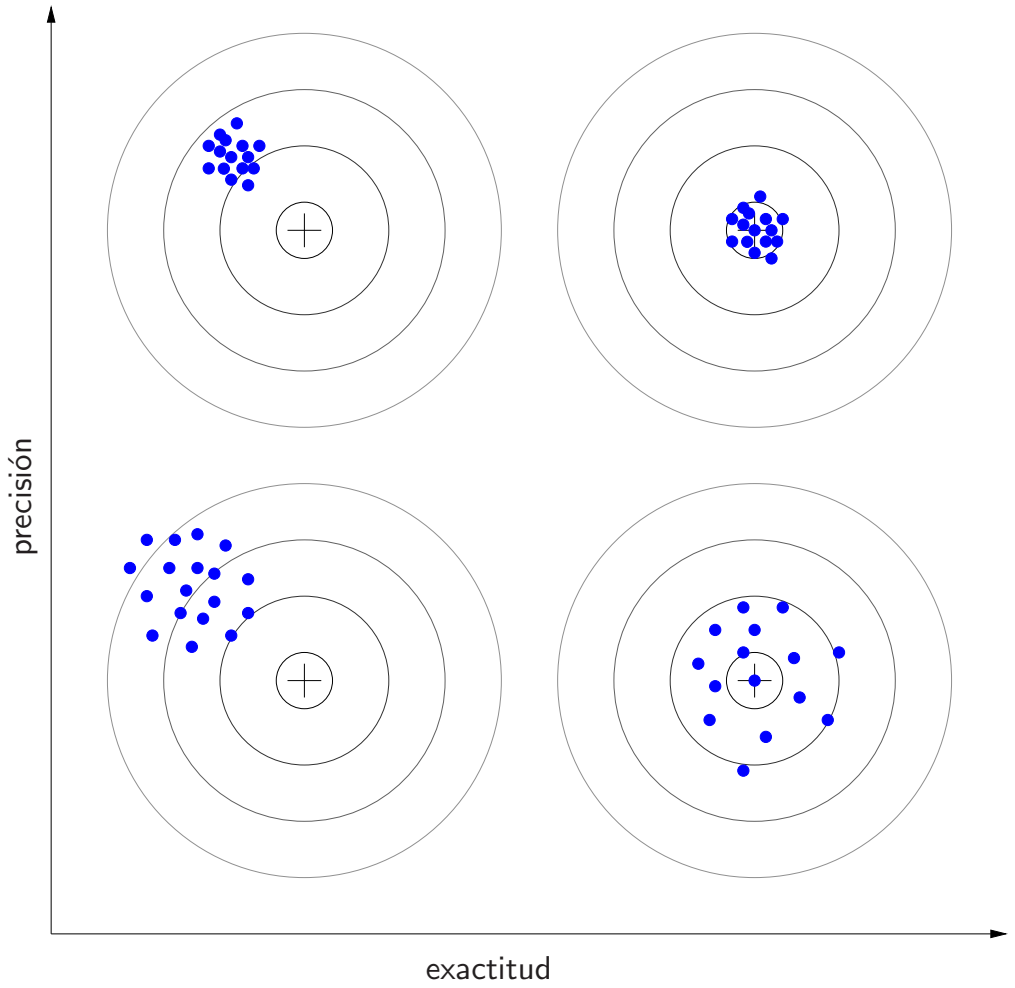
Cifras significativas

- Procesos de medición tienen precisión limitada
- Cifras significativas:
  - Indican confiabilidad de un valor numérico
  - Igual a número de dígitos obtenidos con certeza, más uno estimado
- Los métodos numéricos aproximan resultados
  $\Rightarrow$ debe especificarse cuántas cifras significativas son válidas
- Números irracionales (π, e, $\sqrt{2}$) sin representación exacta
  $\Rightarrow$ *redondeo* a número específico de cifras significativas

Exactitud y precisión

- Exactitud: qué tan cercano está el valor medido o calculado de valor verdadero
  (Sesgo o *bias*)
- Precisión (o incertidumbre): qué tanto se dispersan las mediciones alrededor del valor medido o calculado
  (Varianza)

Exactitud y precisión



Definiciones de error

Dos tipos principales

- **truncamiento**: aproximaciones de un procedimiento matemático exacto
- **redondeo**: representaciones numéricas con cifras significativas limitadas

Error verdadero

- Error verdadero E_t:

$$E_t = \text{valor verdadero} - \text{valor aproximado}$$

- calculable solo si se cuenta con el *valor verdadero*
- (De otro modo debe aproximarse y es error *aproximado*)
- Ignora orden de magnitud de valor estimado

Error relativo verdadero

- El error relativo *fraccional* verdadero considera orden de magnitud de valor estimado:

$$E_{\text{rel}} = \frac{E_t}{\text{valor verdadero}} = \frac{\text{valor verdadero} - \text{valor aproximado}}{\text{valor verdadero}}$$
$$= 1 - \frac{\text{valor aproximado}}{\text{valor verdadero}}$$

- El error relativo *porcentual* verdadero está dado por

$$\epsilon_t = E_{\text{rel}} \times 100\,\%$$

Error porcentual aproximado

- Si **no** se cuenta con el valor verdadero, entonces el error se normaliza con respecto al mismo valor aproximado:

$$\epsilon_a = \frac{\text{error aproximado}}{\text{valor aproximado}} \times 100\,\%$$

Error porcentual aproximado

- Si **no** se cuenta con el valor verdadero, entonces el error se normaliza con respecto al mismo valor aproximado:

$$\epsilon_a = \frac{\text{error aproximado}}{\text{valor aproximado}} \times 100\,\%$$

- Reto real: ¿cómo estimar el error si no se cuenta con el valor verdadero?

Error porcentual aproximado

- Si **no** se cuenta con el valor verdadero, entonces el error se normaliza con respecto al mismo valor aproximado:

$$\epsilon_a = \frac{\text{error aproximado}}{\text{valor aproximado}} \times 100\,\%$$

- Reto real: ¿cómo estimar el error si no se cuenta con el valor verdadero?
- En métodos iterativos se utiliza:

$$\epsilon_a = \frac{\text{aproximación actual} - \text{aproximación anterior}}{\text{aproximación actual}} \times 100\,\%$$

y los métodos se iteran mientras $|\epsilon_a| > \epsilon_s$

Umbral de Scarborough

Si se elige

$$\epsilon_s = (0,5 \times 10^{2-n})\,\%$$

entonces el resultado será correcto en *al menos n* cifras significativas

Ejemplo: Estimación de error en métodos iterativos (1)

Ejemplo

Utilice la representación en serie de e^x para estimar el valor de $e^{0,5}$ y los errores relativos porcentual verdadero y aproximado al agregar términos hasta que se alcancen al menos tres cifras significativas correctas.

Ejemplo: Estimación de error en métodos iterativos (2)

Solución:
La serie de potencias que representa a e^x es

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots + \frac{x^n}{n!} + \ldots$$

Para alcanzar tres cifras significativas se elige:

$$\epsilon_s = (0,5 \times 10^{2-3})\,\% = 0,05\,\%$$

Ejemplo: Estimación de error en métodos iterativos (3)

Considerando N términos, la aproximación de e^x será:

$$e^x \approx f_N(x) = \sum_{k=0}^{N-1} \frac{x^k}{k!}$$

Ejemplo: Estimación de error en métodos iterativos (4)

entonces

$$\epsilon_a = \frac{\text{aproximación actual} - \text{aproximación anterior}}{\text{aproximación actual}} \times 100\,\%$$

$$= \frac{\sum_{k=0}^{N-1} \frac{x^k}{k!} - \sum_{k=0}^{N-2} \frac{x^k}{k!}}{\sum_{k=0}^{N-1} \frac{x^k}{k!}} \times 100\,\%$$

$$= \frac{x^{N-1}}{(N-1)! \sum_{k=0}^{N-1} \frac{x^k}{k!}} \times 100\,\%$$

Ejemplo: Estimación de error en métodos iterativos (5)

Con $x = 0,5$ y $e^{0,5} \approx 1,64872127070013$ se tiene

N	$f_N(x)$	ϵ_t	ϵ_a
1	1	39,3469340287367 %	
2	1,50000000000	9,0204010431050 %	33,3333333333333 %
3	1,62500000000	1,4387677966971 %	7,6923076923077 %
4	1,64583333333	0,1751622556291 %	1,2658227848101 %
5	1,64843750000	0,0172115629956 %	0,1579778830964 %
6	1,64869791667	0,0014164937322 %	0,0157952930027 %
7	1,64871961806	0,0001002379603 %	0,0013162570913 %
8	1,64872116815	0,0000062196909 %	0,0000940182753 %

Tarea 1

Tarea 1

Programas para estimación de errores en el cálculo de funciones y operaciones simples.

Representaciones numéricas

Coma fija

Números codificados con coma fija

Las representaciones con **coma fija** son posicionales, donde el peso de cada bit en la representación es constante.

Número de N bits:

$$\begin{array}{cccccccc} b_{N-1} & \dots & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 \\ \uparrow & & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \\ 2^{N-1} & \dots & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ \text{MSB} & & & & & & & \text{LSB} \end{array}$$

Enteros sin signo

- Sea x un número entero sin signo de N-bits

$$x = \sum_{n=0}^{N-1} b_n 2^n$$

donde $b_n \in \{0, 1\}$ es el n-ésimo dígito de x.

Enteros sin signo

- Sea x un número entero sin signo de N-bits

$$x = \sum_{n=0}^{N-1} b_n 2^n$$

  donde $b_n \in \{0, 1\}$ es el n-ésimo dígito de x.
- El rango representable es entonces desde 0 hasta $2^N - 1$.

Enteros sin signo

- Sea x un número entero sin signo de N-bits

$$x = \sum_{n=0}^{N-1} b_n 2^n$$

  donde $b_n \in \{0, 1\}$ es el n-ésimo dígito de x.
- El rango representable es entonces desde 0 hasta $2^N - 1$.
- El dígito b_0 es el menos significativo (LSB, *least significant bit*) y su peso relativo es igual a uno.

Enteros sin signo

- Sea x un número entero sin signo de N-bits

$$x = \sum_{n=0}^{N-1} b_n 2^n$$

  donde $b_n \in \{0, 1\}$ es el n-ésimo dígito de x.
- El rango representable es entonces desde 0 hasta $2^N - 1$.
- El dígito b_0 es el menos significativo (LSB, *least significant bit*) y su peso relativo es igual a uno.
- El dígito b_{N-1} es el más significativo (MSB, *most significant bit*) y tiene un peso relativo de 2^{N-1}

Ejemplo (1)

Ejemplo

Encuentre el equivalente decimal del número binario

$$(10100101)_2$$

Ejemplo (2)

Solución: El número de 8 bits

$$x = (10100101)_2$$

es equivalente al número en base 10

$$\begin{aligned} x &= 1 \times 2^7 + 1 \times 2^5 + 1 \times 2^2 + 1 \times 2^0 \\ &= 128 + 32 + 4 + 1 \\ &= 165 \quad (= 1 \times 10^2 + 6 \times 10 + 5) \end{aligned}$$

Coma fija sin signo

- Sea x un número sin signo de N-bits

$$x = \frac{1}{M} \sum_{n=0}^{N-1} b_n 2^n$$

donde $b_n \in \{0, 1\}$ es el n-ésimo dígito de x y M es una constante de normalización elegida usualmente como 2^m.

Coma fija sin signo

- Sea x un número sin signo de N-bits

$$x = \frac{1}{M} \sum_{n=0}^{N-1} b_n 2^n$$

donde $b_n \in \{0, 1\}$ es el n-ésimo dígito de x y M es una constante de normalización elegida usualmente como 2^m.

- El rango representable es entonces desde 0 hasta $(2^N - 1)/M$.

Coma fija sin signo

- Sea x un número sin signo de N-bits

$$x = \frac{1}{M} \sum_{n=0}^{N-1} b_n 2^n$$

donde $b_n \in \{0, 1\}$ es el n-ésimo dígito de x y M es una constante de normalización elegida usualmente como 2^m.

- El rango representable es entonces desde 0 hasta $(2^N - 1)/M$.
- El dígito b_0 es el menos significativo (LSB, *least significant bit*) y su peso relativo es igual a $1/M$.

Coma fija sin signo

- Sea x un número sin signo de N-bits

$$x = \frac{1}{M} \sum_{n=0}^{N-1} b_n 2^n$$

  donde $b_n \in \{0, 1\}$ es el n-ésimo dígito de x y M es una constante de normalización elegida usualmente como 2^m.
- El rango representable es entonces desde 0 hasta $(2^N - 1)/M$.
- El dígito b_0 es el menos significativo (LSB, *least significant bit*) y su peso relativo es igual a $1/M$.
- El dígito b_{N-1} es el más significativo (MSB, *most significant bit*) y tiene un peso relativo de $2^{N-1}/M$.

Ejemplo (1)

Ejemplo

Encuentre el equivalente decimal del número binario

$$(10,100101)_2$$

Ejemplo (2)

Solución: El número de 8 bits

$$x = (10,100101)_2$$

es equivalente al número en base 10

$$\begin{aligned} x &= 1 \times 2^1 + 1 \times 2^{-1} + 1 \times 2^{-4} + 1 \times 2^{-6} \\ &= 2 + \frac{1}{2} + \frac{1}{16} + \frac{1}{64} \\ &= \frac{128 + 32 + 4 + 1}{64} = \frac{1}{64} 165 \\ &= 2,578125 \quad (= 2 \times 10^0 + 5 \times 10^{-1} + 7 \times 10^{-2} + \ldots) \end{aligned}$$

Ejemplo (3)

- En este caso se tiene $M = 64$, con dos bits en la parte entera y 6 en la parte fraccionaria
- Nóte que $M = 2^f$ con f el número de bits en la parte fraccionaria

Complemento a dos

- La representación de N bits de un número entero con signo en complemento a dos está dada por

$$x = -b_{N-1}2^{N-1} + \sum_{n=0}^{N-2} b_n 2^n$$

lo que permite representar números en el rango desde -2^{N-1} hasta $2^{N-1} - 1$.

Complemento a dos

- La representación de N bits de un número entero con signo en complemento a dos está dada por

$$x = -b_{N-1}2^{N-1} + \sum_{n=0}^{N-2} b_n 2^n$$

  lo que permite representar números en el rango desde -2^{N-1} hasta $2^{N-1} - 1$.
- El dígito b_0 es el menos significativo (LSB, *least significant bit*) y su peso relativo es igual a uno.

Complemento a dos

- La representación de N bits de un número entero con signo en complemento a dos está dada por

$$x = -b_{N-1}2^{N-1} + \sum_{n=0}^{N-2} b_n 2^n$$

  lo que permite representar números en el rango desde -2^{N-1} hasta $2^{N-1} - 1$.
- El dígito b_0 es el menos significativo (LSB, *least significant bit*) y su peso relativo es igual a uno.
- El dígito b_{N-2} es el más significativo (MSB, *most significant bit*) y tiene un peso relativo de 2^{N-2}.

Complemento a dos

- La representación de N bits de un número entero con signo en complemento a dos está dada por

$$x = -b_{N-1}2^{N-1} + \sum_{n=0}^{N-2} b_n 2^n$$

  lo que permite representar números en el rango desde -2^{N-1} hasta $2^{N-1} - 1$.
- El dígito b_0 es el menos significativo (LSB, *least significant bit*) y su peso relativo es igual a uno.
- El dígito b_{N-2} es el más significativo (MSB, *most significant bit*) y tiene un peso relativo de 2^{N-2}.
- El último bit b_{N-1} codifica al signo.

Sumas con complemento a dos

- El uso del complemento a dos es el más difundido de todas las representaciones de números con signo.

Sumas con complemento a dos

- El uso del complemento a dos es el más difundido de todas las representaciones de números con signo.
- Es posible sumar varios números con signo, y siempre que el resultado **final** se encuentre en el rango de representación, es irrelevante si resultados intermedios producen desbordamiento.

Sumas con complemento a dos

- El uso del complemento a dos es el más difundido de todas las representaciones de números con signo.
- Es posible sumar varios números con signo, y siempre que el resultado **final** se encuentre en el rango de representación, es irrelevante si resultados intermedios producen desbordamiento.
- Por ejemplo, supóngase que se debe hacer la secuencia de operaciones $2 + 3 - 2$ con números de 3 bits. La secuencia de adiciones es entonces

$(x_i)_{10}$	$(x_i)_2$	$(\sum x_i)_2$	$(\sum x_i)_{10}$
2_{10}	010	010	2_{10}
3_{10}	011	101	-3_{10}
-2_{10}	110	011	3_{10}

donde el resultado intermedio 5 fue representado por el número -3, sin afectar el resultado final.

Coma fija con signo

- La representación de N bits de un número con signo en complemento a dos está dada por

$$x = \frac{1}{M}\left(-b_{N-1}2^{N-1} + \sum_{n=0}^{N-2} b_n 2^n\right)$$

con la constante de normalización M elegida usualmente como 2^m.

Coma fija con signo

- La representación de N bits de un número con signo en complemento a dos está dada por

$$x = \frac{1}{M}\left(-b_{N-1}2^{N-1} + \sum_{n=0}^{N-2} b_n 2^n\right)$$

con la constante de normalización M elegida usualmente como 2^m.

- El rango representable será entonces desde $-2^{N-1}/M$ hasta $(2^{N-1}-1)/M$.

Ejemplo (1)

Ejemplo

Encuentre la representación binaria del número decimal $x = -3,125$ con cinco bits para la parte fraccionaria y tres bits para la parte entera utilizando coma fija con complemento a dos.

Ejemplo (2)

Solución:
Con $f = 5$ bits para la parte fraccionaria se obtiene

$$M = 2^f = 32$$

por lo que el número entero a convertir es

$$32 \times -3,125 = -100$$

y finalmente $-100 = -128 + (16 + 8 + 4) = (10011100)_2$

El caso fraccionario

- Un caso frecuentemente utilizado permite representar números de valor absoluto igual o inferior a uno empleando $M = 2^{N-1}$ con lo que se obtiene

$$x = -b_{N-1} + \sum_{n=0}^{N-2} b_n 2^{n-N+1}$$

Coma flotante

Números codificados con coma flotante

- La representación en coma flotante permite ampliar el rango de representación numérica.
- La separación entre dos números adyacentes es variable: pequeña para números pequeños, grande para números grandes.
- Las representaciones de 32 y 64 bits más frecuentemente utilizadas han sido estandarizadas por la IEEE (estándar 754 en su versión original de 1985 y su más reciente versión de 2008).

Codificación según IEEE 754

Un número codificado con el estándar consiste en

- un bit de signo s,
- el exponente e con E bits y
- la mantisa m normalizada (fraccionaria) de M bits,

y se codifica como

s	Exponente e	Mantisa m

Codificación según IEEE 754

Un número codificado con el estándar consiste en

- un bit de signo s,
- el exponente e con E bits y
- la mantisa m normalizada (fraccionaria) de M bits,

y se codifica como

s	Exponente e	Mantisa m

De forma algebraica, el número representado es

$$x = (-1)^s \times (1, m) \times 2^{e-\text{bias}}$$

con

$$\text{bias} = 2^{E-1} - 1$$

Equivalencia decimal de número en coma flotante

Nótese que la mantisa se completa con un 1 bit *oculto* (en el sentido de que no se indica explícitamente en la representación), mientras que los bits especificados en la mantisa representan solo la parte fraccionaria.

Ejemplo (1)

Ejemplo

Indique cuál es la representación en coma flotante del número $10,125_{10}$ en un formato de 14 bits que utiliza $E = 6$ bits y $M = 7$ bits.

Ejemplo (2)

Solución:

Primero, el bias está dado por

$$\text{bias} = 2^{E-1} - 1 = 2^5 - 1 = 31$$

y para la mantisa

$$10,125_{10} = 1010,0010_2 = 1,0100010_2 \times 2^3$$

El exponente corregido se obtiene con

$$e = 3 + \text{bias} = 34_{10} = 100010_2$$

Finalmente, la representación del número es:

s	Exponente e	Mantisa m
0	100010_2	0100010_2

Ejemplo (1)

Ejemplo

Encuentre qué número decimal es representado por el código de coma flotante con $E = 6$ bits y $M = 7$ bits:

s	Exponente e	Mantisa m
1	011110_2	1000000_2

Ejemplo (2)

Solución:
El número representado está dado por

$$-1 \times 1,1000000_2 \times 2^{30-\text{bias}} = -1,1_2 \times 2^{-1} = -0,11_2 = -0,75_{10}$$

Estándar de coma flotante IEEE 754-2008

	Simple	Doble
Ancho de palabra	32	64
Mantisa	23	52
Exponente	8	11
Bias	127	1023
Rango	$2^{128} \approx 3,4 \times 10^{38}$	$2^{1024} \approx 1,8 \times 10^{308}$

Algunos números especiales en precisión simple

Nombre	s	e	m	Hex
			$11\ldots11$	$\texttt{FFFFFFFF}_\text{H}$
-NaN (Quiet)	1	$11\ldots11$	$\vdots$	$\vdots$
			$10\ldots01$	$\texttt{FFC00001}_\text{H}$
			$01\ldots11$	$\texttt{FFBFFFFF}_\text{H}$
-NaN (Signal)	1	$11\ldots11$	$\vdots$	$\vdots$
			$00\ldots01$	$\texttt{FF800001}_\text{H}$
$-\infty$	1	$11\ldots11$	$00\ldots00$	$\texttt{FF800000}_\text{H}$
-0	1	$00\ldots00$	$00\ldots00$	$\texttt{80000000}_\text{H}$
$+0$	0	$00\ldots00$	$00\ldots00$	$\texttt{00000000}_\text{H}$
$+\infty$	0	$11\ldots11$	$00\ldots00$	$\texttt{7F800000}_\text{H}$
			$00\ldots01$	$\texttt{7F800001}_\text{H}$
+NaN (Signal)	0	$11\ldots11$	$\vdots$	$\vdots$
			$01\ldots11$	$\texttt{7FBFFFFF}_\text{H}$
			$10\ldots01$	$\texttt{7FC00000}_\text{H}$
+NaN (Quiet)	0	$11\ldots11$	$\vdots$	$\vdots$
			$11\ldots11$	$\texttt{7FFFFFFF}_\text{H}$

Algunos números especiales en precisión doble

Nombre	s	e	m	Hex
			$11\ldots11$	$\texttt{FFFFFFFFFFFFFFFF}_H$
-NaN (Quiet)	1	$11\ldots11$	$\vdots$	$\vdots$
			$10\ldots01$	$\texttt{FFC0000000000001}_H$
			$01\ldots11$	$\texttt{FFF7FFFFFFFFFFFF}_H$
-NaN (Signal)	1	$11\ldots11$	$\vdots$	$\vdots$
			$00\ldots01$	$\texttt{FFF8000000000001}_H$
$-\infty$	1	$11\ldots11$	$00\ldots00$	$\texttt{FFF0000000000000}_H$
-0	1	$00\ldots00$	$00\ldots00$	$\texttt{8000000000000000}_H$
$+0$	0	$00\ldots00$	$00\ldots00$	$\texttt{0000000000000000}_H$
$+\infty$	0	$11\ldots11$	$00\ldots00$	$\texttt{7FF0000000000000}_H$
			$00\ldots01$	$\texttt{7FF0000000000001}_H$
+NaN (Signal)	0	$11\ldots11$	$\vdots$	$\vdots$
			$01\ldots11$	$\texttt{7FF7FFFFFFFFFFFF}_H$
			$10\ldots01$	$\texttt{7FF8000000000000}_H$
+NaN (Quiet)	0	$11\ldots11$	$\vdots$	$\vdots$
			$11\ldots11$	$\texttt{7FFFFFFFFFFFFFFF}_H$

Error de redondeo

Se produce al utilizar representaciones numéricas incapaces de representar todas las cifras significativas del número a representar. Se produce porque

- El **rango** de cantidades representables es limitado. Fuera del rango representable ocurre el error de **desbordamiento** (*overflow*)
- Número **finito** de números representables en un rango. Al utilizar el número representable más cercano se produce el **error de cuantificación**. Este número se puede asignar por redondeo o por corte.
- Con coma flotante, intervalo Δx entre números aumenta conforme los números crecen en magnitud

Epsilon de formato

- En coma flotante, sea Δx el intervalo entre representaciones válidas alrededor de un valor x.

Epsilon de formato

- En coma flotante, sea Δx el intervalo entre representaciones válidas alrededor de un valor x.
- Si se utiliza corte, el epsilon $\mathscr{E}$ del formato se define como el menor número que cumple con

$$\mathscr{E} \geq \frac{|\Delta x|}{|x|}$$

Epsilon de formato

- En coma flotante, sea Δx el intervalo entre representaciones válidas alrededor de un valor x.
- Si se utiliza corte, el epsilon $\mathscr{E}$ del formato se define como el menor número que cumple con

$$\mathscr{E} \geq \frac{|\Delta x|}{|x|}$$

- Si se utiliza redondeo

$$\frac{\mathscr{E}}{2} \geq \frac{|\Delta x|}{|x|}$$

Epsilon de formato

- En coma flotante, sea Δx el intervalo entre representaciones válidas alrededor de un valor x.
- Si se utiliza corte, el epsilon $\mathscr{E}$ del formato se define como el menor número que cumple con

$$\mathscr{E} \geq \frac{|\Delta x|}{|x|}$$

- Si se utiliza redondeo

$$\frac{\mathscr{E}}{2} \geq \frac{|\Delta x|}{|x|}$$

- En general se cumple $\mathscr{E} = 2^{1-M}$ donde M es el número de bits de la mantisa.

Información sobre tipos en C++

- STL (Standard Template Library)
- `<limits>`
- `std::numeric_limits<float>::epsilon()`
- `std::numeric_limits<double>::max()`
- Ver ejemplo `eps.cpp` ($M_{\texttt{float}} = 23$, $M_{\texttt{double}} = 52$)

Información sobre tipos en C++

- STL (Standard Template Library)
- `<limits>`
- `std::numeric_limits<float>::epsilon()`
- `std::numeric_limits<double>::max()`
- Ver ejemplo `eps.cpp` ($M_{\texttt{float}} = 23$, $M_{\texttt{double}} = 52$)
- ¿Por qué resultado da $\mathscr{E} = 2^{-M}$ y no $\mathscr{E} = 2^{1-M}$?

Resumen

Este documento ha sido elaborado con software libre incluyendo LaTeX, Beamer, GNUPlot, GNU/Octave, XFig, Inkscape, LTI-Lib-2, GNU-Make y Subversion en GNU/Linux