# Unidad 1: Introducción a los métodos numéricos 2024

Prof.: Dr. Juan Ignacio Manassaldi

JTP: Ing. Amalia Rueda

# Importancia de los modelos matemáticos en IQ



- El procedimiento metodológico fundamental para resolver un problema en ingeniería consiste en representarlo de una manera adecuada, de tal forma de lograr una **sustitución del sistema real** (equipo, proceso, etc.) por uno más adecuado para el tratamiento formal.
- Las herramientas lógico-matemáticas nos brindan un marco útil para representar mediante un sistema de símbolos y reglas, el comportamiento de los sistemas reales.
- Un ***modelo matemático*** se define, de manera general, como una formulación o una ecuación que expresa las características esenciales de un sistema físico o de un proceso en términos matemáticos.

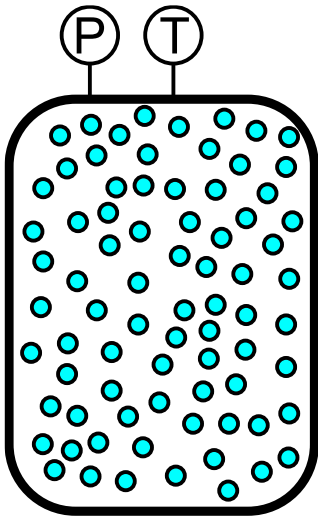
# Ejemplos de modelos matemáticos

EOS
Ecuaciones de estado

$$Pv = RT$$

Ley de los gases ideales

1856




Ecuación de Van der Waals

1873

$$P = \frac{RT}{v-b} - \frac{a}{v^2}$$

$$a = \frac{0.421875 R^2 T_c^{\,2}}{P_c} \quad b = \frac{RT_c}{8P_c}$$

Ecuación Peng Robinson

1976

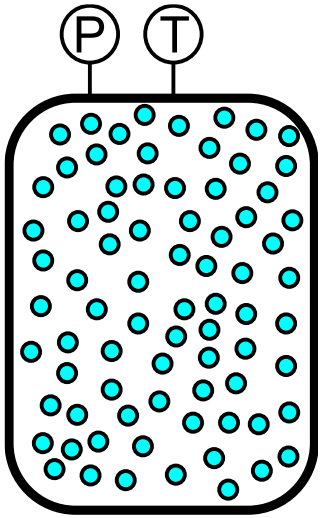
$$P = \frac{RT}{v-b} - \frac{a\alpha}{\left(v^2 + 2bv - b^2\right)}$$

$$a = \frac{0.45724 R^2 T_c^{\,2}}{P_c} \quad b = \frac{0.07780 RT_c}{P_c}$$

$$\alpha(T) = \left[1 + \left(0.37464 + 1.54226\omega - 0.2699\omega^2\right)\left(1 - \sqrt{T/T_c}\right)\right]^2$$

# Ejemplos de modelos matemáticos

EOS
Ecuaciones de estado




$$P = \sum_{n=1}^{9} a_n \rho^n + exp\left(-\rho_r{}^2\right) \sum_{n=10}^{15} a_n \rho^{2n-17}$$

$$a_1 = RT$$
$$a_2 = b_1 T + b_2 T^{1/2} + b_3 + b_4/T + b_5/T^2$$
$$a_3 = b_6 T + b_7 + b_8 T + b_9/T^2$$
$$a_4 = b_{10} T + b_{11} + b_{12}/T$$
$$a_5 = b_{13}$$
$$a_6 = b_{14}/T + b_{15}/T^2$$
$$a_7 = b_{16}/T$$
$$a_8 = b_{17}/T + b_{18} T^2$$
$$a_9 = b_{19}/T^2$$
$$a_{10} = b_{20}/T^2 + b_{21}/T^3$$
$$a_{11} = b_{22}/T^2 + b_{23}/T^4$$
$$a_{12} = b_{24}/T^2 + b_{25}/T^3$$
$$a_{13} = b_{26}/T^2 + b_{27}/T^4$$
$$a_{14} = b_{28}/T^2 + b_{29}/T^3$$
$$a_{15} = b_{30}/T^2 + b_{31}/T^3 + b_{32}/T^4$$

Ecuaciones de Benedict-Webb-Rubin modificadas (mBWR)

1973

# Modelos matemáticos



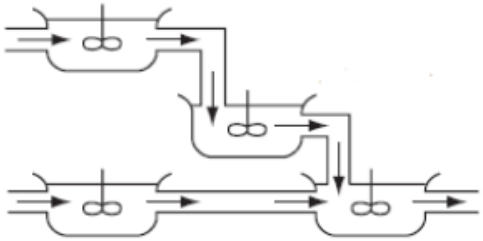
- Si bien el sistema real a estudiar es único, puede existir un número muy grande de modelos asociados al mismo.
- Para obtener un modelo que pueda resolverse (es decir que sea útil), resulta necesario adoptar un conjunto de hipótesis.
- No toda ecuación (o sistema de ecuaciones) puede resolverse fácilmente, al menos desde el punto de vista analítico.

# Modelos matemáticos



- Esto impuso a lo largo de la historia limitaciones importantes al tipo de modelos que podían resolverse, o de otra forma, la necesidad de recurrir a hipótesis inadecuadas o restrictivas (super-simplificaciones) para al menos poder tratar el problema.
- Contar con herramientas potentes para resolver sistemas de ecuaciones, o lo que es lo mismo, relativizar la necesidad de adoptar hipótesis inadecuadas al plantear modelos para resolver problemas complejos, resultó un gran paso adelante.

- La simulación es el acto de representar algunos aspectos del mundo real por números o símbolos que pueden manipularse fácilmente para facilitar su estudio.
- La simulación digital por computadora constituye una poderosa herramienta para la resolución de las ecuaciones (modelos matemáticos) que describen a los sistemas en ingeniería química.
- Durante el cursado iremos desarrollando métodos que facilitan la resolución de modelos matemáticos complejos. En su mayoría se agrupan bajo la clasificación de **métodos numéricos**.



# ¿Qué son los métodos numéricos?

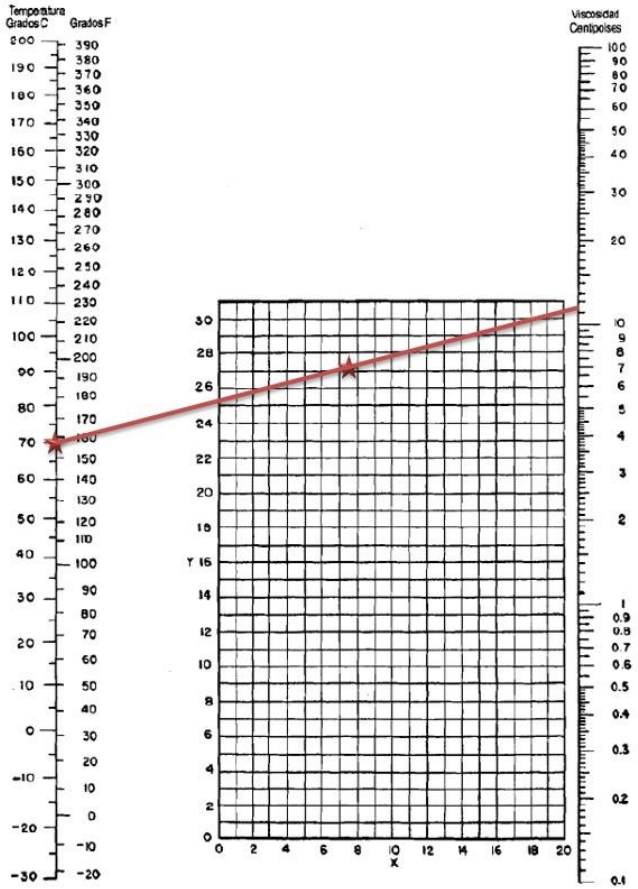
- Los métodos numéricos constituyen técnicas mediante las cuales es posible formular problemas matemáticos, de tal forma que puedan resolverse utilizando operaciones aritméticas.

- Utilizan únicamente operaciones lógicas y aritméticas; por consiguiente, pueden implementarse sobre computadoras digitales.

- Fueron desarrollados hace muchos años, pero su uso estuvo restringido hasta el advenimiento de las computadoras personales.

# Antes de las computadoras…

- Se encontraban las soluciones de algunos problemas usando <u>métodos exactos o analíticos</u>.
- Para analizar el comportamiento de los sistemas se usaban soluciones gráficas, las cuales tomaban la forma de <u>gráficas o nomogramas</u>.
- Para implementar los métodos numéricos se utilizaban <u>calculadoras y reglas de cálculo</u>.
- Se <u>gastaba bastante energía en la técnica misma de solución</u> en lugar de usarla en la definición del problema y su interpretación.



# A partir de las computadoras…



- Las computadoras y los métodos numéricos ofrecen una alternativa para los cálculos complicados.
- Se obtienen soluciones directamente, de esta manera se pueden aproximar los cálculos sin tener que recurrir a consideraciones de simplificación o a técnicas muy lentas.
- Los métodos numéricos representan opciones que aumentan, en forma considerable, la capacidad para enfrentar y resolver los problemas.
- Los métodos numéricos son herramientas muy poderosas para la solución de problemas. Son capaces de manipular sistemas de ecuaciones grandes, manejar no linealidades y resolver geometrías complicadas, comunes en la práctica de la ingeniería y, a menudo, imposibles de resolver en forma analítica.

# Antes vs Ahora

FORMULACIÓN

Leyes fundamentales explicadas brevemente

SOLUCIÓN

Métodos muy elaborados y con frecuencia complicados para hacer manejable el problema

INTERPRETACIÓN

Análisis profundo limitado por una solución que consume tiempo

FORMULACIÓN

Exposición profunda de la relación del problema con las leyes fundamentales

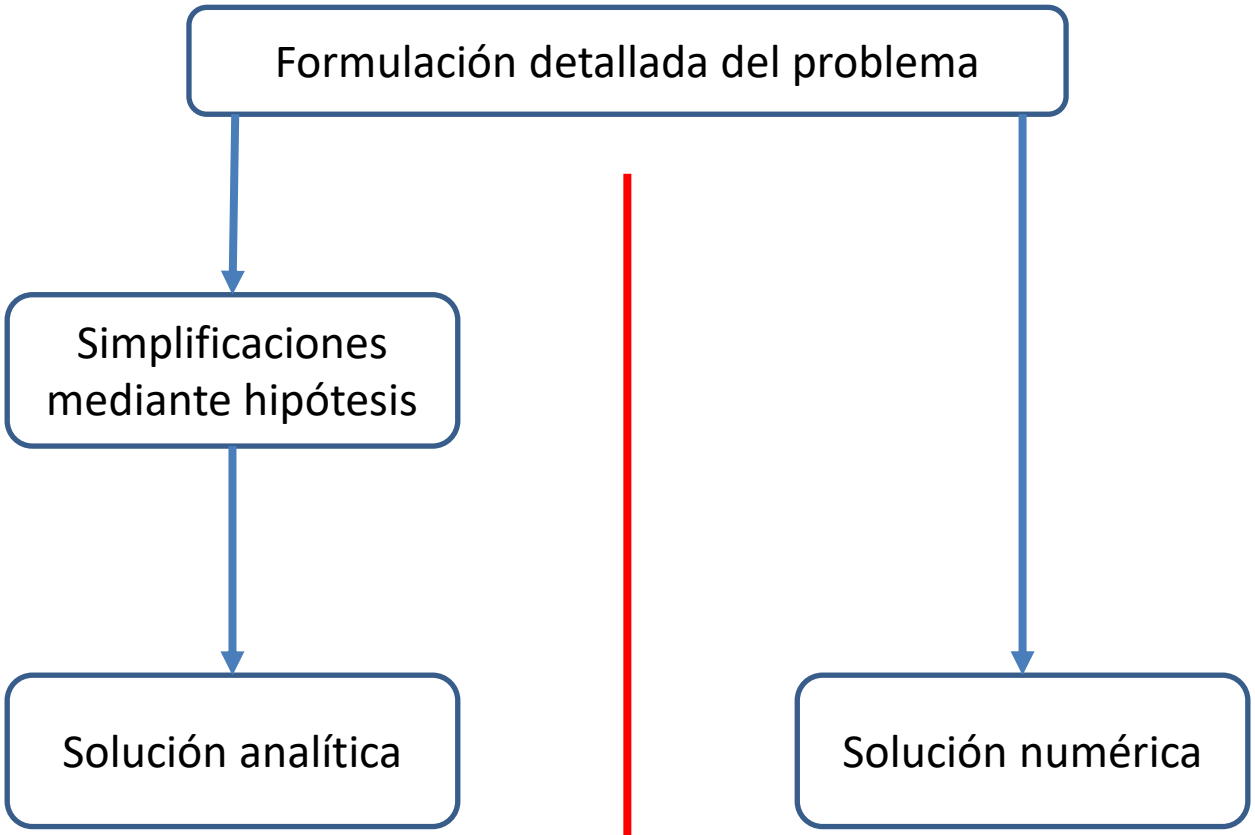
SOLUCIÓN

Método de la computadora fácil de usar

INTERPRETACIÓN

La facilidad de calcular permite pensar holísticamente y desarrollar la intuición; es factible estudiar la sensibilidad y el comportamiento del sistema

# Antes vs Ahora

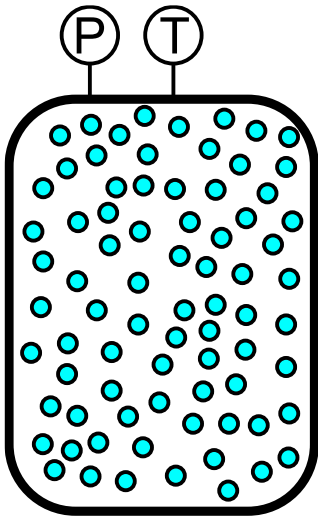



**Solución exacta de un modelo simplificado**

**Solución aproximada de un modelo detallado**

Tendencia actual

# Simple vs Riguroso




$$Pv = RT$$

Ley de los gases ideales

$$P = \frac{RT}{v-b} - \frac{a\alpha}{\left(v^2 + 2bv - b^2\right)}$$

Ecuación Peng Robinson

$$a = \frac{0.45724 R^2 T_c^{\,2}}{P_c} \quad b = \frac{0.07780 R T_c}{P_c}$$

$$\alpha(T) = \left[1 + \left(0.37464 + 1.54226\omega - 0.2699\omega^2\right)\left(1 - \sqrt{T/T_c}\right)\right]^2$$

# Fuente de Errores en la resolución de problemas



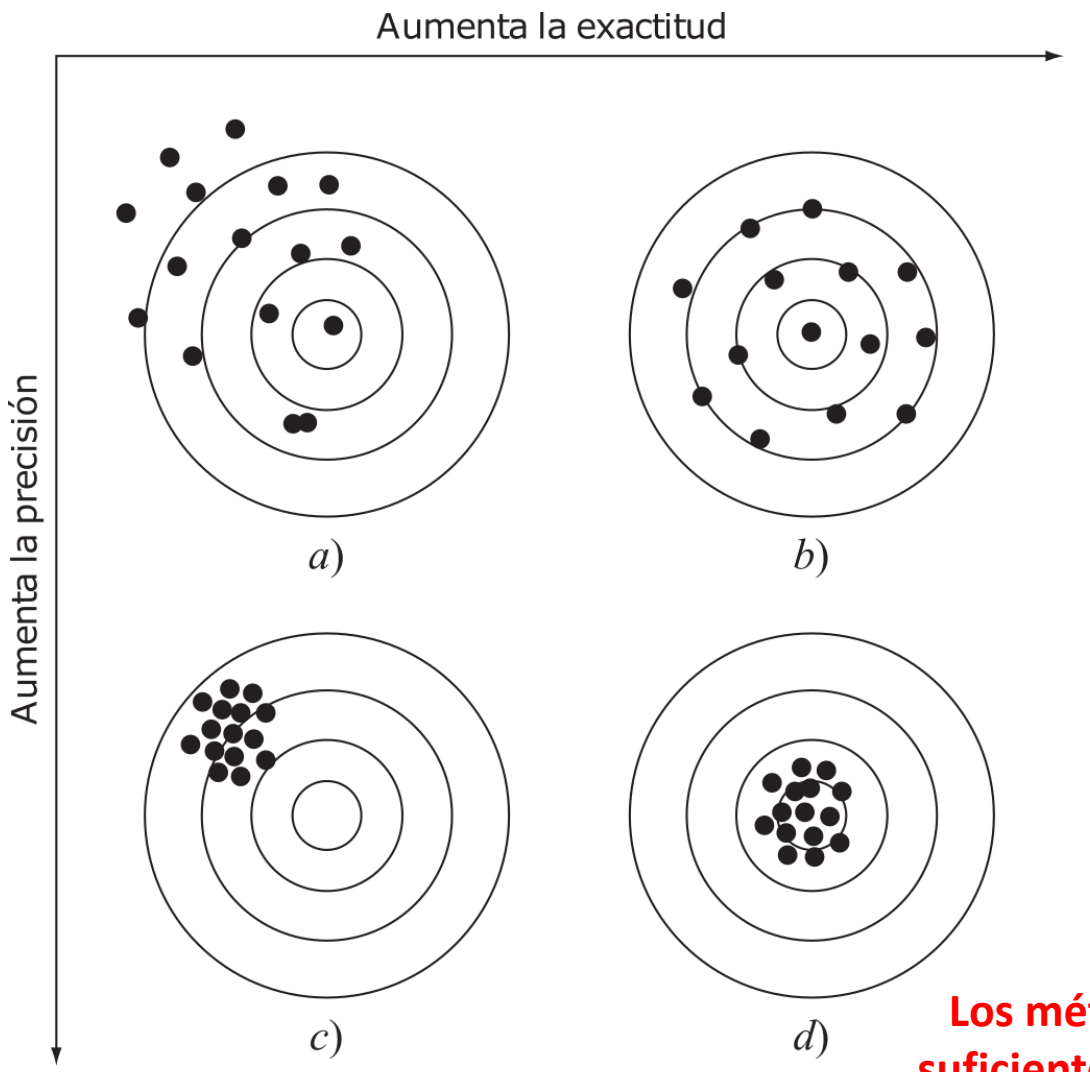
- Las aproximaciones numéricas presentan discrepancias en la resolución. Corresponden a una **aproximación de la solución**.
- Existen dos fuentes de errores:
  1. Errores en la formulación del modelo o incertidumbres en la obtención de los parámetros.
  2. **Errores numéricos propios de la resolución.**
- Durante el cursado nos centraremos principalmente en el segundo caso y trataremos de cuantificarlo y acotarlo dentro de una tolerancia.

# Exactitud y Precisión



- Los errores en cálculos y medidas se pueden caracterizar con respecto a su ***exactitud*** y su ***precisión***.
- La ***exactitud*** se refiere a qué tan cercano está el valor calculado o medido del valor verdadero.
- La ***precisión*** se refiere a qué tan cercanos se encuentran, unos de otros, diversos valores calculados o medidos.

# Exactitud y Precisión




a) Inexacto e impreciso
b) Exacto e impreciso
c) Inexacto y preciso
d) Exacto y preciso

**Los métodos numéricos deben ser lo suficientemente exactos y precisos para ser adecuados en el diseño de la ingeniería.**

# Definiciones de error



- Los errores numéricos surgen del uso de aproximaciones para representar operaciones y cantidades matemáticas exactas.

$$\text{Error verdadero} = \left|\text{Valor verdadero} - \text{Valor aproximado}\right|$$

No contempla el orden de magnitud.

$$\text{Error relativo fraccional} = \left|\frac{\text{Error verdadero}}{\text{Valor verdadero}}\right|$$

Contempla el orden de magnitud.

$$\text{Error relativo porcentual} = \left|\frac{\text{Error verdadero}}{\text{Valor verdadero}}\right| 100\%$$

Contempla el orden de magnitud.

Necesitamos conocer el valor verdadero para calcular el error.

Sin embargo, lo habitual es que el valor verdadero no esté disponible.

# Definiciones de error



- En numerosos métodos numéricos se usan procesos iterativos para calcular resultados.
- En tales procesos se hace una aproximación del resultado a partir de una aproximación disponible.
- Este proceso se efectúa varias veces, o de forma iterativa, esperando cada vez mejores aproximaciones.
- Una solución iterativa significa comenzar con un valor inicial, $x^{(0)}$, y generar una sucesión (secuencia) $x^{(0)}$, $x^{(1)}$, $x^{(2)}$, ..., $x^{(n)}$, tal que:

$$\lim_{n\to\infty} x^{(n)} = x^*$$

donde $\boldsymbol{n}$ representa el número de iteración (término de la correspondiente sucesión) y $\boldsymbol{x^*}$ representa la solución o valor deseado.

# Definiciones de error



- En estos casos se utiliza una aproximación del error:

$$\text{Error aprox. verdadero} = \left|\text{Valor actual} - \text{Valor anterior}\right|$$

$$\text{Error aprox. relativo fraccional} = \left|\frac{\text{Error aprox. verdadero}}{\text{Valor actual}}\right|$$

$$\text{Error aprox. relativo porcentual} = \left|\frac{\text{Error aprox. verdadero}}{\text{Valor actual}}\right| 100\%$$

$$\varepsilon_a = \left|x^{(i+1)} - x^{(i)}\right| \qquad \varepsilon_r = \frac{\left|x^{(i+1)} - x^{(i)}\right|}{\left|x^{(i+1)}\right|} \qquad \varepsilon_{\%} = \frac{\left|x^{(i+1)} - x^{(i)}\right|}{\left|x^{(i+1)}\right|} \times 100\%$$

# Definiciones de error



- Si se cumple la siguiente relación, entonces se considera que el resultado obtenido está dentro del nivel aceptable fijado previamente.

$$\varepsilon_a \leq \varepsilon_s$$

donde $\boldsymbol{\varepsilon_s}$ se conoce como tolerancia.

# Causas principales de errores en los métodos numéricos



## Errores de redondeo

Se deben a que la computadora tan sólo representa cantidades con un número finito de dígitos.

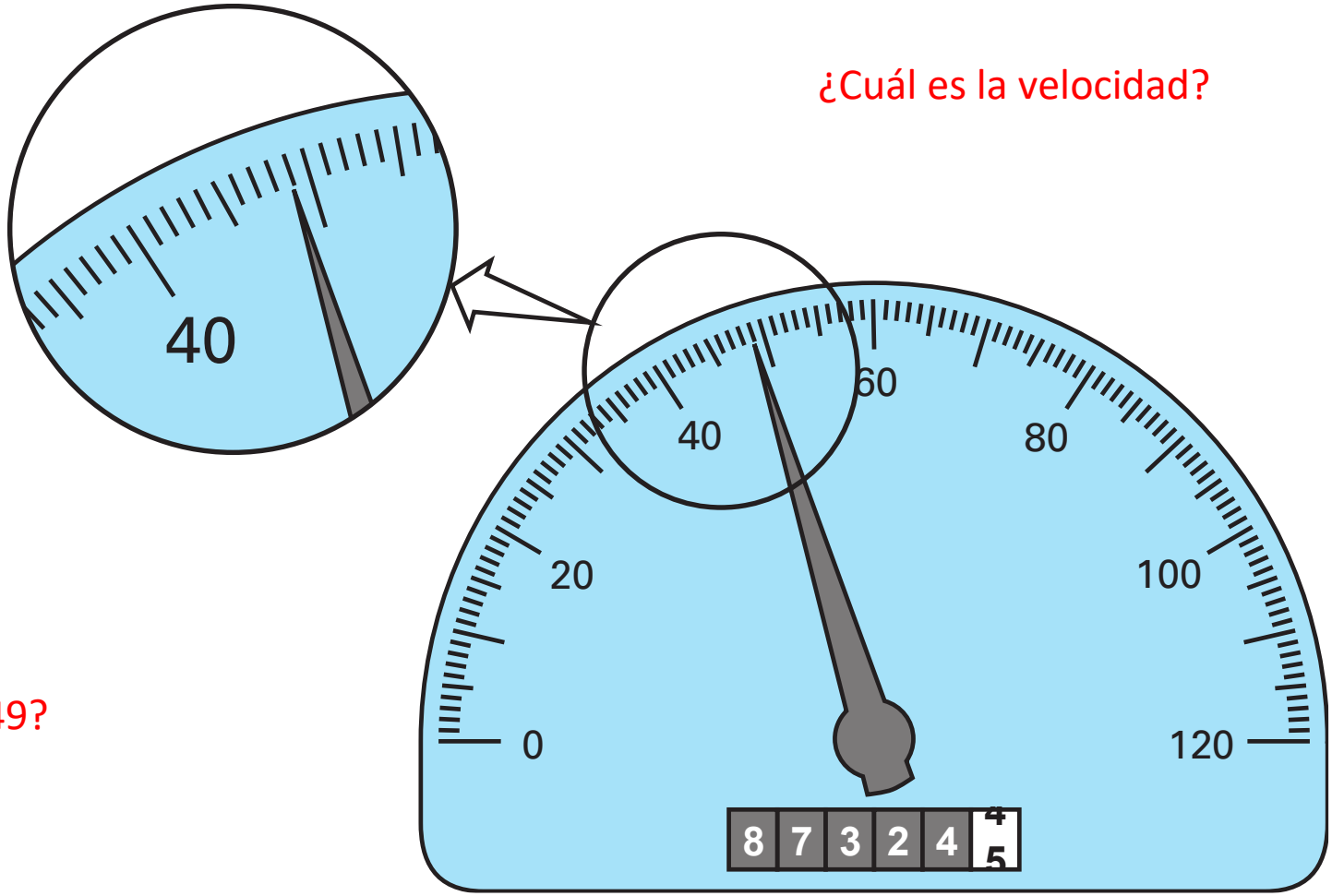
## Errores de truncamiento

Se deben a que los métodos numéricos pueden emplear aproximaciones para representar operaciones y cantidades matemáticas exactas.

# Errores de redondeo: Cifras significativas

- Cuando se emplea un número para realizar un cálculo, debe haber seguridad de que pueda usarse con confianza.

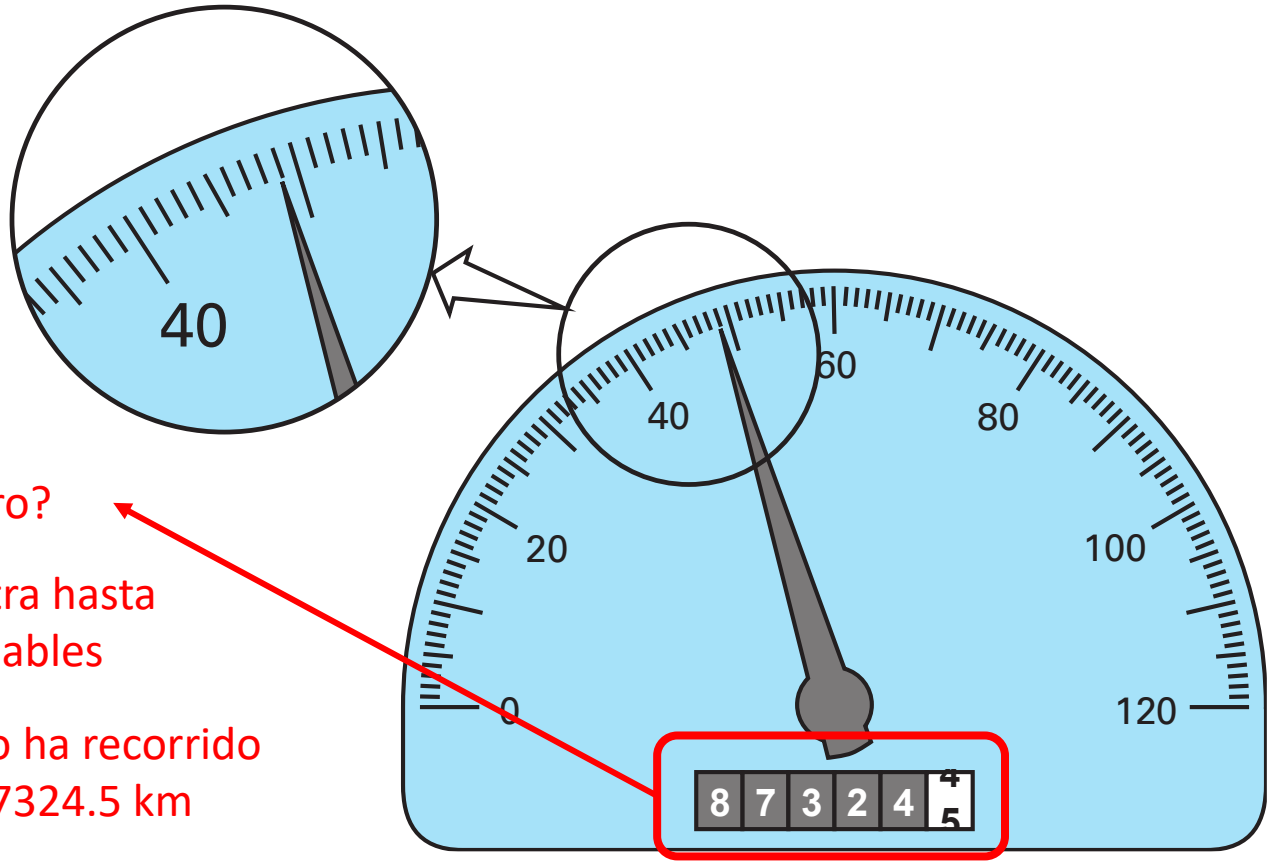
¿Cuál es la velocidad?




¿48?
¿49?
¿Entre 48 y 49?
¿48.8?
¿48.86421?
¿48.5?

# Errores de redondeo: Cifras significativas

- Debido a los límites del instrumento, únicamente se emplean con confianza los dos primeros dígitos.
- Estimaciones del tercer dígito (o más allá) sólo se considerarían aproximaciones.




¿Y el odómetro?

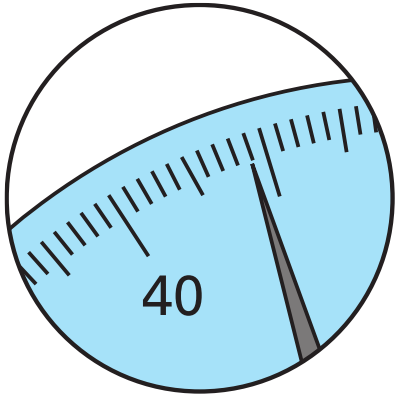
El odómetro muestra hasta seis dígitos confiables

A simple vista el vehículo ha recorrido un poco menos de 87324.5 km

# Errores de redondeo: Cifras significativas

- Las cifras significativas de un número son aquellas que pueden utilizarse en forma confiable.
- Se trata del número de dígitos que se ofrecen con certeza, más uno estimado.

3 cifras significativas



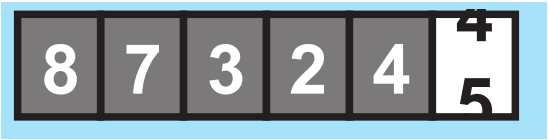

**Por convención al dígito estimado se le da el valor de la mitad de la escala menor de división en el instrumento de medición.**

**Lecturas:**

**48.5 km/h**

**87324.45 km**

7 cifras significativas



# Errores de redondeo: Cifras significativas



- Determinar las cifras significativas de un número es un procedimiento sencillo, aunque en algunos casos genera cierta confusión.
- Los ceros no siempre son cifras significativas, ya que pueden usarse sólo para ubicar el punto decimal: 0.00001845, 0.0001845 y 0.001845 tienen todos 4 cifras significativas.
- Por otro lado, cuando se incluyen ceros en números muy grandes, no queda claro cuántos son significativos: 45300 puede tener tres, cuatro o cinco dígitos significativos.
- La notación científica elimina estas incertidumbres: $0.1845\times10^{-4}$, $0.1845\times10^{-3}$, $4.53\times10^{4}$, $4.530\times10^{4}$, $4.5300\times10^{4}$, etc.

# Errores de redondeo: Cifras significativas



- Las computadoras retienen sólo un número finito de cifras significativas, algunos números jamás se podrán representar con exactitud. A la omisión de cifras significativas se le conoce como ***error de redondeo***.

$\pi$ 3.14159265358979323846264338327950288419716939937 51… (exacto)

PI() = 3.14159265358979 (Excel)

%pi = 3.14159265358979311599 8 (Scilab)

# Representación de números en la computadora



- Los errores de redondeo se relacionan de manera directa con la forma en que se guardan los números en la memoria de la computadora.
- La unidad fundamental mediante la cual se representa la información se llama palabra. Ésta es una entidad que consiste en una cadena de dígitos binarios o bits (binary digits).
- Los números son guardados en una o más palabras.

# Sistemas numéricos



- Es simplemente una convención para representar cantidades.
- Debido a que se tienen 10 dedos en las manos y 10 dedos en los pies, el sistema de numeración que nos es muy familiar es el decimal o de base 10.
- Una base es el número que se usa como referencia para construir un sistema.
- El sistema de base 10 utiliza 10 dígitos (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) para representar números.
- Tales dígitos son satisfactorios por sí mismos para contar de 0 a 9.
- Para grandes cantidades se usa la combinación de estos dígitos básicos; con la posición o valor de posición se especifica su magnitud (**notación posicional**).

# Ejemplo expresado en base decimal (10)

$$86409$$

$$\begin{array}{ccccc} 10^4 & 10^3 & 10^2 & 10^1 & 10^0 \\ | & | & | & | & | \\ 8 & 6 & 4 & 0 & 9 \end{array}$$

$$\begin{array}{ccrcr} 9 & \times & 1 & = & 9 \\ 0 & \times & 10 & = & 0 \\ 4 & \times & 100 & = & 400 \\ 6 & \times & 1000 & = & 6000 \\ 8 & \times & 10000 & = & 80000 \\ \hline & & & & 86409 \end{array}$$

**La potencia de la base está relacionada con la posición.**

**¿Tiene un límite para contar?**

**La disponibilidad de posiciones establece un límite a las cantidades que se pueden almacenar en un sistema real.**

# Ejemplo expresado en base decimal (10)

$$N = \sum_{i=0}^{n-1} a_i 10^i$$

**Número $N$ expresado en base 10 con $n$ posiciones disponibles**

| | | | | | |
|---|---|---|---|---|---|
| 8 | 6 | 4 | 0 | 9 | $\rightarrow a$ |
| 4 | 3 | 2 | 1 | 0 | $\rightarrow i$ |

***n=5***

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 8 | 6 | 4 | 0 | 9 |
| 6 | 5 | 4 | 3 | 2 | 1 | 0 |

***n=7***

$$N = \sum_{i=0}^{n-1} a_i b^i$$

**Número $N$ expresado en base $b$ con $n$ posiciones disponibles**

# Ejemplos expresados en base dos (binario)



$$(10101)_2$$

$$\begin{array}{ccccc} 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ | & | & | & | & | \\ 1 & 0 & 1 & 0 & 1 \end{array}$$

$$\begin{array}{ccccc} 1 & \times & 1 & = & 1 \\ 0 & \times & 2 & = & 0 \\ 1 & \times & 4 & = & 4 \\ 0 & \times & 8 & = & 0 \\ 1 & \times & 16 & = & 16 \\ & & & & \overline{21} \end{array}$$

$$(10101)_2 = 21$$

**Los números en la computadora se representan con un sistema binario o de base 2.**

**¿Por qué?**

**Se relaciona con el hecho de que las unidades lógicas fundamentales de las computadoras digitales sean componentes electrónicos de apagado/encendido (0 o 1).**

$$N = \sum_{i=0}^{n-1} a_i 2^i$$

# Ejemplos expresados en base 2 (binario)



- Por su naturaleza, las herramientas digitales almacenan los números en base 2 (binaria).
- La unidad fundamental mediante la cual se representa la información se llama palabra que consiste en una cadena de dígitos binarios o bits (binary digits).
- Los números son guardados en una o más palabras y cada bit representa una posición en el número.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | |

8 bits

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **0** | **0** | **0** | **1** | **0** | **1** | **0** | **1** |

Cadena de bits que almacena el numero 21

# Ejemplos expresados en base decimal (10)

$$86409.654 \Rightarrow \begin{array}{ccccccccc} 10^4 & 10^3 & 10^2 & 10^1 & 10^0 & & 10^{-1} & 10^{-2} & 10^{-3} \\ | & | & | & | & | & & | & | & | \\ 8 & 6 & 4 & 0 & 9 & . & 6 & 5 & 4 \end{array}$$

$$\begin{array}{ccccc} 4 & \times & 0.001 & = & 0.004 \\ 5 & \times & 0.01 & = & 0.05 \\ 6 & \times & 0.1 & = & 0.6 \\ \hline 9 & \times & 1 & = & 9 \\ 0 & \times & 10 & = & 0 \\ 4 & \times & 100 & = & 400 \\ 6 & \times & 1000 & = & 6000 \\ 8 & \times & 10000 & = & 80000 \\ \hline & & & & 86409.654 \end{array}$$

Potencias negativas de la base

Potencias positivas de la base

# Ejemplos expresados en base 2 (binario)

$$(10101.101)_2$$

| $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ | | $2^{-1}$ | $2^{-2}$ | $2^{-3}$ |
|---|---|---|---|---|---|---|---|---|
| \| | \| | \| | \| | \| | | \| | \| | \| |
| 1 | 0 | 1 | 0 | 1 | . | 1 | 0 | 1 |

$$\begin{array}{ccccc}
1 & \times & 0.125 & = & 0.125 \\
0 & \times & 0.25 & = & 0 \\
1 & \times & 0.5 & = & 0.5 \\
\hline
1 & \times & 1 & = & 1 \\
0 & \times & 2 & = & 0 \\
1 & \times & 4 & = & 4 \\
0 & \times & 8 & = & 0 \\
1 & \times & 16 & = & 16 \\
\hline
 & & & & 21.625
\end{array}$$

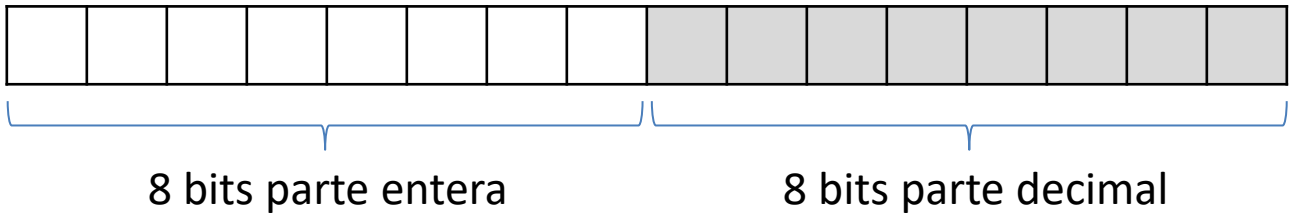
$$(10101.101)_2 = 21.625$$

**Podríamos establecer posiciones para los decimales, pero también tendríamos una limitación**

# Sistema de punto fijo



- Se asigna una cantidad fija de bits a la parte entera y otra cantidad fija de bits para la parte decimal.




| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

Cadena de bits que almacena el numero 21.625

**¿Cómo podríamos manejar los signos para almacenar números negativos?**

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

Cadena de bits que almacena el numero +21.625

**Se agrega el bit de signo. 0 positivo 1 negativo**

## ¿Qué limitaciones observan?

# Sistema de punto flotante



- El número se expresa como una parte fraccionaria, llamada ***mantisa (m)*** o significando, y una parte entera, denominada ***exponente (e)*** o característica.

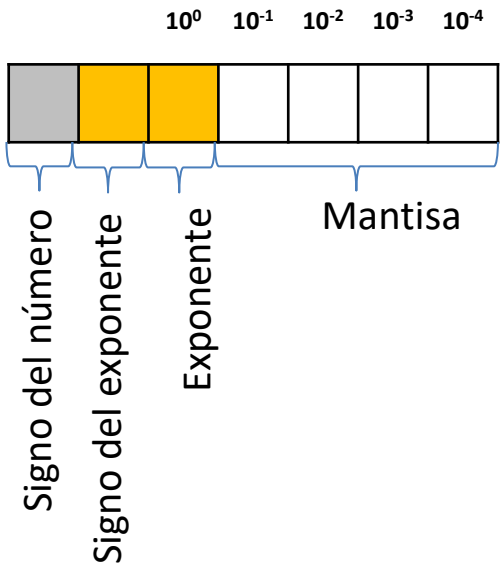
$$N = m \times b^{e} \qquad \rightarrow 156.78 = 0.15678 \times 10^{3}$$

- El sistema de punto flotante almacena el signo del número, el exponente con su signo y la mantisa normalizada.

# Sistema de punto flotante



- Ejemplo: Sistema de base 10 con cuatro puntos decimales para la mantisa y uno para el exponente.




$$10.78 = 0.1078 \times 10^2$$

| | | | | | | |
|---|---|---|---|---|---|---|
| + | + | 2 | 1 | 0 | 7 | 8 |

¿Cuál es el mayor número que podemos almacenar?
¿el menor positivo?

| | | | | | | |
|---|---|---|---|---|---|---|
| + | + | 9 | 9 | 9 | 9 | 9 |

$$0.9999 \times 10^9$$

| | | | | | | |
|---|---|---|---|---|---|---|
| + | - | 9 | 1 | 0 | 0 | 0 |

$$0.1000 \times 10^{-9}$$

$$1/34 = 0.029411765\ldots = \begin{cases} \cancel{0.0294 \times 10^0} \\ 0.2941 \times 10^{-1} \end{cases}$$

¡mantisa normalizada!

| | | | | | | |
|---|---|---|---|---|---|---|
| + | - | 1 | 2 | 9 | 4 | 1 |

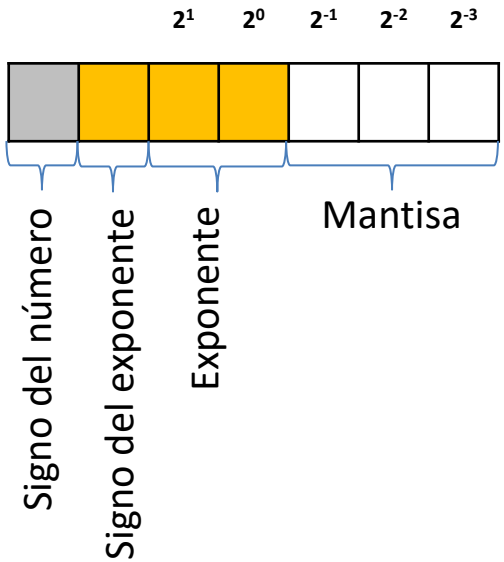
¿Qué pasó con el resto de los decimales?

¡Error de redondeo!

# Sistema de punto flotante



- En las computadoras el funcionamiento es similar, pero en base 2.




| | $2^1$ | $2^0$ | $2^{-1}$ | $2^{-2}$ | $2^{-3}$ |
|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 0 | 1 |

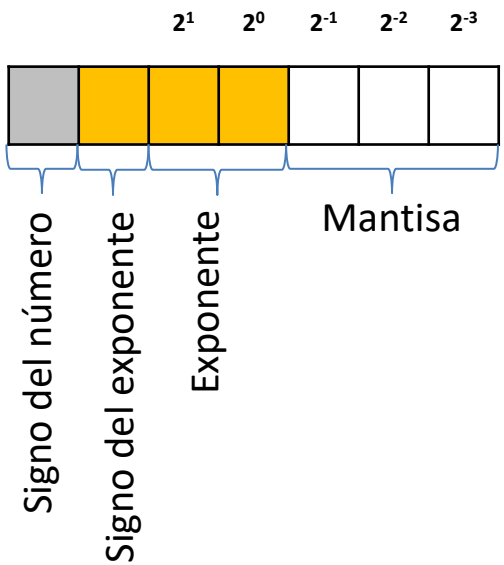
$$m = 1\times 2^{-1} + 0\times 2^{-2} + 1\times 2^{-3} = 0.625$$

$$e = -\left(1\times 2^{0} + 1\times 2^{1}\right) = -3$$

$$0.625\times 2^{-3} = 0.078125$$

# Sistema de punto flotante






Para pensar…

¿Cuál es el número más chico que se puede almacenar?

¿Cuál es el número más grande que se puede almacenar?

¡mantisa normalizada!

| | $2^1$ | $2^0$ | $2^{-1}$ | $2^{-2}$ | $2^{-3}$ |
|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 | 0 |

$$m = 1\times 2^{-1} + 0\times 2^{-2} + 0\times 2^{-3} = 0.5$$

$$e = -\left(1\times 2^{0} + 1\times 2^{1}\right) = -3$$

$$0.5\times 2^{-3} = 0.0625$$

| | $2^1$ | $2^0$ | $2^{-1}$ | $2^{-2}$ | $2^{-3}$ |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 1 |

$$m = 1\times 2^{-1} + 1\times 2^{-2} + 1\times 2^{-3} = 0.875$$

$$e = \left(1\times 2^{0} + 1\times 2^{1}\right) = 3$$

$$0.875\times 2^{3} = 7$$

# Sistema de punto flotante

| | $2^1$ | $2^0$ | $2^{-1}$ | $2^{-2}$ | $2^{-3}$ | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | $(0.0625)_{10}$ |

| | $2^1$ | $2^0$ | $2^{-1}$ | $2^{-2}$ | $2^{-3}$ | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | $(0.125000)_{10}$ |

| | $2^1$ | $2^0$ | $2^{-1}$ | $2^{-2}$ | $2^{-3}$ | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | $(0.078125)_{10}$ |

| | $2^1$ | $2^0$ | $2^{-1}$ | $2^{-2}$ | $2^{-3}$ | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | $(0.156250)_{10}$ |

| | $2^1$ | $2^0$ | $2^{-1}$ | $2^{-2}$ | $2^{-3}$ | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | $(0.093750)_{10}$ |

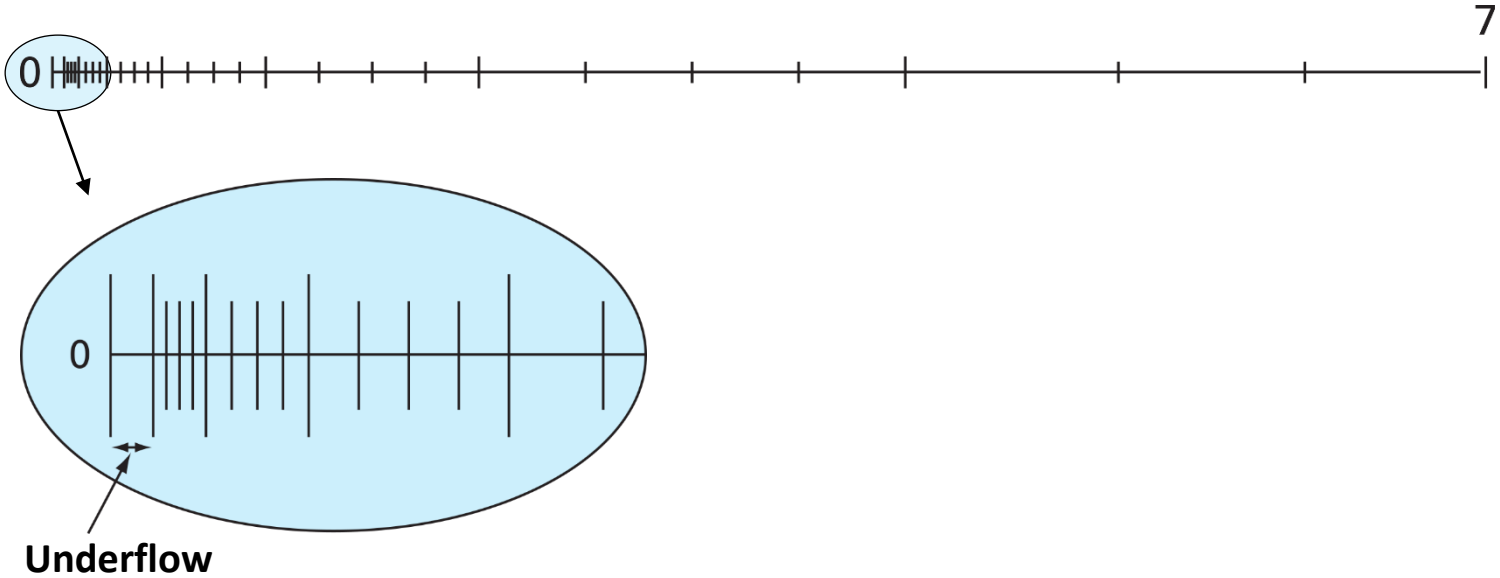
| | $2^1$ | $2^0$ | $2^{-1}$ | $2^{-2}$ | $2^{-3}$ | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | $(0.187500)_{10}$ |

| | $2^1$ | $2^0$ | $2^{-1}$ | $2^{-2}$ | $2^{-3}$ | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | $(0.109375)_{10}$ |

| | $2^1$ | $2^0$ | $2^{-1}$ | $2^{-2}$ | $2^{-3}$ | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | $(0.218750)_{10}$ |

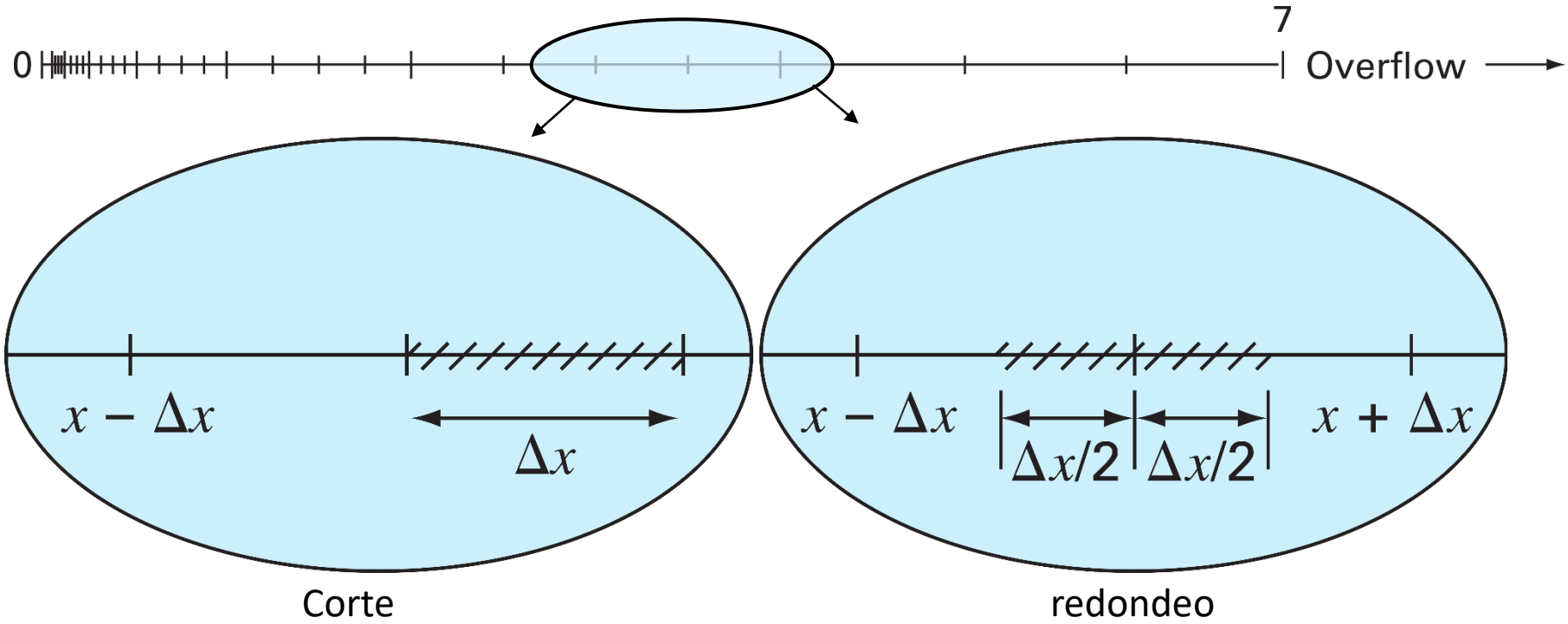
¡La recta numérica es discreta!

# Sistema de punto flotante



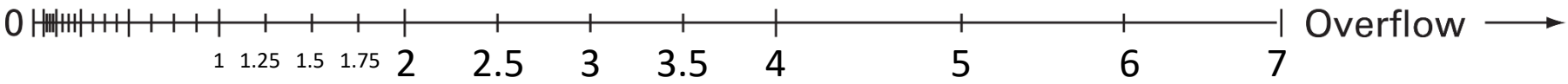

- El rango de cantidades que pueden representarse es limitado.
- Números fuera del rango dará como resultado el llamado error de desbordamiento (overflow) y números muy pequeños el denominado underflow.

# Sistema de punto flotante




- Los números irracionales no pueden representarse de manera exacta, a su vez, los números racionales que no concuerdan exactamente con uno de los valores en el conjunto tampoco pueden ser representados en forma precisa.
- A los errores ocasionados por la aproximación en ambos casos (corte o redondeo) se les conoce como errores de cuantificación.
- En general se utiliza el sistema de corte (menos costoso computacionalmente)

# Sistema de punto flotante



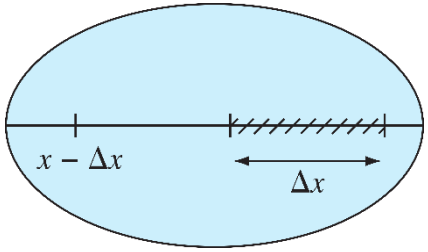

- El intervalo entre los números, Δx, aumenta conforme los números crecen en magnitud.
- Los errores de cuantificación son proporcionales a la magnitud del número que será representado.
- Esta proporcionalidad se expresa como:

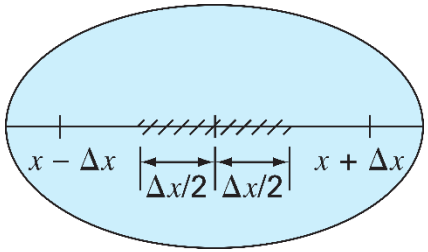
corte

$$\frac{|\Delta x|}{|x|} \leq \varepsilon$$

redondeo

$$\frac{|\Delta x|}{|x|} \leq \frac{\varepsilon}{2}$$



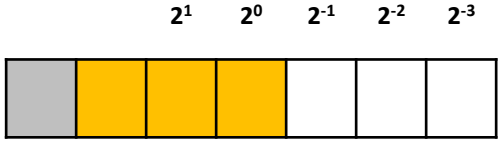

ε: épsilon de la máquina

$$\varepsilon = b^{1-t}$$

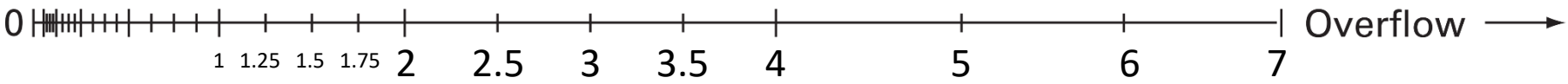
$b$: base

$t$: dígitos significativos en la mantisa

$$\varepsilon = 2^{1-3} = 0.25$$



# Ejemplo: $(4.75)_{10}$




$$4.75 = (100.11)_2 = (0.10011)_2 \times 2^3$$

| | | $2^1$ | $2^0$ | $2^{-1}$ | $2^{-2}$ | $2^{-3}$ |
|---|---|---|---|---|---|---|
| + | + | 1 | 1 | 1 | 0 | 0 |

$$(0.100)_2 \times 2^3 = 0.5 \times 2^3 = 4$$

$$4.75 \Rightarrow 4$$

corte

| | | $2^1$ | $2^0$ | $2^{-1}$ | $2^{-2}$ | $2^{-3}$ |
|---|---|---|---|---|---|---|
| + | + | 1 | 1 | 1 | 0 | 1 |

$$(0.101)_2 \times 2^3 = 0.625 \times 2^3 = 5$$

$$4.75 \Rightarrow 5$$

redondeo

# Ejemplo: $(4.75)_{10}$ con bit oculto

$$4.75 = (100.11)_2 = (0\boxed{1}0011)_2 \times 2^3 = (1.0011)_2 \times 2^2$$

¡Se aprovecha la mantisa normalizada!

| | | $2^1$ | $2^0$ | $2^{-1}$ | $2^{-2}$ | $2^{-3}$ |
|---|---|---|---|---|---|---|
| + | + | 1 | 0 | 0 | 0 | 1 |

$$(1.001)_2 \times 2^2 = 1.125 \times 2^2 = 4.5$$

$$4.75 \Rightarrow 4.5$$

corte

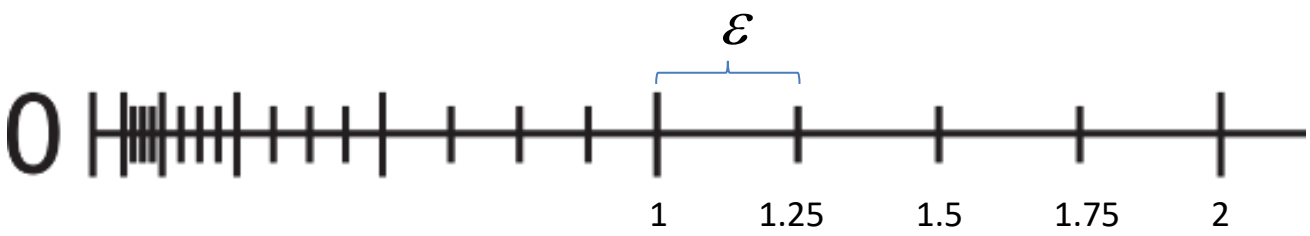
| | | $2^1$ | $2^0$ | $2^{-1}$ | $2^{-2}$ | $2^{-3}$ |
|---|---|---|---|---|---|---|
| + | + | 1 | 0 | 0 | 1 | 0 |

$$(1.010)_2 \times 2^2 = 1.25 \times 2^2 = 5$$

$$4.75 \Rightarrow 5$$

redondeo

# Sistema de punto flotante



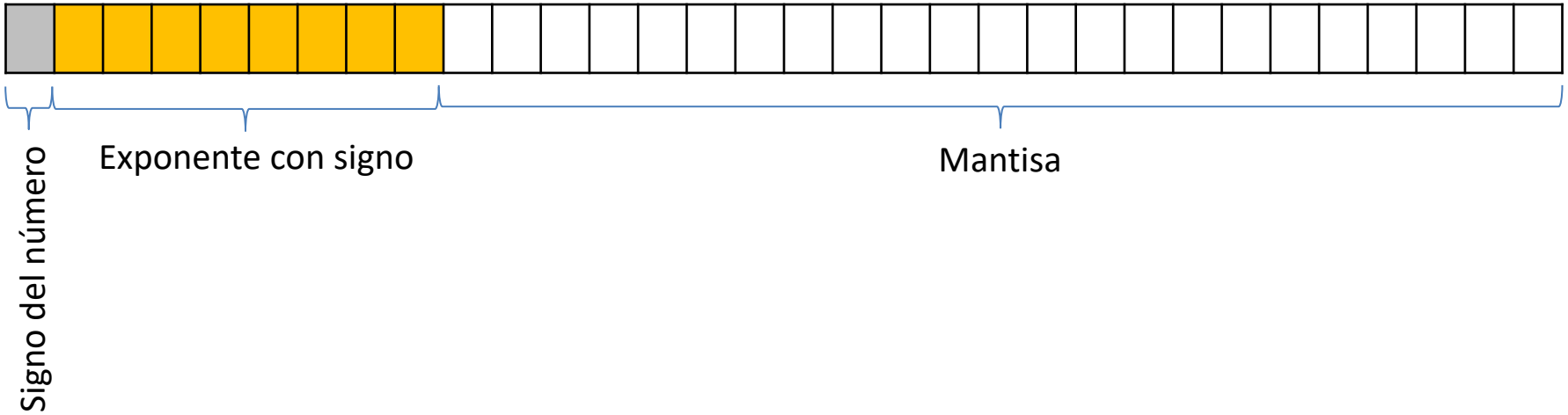

$$\forall \alpha \,/\, 0 < \alpha < \varepsilon \rightarrow 1 + \alpha = 1$$

- El sistema numérico hipotético (de 7 bits) que venimos analizando es una enorme exageración que se usó con propósitos ilustrativos.
- El número de dígitos significativos que tiene la mayoría de las computadoras permite que muchos cálculos de ingeniería se realicen con una precisión más que aceptable.

# Sistema de punto flotante (Precisión simple)

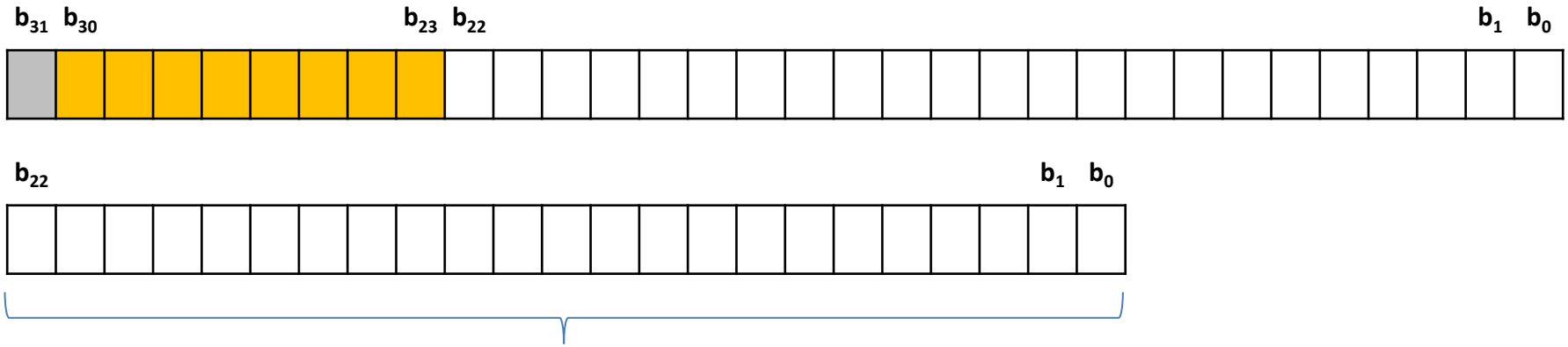
**Estándar IEEE 754 de 32 bits (bynary32).**

- 1 bit para el signo del numéro.
- Significando o Mantisa (T): 24 bits (23 almacenados explícitamente).
- 8 bits para el exponente.
- El exponente se codifica mediante una representación desplazada en binario, siendo **+127** el desplazamiento.




- Como es un sistema signo-magnitud, se tiene 1 bit de signo y 24 bits de magnitud (mantisa). De los bits de la magnitud, 1 está implícito y los otros 23 son los que se usan explícitamente. Entonces 23 bits son fraccionarios y el bit implícito es entero (e igual a 1).

# Sistema de punto flotante (Precisión simple)




$$m = (-1)^{b_{31}} (1.b_{22}b_{21}\ldots b_0)_2 = (-1)^{b_{31}} \left(1 + \sum_{i=1}^{23} b_{23-i}\, 2^{-i}\right)$$

- La mantisa es un numero fraccionario en base 2.
- La parte entera siempre es 1 (bit implícito) y la fraccionaria corresponde a los 23 bits almacenados.

# Sistema de punto flotante (Precisión simple)

Ejemplo:

| $b_{31}$ | $b_{30}$ | | | | | | | $b_{23}$ | $b_{22}$ | | | | | | | | | | | | | | | | | | | | | | | $b_1$ | $b_0$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

$$e = (01111100)_2 - 127$$

$$e = \left(0\times 2^0 + 0\times 2^1 + 1\times 2^2 + 1\times 2^3 + 1\times 2^4 + 1\times 2^5 + 1\times 2^6 + 0\times 2^7\right) - 127$$

$$e = 124 - 127 = -3$$

$$m = (-1)^0 (1.01000000000000000000000)_2$$

$$m = 1 + 0\times 2^{-1} + 1\times 2^{-2} + 0\times 2^{-3} + \ldots + 0\times 2^{-23} = 1.25$$

$$(N)_{10} = m\times 2^e = 1.25\times 2^{-3} = 0.15625$$

# Precisión simple: Ejemplos

| $b_{31}$ | $b_{30}$ | | | | | | | $b_{23}$ | $b_{22}$ | | | | | | | | | | | | | | | | | | | | | | $b_1$ | $b_0$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

$$(-1)^{b_{31}} \times (0.b_{22} \ldots b_0)_2 \times 2^{e_{\min}} = 5.87747 \times 10^{-39}$$

| $b_{31}$ | $b_{30}$ | | | | | | | $b_{23}$ | $b_{22}$ | | | | | | | | | | | | | | | | | | | | | | $b_1$ | $b_0$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

$$+\infty$$

| $b_{31}$ | $b_{30}$ | | | | | | | $b_{23}$ | $b_{22}$ | | | | | | | | | | | | | | | | | | | | | | $b_1$ | $b_0$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

$$-\infty$$

| $b_{31}$ | $b_{30}$ | | | | | | | $b_{23}$ | $b_{22}$ | | | | | | | | | | | | | | | | | | | | | | $b_1$ | $b_0$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

$$NaN$$

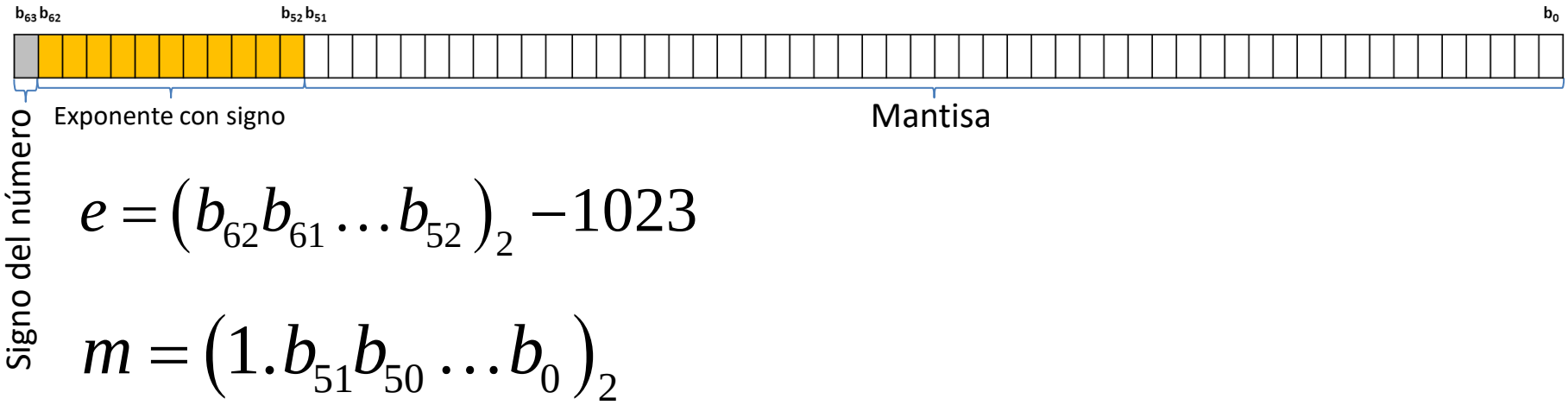
| $b_{31}$ | $b_{30}$ | | | | | | | $b_{23}$ | $b_{22}$ | | | | | | | | | | | | | | | | | | | | | | $b_1$ | $b_0$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

$$(-1)^{b_{31}} \times (1.b_{22} \ldots b_0)_2 \times 2^{e} = (1.1)_2 \times 2^{129-127} = 1.5 \times 2^2 = 6$$

# Sistema de punto flotante (Precisión doble)

**Estándar IEEE 754 de 64 bits (bynary64).**

- 1 bit para el signo del numéro.
- Significando o Mantisa (T): 53 bits (52 almacenados explícitamente).
- 11 bits para el exponente.
- El exponente se codifica mediante una representación desplazada en binario, siendo **+1023** el desplazamiento.




$$e = \left(b_{62}b_{61}\ldots b_{52}\right)_2 - 1023$$

$$m = \left(1.b_{51}b_{50}\ldots b_0\right)_2$$

- Como es un sistema signo-magnitud, se tiene 1 bit de signo y 53 bits de magnitud (mantisa). De los bits de la magnitud, 1 está implícito y los otros 52 son los que se usan explícitamente. Entonces 52 bits son fraccionarios y el bit implícito es entero (e igual a 1).

# Sistema de punto flotante (Resumen)

| | **binary32** | **binary64** |
|---|---|---|
| Bits totales | 32 | 64 |
| Bits de la mantisa | 24 | 53 |
| Bits del exponente | 8 | 11 |
| Mínimo exponente | -126 | -1022 |
| Máximo exponente | 127 | 1023 |
| Real min | $1.1755 \times 10^{-38}$ | $2.225 \times 10^{-308}$ |
| Real max | $3.40282 \times 10^{38}$ | $1.797 \times 10^{308}$ |
| Épsilon | $1.192 \times 10^{-7}$ | $2.22 \times 10^{-16}$ |

# Errores de truncamiento: Serie de Taylor



- Los errores de truncamiento son aquellos que resultan del uso de una aproximación en lugar de un procedimiento matemático exacto.
- El teorema de Taylor establece que cualquier función uniforme puede aproximarse mediante un polinomio y su serie proporciona un medio para expresar esta idea matemáticamente.
- Un caso frecuente es usar series de Taylor para predecir el valor de una función en un punto en términos del valor de la función y sus derivadas en otro punto

$$f\left(x\right)=\sum_{n=0}^{\infty}\frac{f^{n}\left(x_{0}\right)}{n!}\left(x-x_{0}\right)^{n}$$

# Errores de truncamiento: Serie de Taylor



$$f\left(x\right)=\sum_{n=0}^{\infty}\frac{f^{n}\left(x_{0}\right)}{n!}\left(x-x_{0}\right)^{n}$$

$$f(x)=f(x_0)+f^{'}(x_0)(x-x_0)+\frac{f^{''}(x_0)}{2!}(x-x_0)^2+\ldots$$

$$\ldots+\frac{f^{'''}(x_0)}{3!}(x-x_0)^3+\frac{f^{''''}(x_0)}{4!}(x-x_0)^4+\ldots$$

# Errores de truncamiento: Serie de Taylor



Aproximación constante

$$f(x) = f(x_0)$$

Aproximación lineal

$$f(x) = f(x_0) + f'(x_0)(x - x_0)$$

Aproximación cuadrática

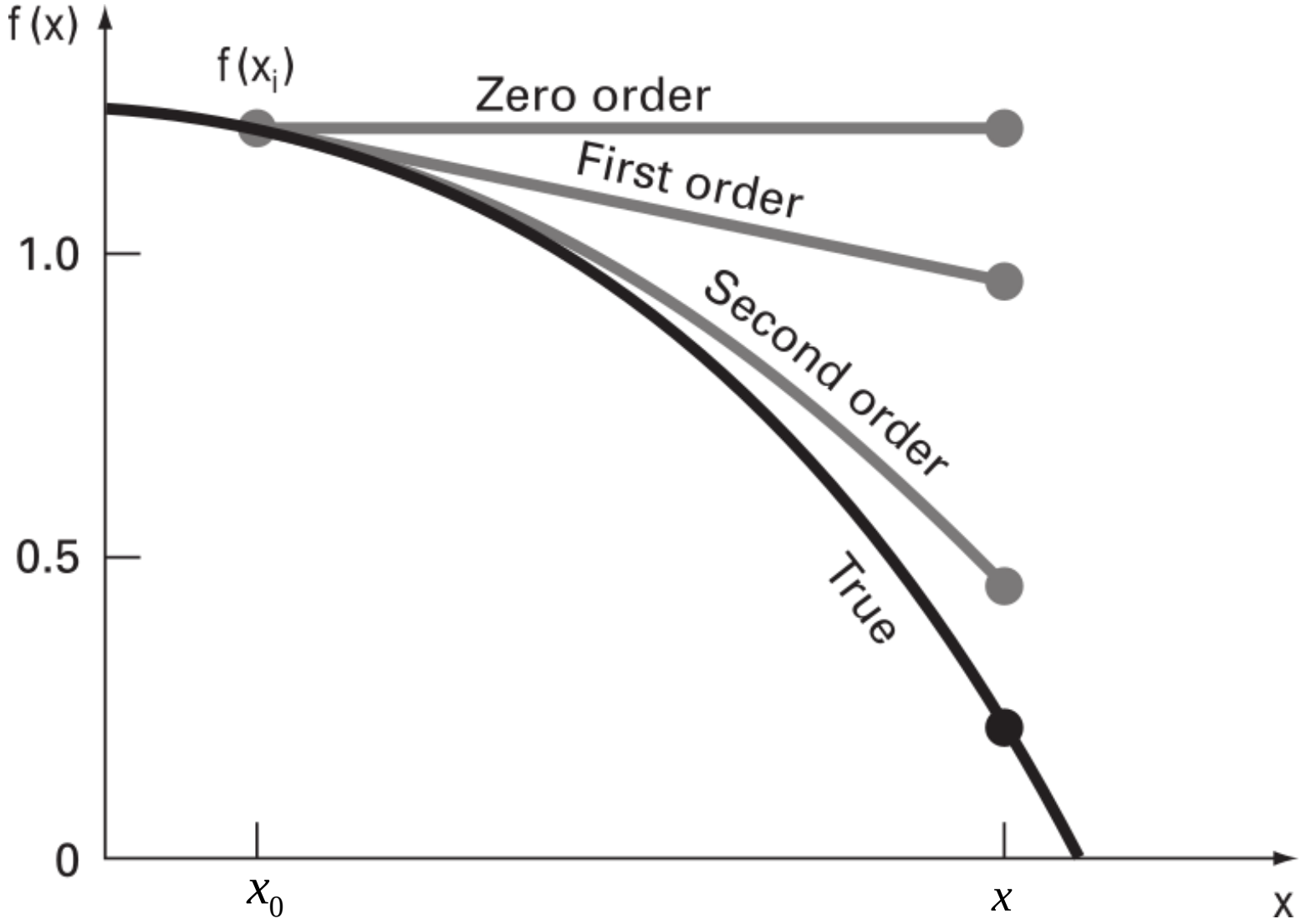
$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2$$

Forma General

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \ldots + \frac{f^n(x_0)}{n!}(x - x_0)^n + R_n$$

$$R_n = \frac{f^{n+1}(\xi)}{n+1!}(x - x_0)^{n+1}$$

# Errores de truncamiento: Serie de Taylor



# Serie de Maclaurin ($x_0=0$)

$$f(x)=\sum_{n=0}^{\infty}\frac{f^{(n)}(x_0)}{n!}(x-x_0)^n=\sum_{n=0}^{\infty}\frac{f^{(n)}(0)}{n!}(x-0)^n \Rightarrow f(x)=\sum_{n=0}^{\infty}\frac{f^{(n)}(0)}{n!}x^n$$

$$f(x)=f(0)+f'(0)x+\frac{f''(0)}{2!}x^2+\ldots$$

$$\ldots+\frac{f'''(0)}{3!}x^3+\frac{f''''(0)}{4!}x^4+\ldots$$

# Serie de Maclaurin: seno(x)

$$f(x)=\sin(x) \qquad f'(x)=\cos(x) \qquad f''(x)=-\sin(x) \qquad f'''(x)=-\cos(x)$$

$$f''''(x)=\sin(x) \quad f'''''(x)=\cos(x) \quad f''''''(x)=-\sin(x) \quad f'''''''(x)=-\cos(x)$$

$$f''''''''(x)=\sin(x)\ldots$$

$$f(x)=f(0)+f'(0)x+\frac{f''(0)}{2!}x^2+\frac{f'''(0)}{3!}x^3+\frac{f''''(0)}{4!}x^4+\ldots \qquad \begin{cases}\sin(0)=0\\ \cos(0)=1\end{cases}$$

$$\sin(x)=0+1x-\frac{0}{2!}x^2-\frac{1}{3!}x^3+\frac{0}{4!}x^4+\ldots$$

$$\sin(x)=x-\frac{x^3}{3!}+\frac{x^5}{5!}-\frac{x^7}{7!}+\frac{x^9}{9!}-\frac{x^{11}}{11!}+\ldots$$

# Serie de Maclaurin: seno(x)

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \frac{x^{11}}{11!} + \frac{x^{13}}{13!} - \ldots$$

**¡No podemos utilizar infinitos términos!**

$$\sin\left(\frac{\pi}{2}\right) = 1$$

| **x=1.570796327** | | **1** |
|---|---|---|
| Términos | Aproximación del sin(x) | Error porcentual |
| 1 | 1.5707963267949 | 57.08% |
| 3 | 0.9248322292887 | 7.52% |
| 5 | 1.0045248555348 | 0.45% |
| 7 | 0.9998431013995 | 0.02% |
| 9 | 1.0000035425843 | 0.00% |
| 11 | 0.9999999437411 | 0.00% |
| 13 | 1.0000000006628 | 0.00% |

Valor exacto

Términos de la serie

$\rightarrow 6.63 \times 10^{-8}$

# Serie de Maclaurin: $e^x$ o exp(x)

$$f(x) = f'(x) = f'(x) = \ldots = f^{n}(x) = \exp(x)$$

$$f(x) = f(0) + f'(0)x + \frac{f''(0)}{2!}x^2 + \frac{f'''(0)}{3!}x^3 + \frac{f''''(0)}{4!}x^4 + \ldots \qquad \exp(0) = 1$$

$$\exp(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \ldots$$

$$\exp(x) = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

| x=1 | | 2.7182818284591 |
|---|---|---|
| Términos | Aproximación de exp(x) | Error porcentual |
| 0 | 1.0000000000000 | 63.21% |
| 1 | 2.0000000000000 | 26.42% |
| 2 | 2.5000000000000 | 8.03% |
| 3 | 2.6666666666667 | 1.90% |
| 4 | 2.7083333333333 | 0.37% |
| 5 | 2.7166666666667 | 0.06% |
| 6 | 2.7180555555556 | 0.01% |
| 7 | 2.7182539682540 | 0.00% |
| 8 | 2.7182787698413 | 0.00% |
| 9 | 2.7182815255732 | 0.00% |
| 10 | 2.7182818011464 | 0.00% |

$1.00 \times 10^{-6} \leftarrow$

# Serie de Maclaurin: $e^x$ o exp(x)

| x=1 | | |
|---|---|---|
| Términos | Aproximación de exp(x) | Error aprox. |
| 0 | 1.0000000000000 | |
| 1 | 2.0000000000000 | 50.00% |
| 2 | 2.5000000000000 | 20.00% |
| 3 | 2.6666666666667 | 6.25% |
| 4 | 2.7083333333333 | 1.54% |
| 5 | 2.7166666666667 | 0.31% |
| 6 | 2.7180555555556 | 0.05% |
| 7 | 2.7182539682540 | 0.01% |
| 8 | 2.7182787698413 | 0.00% |
| 9 | 2.7182815255732 | 0.00% |
| 10 | 2.7182818011464 | 0.00% |

$$\leftarrow \varepsilon_a = \frac{f_1(1) - f_0(1)}{f_1(1)} \times 100 = \frac{2-1}{2} \times 100 = 50\%$$

$$\rightarrow 1.01 \times 10^{-5}$$

Los sucesivos valores convergen al valor exacto

# Resumen



## Errores de redondeo

Se deben a que la computadora tan sólo representa cantidades con un número finito de dígitos.

## Errores de truncamiento

Se deben a que los métodos numéricos pueden emplear aproximaciones para representar operaciones y cantidades matemáticas exactas.

**Trabajar en doble precisión ayuda a minimizar estos errores.**

**La mayoría de los softwares utilizados trabajan por defecto en doble precisión**