

# Cálculo numérico 1

Formulario · Primavera 2021

Carlos Lezama MAT - 14400

ITAM

Fundamentos

0.1. Aritmética de punto flotante

Un número de punto flotante consiste en tres partes:

- 1. el **signo** (+ o −);
- 2. la **mantisa**, que contiene la cadena de bits significativos;
- 3. el **exponente**.

±

d<sub>1</sub>

d<sub>2</sub>

⋯

d<sub>m−1</sub>

d<sub>m</sub>

s

i.e.  $\pm 0.d_1d_2 \dots d_{m-1}d_m \times 10^s, \quad m \in \mathbb{N}, s \in \mathbb{Z}.$

Formatos y precisión

Precisión	Signo	Mantisa ( <i>m</i> )	Exponente ( <i>s</i> )	Bits totales
<i>half</i>	1	10	5	16
<i>single</i>	1	23	8	32
<i>double</i>	1	52	11	64
<i>long double</i>	1	64	15	80
<i>quad</i>	1	112	15	128

Proposición 0.1

El número positivo más grande está dado por:

$$X_{\text{máx}} = (1 - 10^{-m}) 10^{E_{\text{máx}}},$$

donde  $E_{\text{máx}}$  es el exponente entero positivo más grande.

Proposición 0.2

El número positivo más pequeño está dado por:

$$X_{\text{mín}} = 10^{E_{\text{mín}}-1},$$

donde  $E_{\text{mín}}$  es el exponente entero negativo más pequeño.

**Definición 0.1** (Método de corte). *Sea algún número real*

$$x = (0.d_1d_2 \dots d_md_{m+1} \dots) \times 10^s,$$

$$fl(x) = (0.d_1d_2 \dots d_m) \times 10^s.$$

**Definición 0.2** (Método de redondeo). *Sea algún número real*

$$x = (0.d_1d_2 \dots d_md_{m+1} \dots) \times 10^s,$$

$$fl(x) = \begin{cases} (0.d_1d_2 \dots d_m) \times 10^s, & \text{si } d_{m+1} < 5 \\ (0.d_1d_2 \dots d_m) \times 10^s + (0.0 \dots 01) \times 10^s, & \text{si } d_{m+1} \geq 5 \end{cases}.$$

Teorema 0.1

Al usar el método de corte, para toda  $x \in [X_{\text{mín}}, X_{\text{máx}}]$  se cumple:

$$\frac{|x - fl(x)|}{|x|} \leq 10^{1-m}.$$

**Definición 0.3** (Operación suma). *Definimos el algoritmo de la suma como sigue:*

**Input:**  $x, y$   
**Output:**  $\tilde{z}$

1  $\tilde{x} \leftarrow fl(x);$   
2  $\tilde{y} \leftarrow fl(y);$   
3  $z \leftarrow \tilde{x} + \tilde{y};$   
4  $\tilde{z} \leftarrow fl(z)$

**Definición 0.4** (Épsilon de máquina). *La precisión numérica es el primer número positivo  $\varepsilon$  tal que:*

$$1 \oplus \varepsilon > 1.$$

## 1. Localización de raíces y extremos locales

**Input:**  $a, b, k_{\text{máx}}, \text{TOL}$   
**Output:**  $c, k$

```
1  $k \leftarrow 0$ ;  
2 while  $|b - a| > \text{TOL}$  and  $k < k_{\text{máx}}$  do  
3    $c \leftarrow (a + b)/2$ ;  
4   if  $f(c) = 0$  then  
5     stop  
6   end  
7   if  $f(a)f(b) < 0$  then  
8      $b \leftarrow c$ ;  
9   else  
10     $a \leftarrow c$ ;  
11  end  
12   $k \leftarrow k + 1$ ;  
13 end
```

**Algoritmo 1:** Método de bisección

**Input:**  $x_0, k_{\text{máx}}, \text{TOL}$   
**Output:**  $k, x_k$

```
1  $k \leftarrow 0$ ;  
2 while  $|g(x_k)| > \text{TOL}$  and  $k < k_{\text{máx}}$  do  
3    $x_{k+1} \leftarrow g(x_k)$ ;  
4    $k \leftarrow k + 1$ ;  
5 end
```

**Algoritmo 2:** Iteración de punto fijo

**Input:**  $x_0, k_{\text{máx}}, \text{TOL}$   
**Output:**  $k, x_k$

```
1  $k \leftarrow 0$ ;  
2 while  $|f(x_k)| > \text{TOL}$  and  $k < k_{\text{máx}}$  do  
3    $x_{k+1} \leftarrow x_k - \frac{f(x_k)}{f'(x_k)}$ ;  
4    $k \leftarrow k + 1$ ;  
5 end
```

**Algoritmo 3:** Método de Newton

**Input:**  $x_0, x_1, k_{\text{máx}}, \text{TOL}$   
**Output:**  $k, x_k$

```
1  $k \leftarrow 1$ ;  
2 while  $|f(x_k)| > \text{TOL}$  and  $k < k_{\text{máx}}$  do  
3    $x_{k+1} \leftarrow x_k - \frac{f(x_k)(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})}$ ;  
4    $k \leftarrow k + 1$ ;  
5 end
```

**Algoritmo 4:** Método de la secante

**Input:**  $a, b, k_{\text{máx}}, \text{TOL}$   
**Output:**  $c, k$

```
1  $k \leftarrow 0$ ;  
2 while  $|b - a| > \text{TOL}$  and  $k < k_{\text{máx}}$  do  
3    $c \leftarrow \frac{bf(a) - af(b)}{f(a) - f(b)}$ ;  
4   if  $f(c) = 0$  then  
5     stop  
6   end  
7   if  $f(a)f(b) < 0$  then  
8      $b \leftarrow c$ ;  
9   else  
10     $a \leftarrow c$ ;  
11  end  
12   $k \leftarrow k + 1$ ;  
13 end
```

**Algoritmo 5:** Regula Falsi