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**.

$\pm$	d_1	d_2	$\cdots$	d_{m-1}	d_m	s

$$\text{i.e.} \quad \pm 0.d_1 d_2 \ldots d_{m-1} d_m \times 10^s, \quad m \in \mathbb{N},\ s \in \mathbb{Z}.$$

Formatos y precisión

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

Proposición 0.1

El número positivo más grande está dado por:

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

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*

$$x = (0.d_1 d_2 \ldots d_m d_{m+1} \ldots) \times 10^s,$$

$$fl(x) = (0.d_1 d_2 \ldots d_m) \times 10^s.$$

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

$$x = (0.d_1 d_2 \ldots d_m d_{m+1} \ldots) \times 10^s,$$

$$fl(x) = \begin{cases} (0.d_1 d_2 \ldots d_m) \times 10^s, & \text{si } d_{m+1} < 5 \\ (0.d_1 d_2 \ldots d_m) \times 10^s + (0.0 \ldots 01) \times 10^s, & \text{si } d_{m+1} \geq 5 \end{cases}.$$

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: $\tilde{z}$

$\tilde{x} \longleftarrow fl(x)$;
$\tilde{y} \longleftarrow fl(y)$;
$z \longleftarrow \tilde{x} + \tilde{y}$;
$\tilde{z} \longleftarrow fl(z)$

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

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

1. Localización de raíces y extremos locales

Input: a, b, $k_{\text{máx}}$, TOL
Output: c, k
$k \longleftarrow 0$;
while $|b-a| > \text{TOL}$ **and** $k < k_{\text{máx}}$ **do**
    $c \longleftarrow (a+b)/2$;
    if $f(c) = 0$ **then**
        stop
    end
    if $f(a)f(b) < 0$ **then**
        $b \longleftarrow c$;
    else
        $a \longleftarrow c$;
    end
    $k \longleftarrow k+1$;
end

Algoritmo 1: Método de bisección

Input: x_0, $k_{\text{máx}}$, TOL
Output: k, x_k
$k \longleftarrow 0$;
while $|g(x_k)| > \text{TOL}$ **and** $k < k_{\text{máx}}$ **do**
    $x_{k+1} \longleftarrow g(x_k)$;
    $k \longleftarrow k+1$;
end

Algoritmo 2: Iteración de punto fijo

Input: x_0, $k_{\text{máx}}$, TOL
Output: k, x_k
$k \longleftarrow 0$;
while $|f(x_k)| > \text{TOL}$ **and** $k < k_{\text{máx}}$ **do**
    $x_{k+1} \longleftarrow x_k - \dfrac{f(x_k)}{f'(x_k)}$;
    $k \longleftarrow k+1$;
end

Algoritmo 3: Método de Newton

Input: x_0, x_1, $k_{\text{máx}}$, TOL
Output: k, x_k
$k \longleftarrow 1$;
while $|f(x_k)| > \text{TOL}$ **and** $k < k_{\text{máx}}$ **do**
    $x_{k+1} \longleftarrow x_k - \dfrac{f(x_k)(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})}$;
    $k \longleftarrow k+1$;
end

Algoritmo 4: Método de la secante

Input: a, b, $k_{\text{máx}}$, TOL
Output: c, k
$k \longleftarrow 0$;
while $|b-a| > \text{TOL}$ **and** $k < k_{\text{máx}}$ **do**
    $c \longleftarrow \dfrac{bf(a) - af(b)}{f(a) - f(b)}$;
    if $f(c) = 0$ **then**
        stop
    end
    if $f(a)f(b) < 0$ **then**
        $b \longleftarrow c$;
    else
        $a \longleftarrow c$;
    end
    $k \longleftarrow k+1$;
end

Algoritmo 5: Regula Falsi