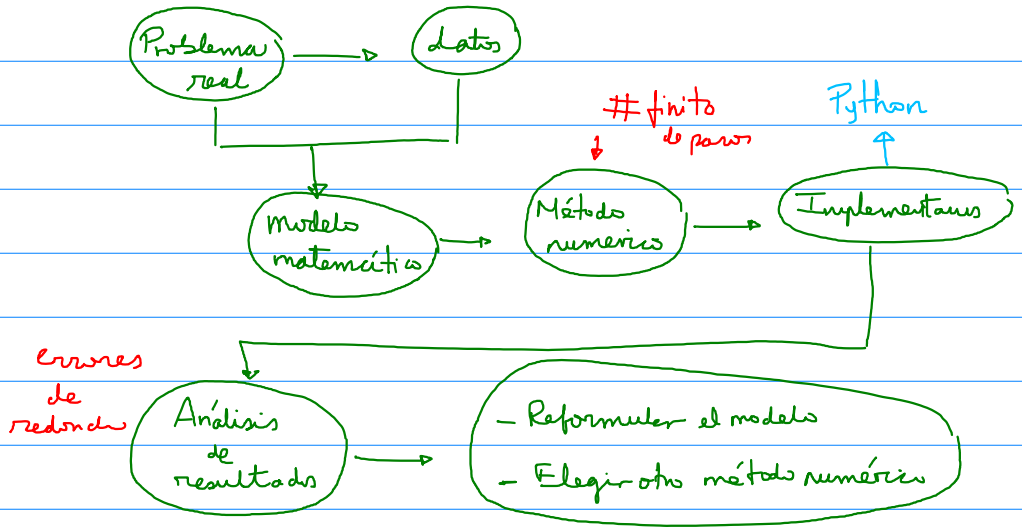
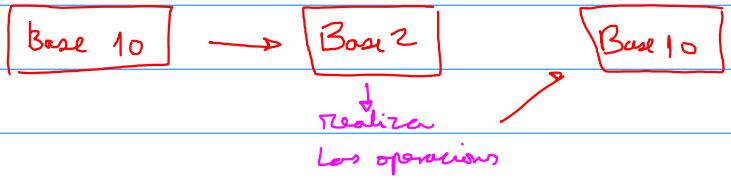
Fuentes de errores





Interacción entre el usuario y el computador → Errores de redondeo.

1. Conversión del sistema binario al decimal.

Base: $\beta \geq 2$

Número en base β: $(a_j a_{j-1} \dots a_2 a_1 a_0)_\beta$, $0 \leq a_k \leq \beta - 1$, $k = 1, \dots, j$

$= a_j \beta^j + a_{j-1} \beta^{j-1} + \dots + a_2 \beta^2 + a_1 \beta^1 + a_0 \beta^0$ → representación en base 10.

Ejm

$(1011)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 8 + 0 + 2 + 1 = 11 = (11)_{10}$ 12 operaciones

$= 2(2(1 \times 2 + 0) + 1) + 1$ 6 operaciones

$(a_3 a_2 a_1 a_0)_2 = (1011)_2$

$= 2(2(a_3 \times 2 + a_2) + a_1) + a_0$ operamos de adentro hacia afuera

$b_3 = a_3$ $n = 3$

$b_2 = 2 \times b_3 + a_2$

$b_1 = 2 \times b_2 + a_1$

$b_0 = 2 \times b_1 + a_0$

→ devolvemos b_0

$a = [a_0\ a_1 \dots a_n]$

$b = [b_0\ b_1 \dots b_n]$

$b_n = a_n$

For $i = 1:n$

$\{\ b_{n-i} = 2b_{n+1-i} + a_{n-i}\ \}$

End

devolvemos b_0

Conversión del sistema decimal al binario

$N_0 = 11$ y $(a_j a_{j-1} \dots a_1 a_0)_2$ representación binaria de N_0

$$11 = a_j 2^j + a_{j-1} 2^{j-1} + \dots + a_1 \times 2 + a_0.$$

$$2\times 5 + 1 = 2(a_j 2^{j-1} + a_{j-1} 2^{j-2} + \dots + a_2 \times 2 + a_1) + a_0$$

$\Rightarrow$ $a_0 = 1$ representa el residuo de dividir N_0 por 2.

división entera

$N_1 = 5 = a_j 2^{j-1} + a_{j-1} 2^{j-2} + \dots + a_2 \times 2 + a_1$

$\hookrightarrow 2\times 2 + 1$

a_1 es el resto de dividir N_1 por 2 $\quad N_1 = 2\times 2 + 1 \rightarrow a_1 = 1$

$N_2 = 2 \rightarrow N_2 = 2\times 1 + 0 \rightarrow a_2 = 0$

$N_3 = 1 \rightarrow N_3 = 2\times 0 + 1 \rightarrow a_3 = 1$ (FIN) $\leftarrow$

$(a_3 a_2 a_1 a_0)_2 = (1011)_2$

Python

a div b	cociente de la división entera	$a // b$ ó .math.floor
a mod b	el residuo de la división entera	$a \% b$

Ejm 13 div $5 = 2$, 13 mod $5 = 3$ $\}$ $13 = 2\times 5 + 3$

$i = 0$

while (N div $2 \neq 0$)

{ $a_i = N$ mod 2

$N = N$ div 2

$i = i + 1$ } devuelve $(a_i\ a_{i-1} \dots a_2 a_1 a_0)_2$.

$a_i = N$ mod 2

Número fraccionario en base 10 a base 2

$r \in (0,1)$ y su representación en base 2 es $(0.d_1 d_2 \ldots d_j \ldots)_2$

Ejm

$$0.125 = d_1 \times 2^{-1} + d_2 \times 2^{-2} + \ldots + d_j \times 2^{-j} + \ldots$$

$\times 2$ ↓ (parte entera)

parte fraccionaria: $0.250 = d_1 + d_2 \times 2^{-1} + d_3 \times 2^{-2} + \ldots + d_j \times 2^{-j+1} + \ldots \rightarrow d_1 = 0$

$\times 2$ ↓

$$0.5 = d_2 + d_3 \times 2^{-1} + \ldots + d_j \times 2^{-j+2} + \ldots \rightarrow d_2 = 0$$

$\times 2$ ↓

$$1.0 = d_3 + \ldots + d_j \times 2^{-j+3} + \ldots \rightarrow d_3 = 1$$

La parte fraccionaria es 0. FIN.

$\rightarrow 0{,}125 = (0.001)_2$.

Ejm2 $r = 0.1 = (0.d_1 d_2 \ldots d_j \ldots)_2$

r_k

$$r_1 = 0.1 = d_1 \times 2^{-1} + d_2 \times 2^{-2} + \ldots + d_j \times 2^{-j} + \ldots$$

$\times 2$ ↓

$$r_2 = 0.2 = d_1 + d_2 \times 2^{-1} + \ldots + d_j \times 2^{-j+1} + \ldots \rightarrow d_1 = 0$$

$\times 2$ ↓

$$r_3 = 0.4 = d_2 + d_3 \times 2^{-1} + \ldots + d_j \times 2^{-j+2} + \ldots \rightarrow d_2 = 0$$

$\times 2$ ↓

$$r_4 = 0.8 = d_3 + d_4 \times 2^{-1} + \ldots + d_j \times 2^{-j+3} + \ldots \rightarrow d_3 = 0$$

$\times 2$ ↓

$$1.6 = d_4 + d_5 \times 2^{-1} + \ldots + d_j \times 2^{-j+4} + \ldots \rightarrow d_4 = 1$$

$$r_5 = 0.6 = d_5 \times 2^{-1} + \ldots + d_j \times 2^{-j+4} + \ldots$$

$\times 2$ ↓

$$1.2 = d_5 + d_6 \times 2^{-1} + \ldots + d_j \times 2^{-j+5} + \ldots \rightarrow d_5 = 1$$

$$r_6 = 0.2 = d_6 \times 2^{-1} + \ldots + d_j \times 2^{-j+5} + \ldots$$

$\times 2$ ↓

$$r_7 = 0.4 = d_6 + d_7 \times 2^{-1} + \ldots + d_j \times 2^{-j+6} + \ldots \rightarrow d_6 = 0$$

$\vdots$

Los coeficientes se repiten desde $k = 2$ a 5.

$r_{10} = r_6 = r_2 = 0.2$ y así sucesivamente

$(0.1)_{10} = (0.00011\ 0011\ 0011\ \ldots)_2$

almacenar una aproximación en el computador

El resultado final no es exacto.

* Número fraccionario en base 2 a base 10

Ejm

$r_1 = (0.001)_2 = (0.b_1 b_2 \ldots b_j)_{10}$ (×10 / ×10)

en base

$w_1 = (1010)_2 \times (0.001)_2 = (b_1.b_2 b_3 \ldots b_j)_{10}$

Multiplicación en binario

$0 \times 0 = 0$	$0 + 0 = 0$
$0 \times 1 = 0$	$0 + 1 = 1$
$1 \times 0 = 0$	$1 + 0 = 1$
$1 \times 1 = 1$	$1 + 1 = (2) \rightarrow 0 \rightarrow 1$ (unidad suponer)

$$\begin{array}{r} 1010 \times \\ 0001 \\ \hline 1,010 \end{array}$$

$(1010)_2 \times (0.001)_2 = (b_1.b_2 b_3 \ldots b_j)_{10}$

$(1.01)_2$

$1 + (0.01)_2 = (b_1.b_2 b_3 \ldots b_j)_{10} \rightarrow b_1 = 1$

$r_2 = (0.01)_2 = (0, b_2 b_3 \ldots b_j)_{10}$

$w_2 = (1010)_2 \times (0.01)_2 = (b_2.b_3 b_4 \ldots b_j)_{10}$

$(10.10)_2 = (b_2.b_3 - \ldots b_j)_{10}$

$(10)_2 + (0.10)_2$

$2 + (0.10)_2 = (b_2.b_3 - \ldots b_j)_{10} \rightarrow b_2 = 2$

$r_3 = (0.10)_2 = 0.b_3 \ldots b_j$

$w_3 = (1010)_2 \times (0,1)_2 = b_3.b_4 \dots b_j$

$5 = (101)_2 = b_3.b_4 \dots b_j \rightarrow b_3 = 5$ FIN.

$(0,0,1)_2 = (0.125)_{10}$ ✓

i) Representación posicional de x con respecto a β.

• Base: $\beta \in \mathbb{N}$, $\beta \geq 2$

• $x \in \mathbb{R}$ con un número finito de dígitos x_k, $0 \leq x_k \leq \beta - 1$, $k = -m, \dots, n$

Tenemos $n+m+1$ dígitos.

$$x_\beta = (-1)^{\Delta} \left[x_n x_{n-1} \cdots x_1 x_0 \bullet x_{-1} x_{-2} \cdots x_{-m} \right], \quad x_n \neq 0$$

(punto decimal)

$$= (-1)^{\Delta} \left[x_n \beta^n + \cdots x_1 \beta + x_0 + x_{-1} \beta^{-1} + x_{-2} \beta^{-2} + \cdots + x_{-m} \beta^{-m} \right]$$

$$= (-1)^{\Delta} \sum_{k=-m}^{n} x_k \beta^k$$

• Si $x \in \mathbb{R}$ tiene un número infinito de dígitos.

r dígitos

$$x_\beta = x_n x_{n-1} . x_* \, x_1 x_0, x_{-1} x_{-2} \dots x_{-m} \dots$$

Truncar en $r \geq 1$ dígitos.

$r < n$: $= x_n x_{n-1} \cdots x_{n-r+1} 0 \dots 0, 0 \dots 0 = \sum_{k=0}^{r-1} x_{n-k} \beta^{n-k}$

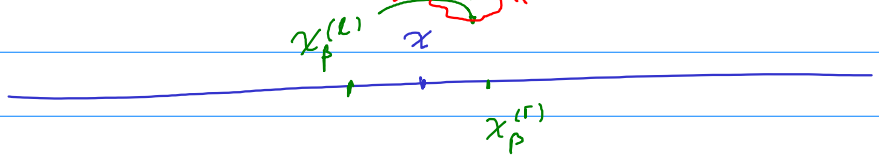
$x_\beta^{(l)}$

$r \geq n$ $= x_n x_{n-1} \cdots x_1 x_0, x_{-1} \cdots x_* \, 0 0 \dots$

x_{n-r+1}

Truncar r decimales: $x_n x_{n-1} \dots x_1 x_0, x_{-1} x_{-2} \dots x_{n-r+1} \, 0 0 \dots$

$x_\beta^{(l)}$ tiene al menos r dígitos posiblemente no nulos.



sumar 1 al final del dígito posiblemente no nulo.

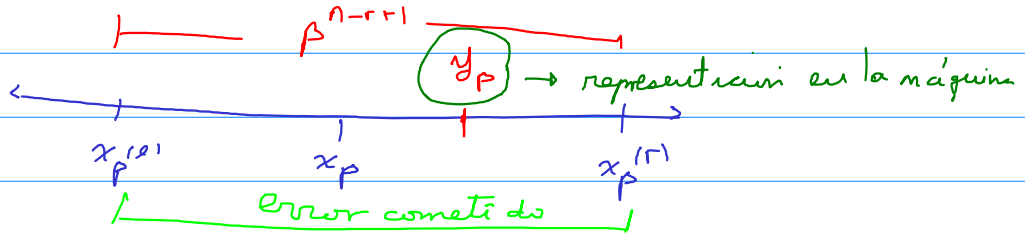
$0\ 0 \dots 0\ 1 \equiv 1 \times \beta^{n-r+1}$

posiciones: 1, 2, $r-1$, r

$$x_\beta^{(r)} = x_\beta^{(l)} + 1 \times \beta^{n-r+1}$$

, donde hay $n+1$ dígitos enteros y r el número de dígitos del truncamiento.

$$x_\beta^{(l)} \leq x_\beta \leq x_\beta^{(r)}$$



Si $y_\beta \in [x_\beta^{(l)}, x_\beta^{(r)}] \Rightarrow |y_\beta - x_\beta| \leq |x_\beta^{(r)} - x_\beta^{(l)}| = \beta^{n-r+1}$

deseo controlarlo con el parámetro ε

Encontrar un número de dígitos r suf. grande t.q. $\beta^{n-r+1} < \varepsilon$

$\therefore \forall \varepsilon > 0, \forall x_\beta, \exists\, y_\beta$ (depende ε) tal que $|y_\beta - x_\beta| < \varepsilon$

↓

debe tener un número finito de dígitos (máximo r)