Introdução ao Cálculo Numérico

Sistemas de numeração e conversão de base

Instituto de MATEMÁTICA E ESTATÍSTICA
UFRGS
Prof. Rudnei Dias da Cunha

- Existem basicamente dois **sistemas de numeração** utilizados para representar números:
  - Não-posicional: romano
  - Posicional: indo-arábico

- No **sistema romano**, os números são representados por uma sequência de letras (I, V, X, L, C, M, D) que, dependendo da **ordem** em que se encontram, podem representar quantidades diferentes:
  - **XLVIII** (=48; o X e o L juntos representam o valor 40)
  - **XXVIII** (=28; os dois X iniciais representam 20)
  - Uma letra só pode ser repetida até 3 vezes
- **Não há a noção do valor 0**

- No **sistema indo-arábico decimal**, um conjunto de dez algarismos (0, 1, 2, ..., 8, 9) é utilizado para representar quantas vezes uma potência de 10 encontra-se presente no número:
  - Tanto no número **48** como no número **28**, o algarismo 8 representa a mesma quantidade
- A posição (ou **casa**) de cada algarismo, contada a partir de **0**, da direita para a esquerda, indica qual **potência de 10 (**10 elevado à casa**)** deve multiplicar o algarismo
- **A existência do 0 (e de sua noção) é fundamental!**

Sistema posicional

- **<u>Def. 1</u>:** um número $x \in \mathbb{Z}$ contendo n algarismos é representado no sistema indo-arábico decimal através da sequência de algarismos

$$\pm x_{n-1} x_{n-2} \dots x_1 x_0$$

e seu valor é dado por

$$x = \pm \sum_{i=0}^{n-1} \left(x_i \times 10^i \right)$$

onde $0 \leq x_i \leq 9$.

- **Def. 2**: um número $x \in \mathbb{R}$ contendo n algarismos na sua parte inteira e m algarismos na sua parte fracionária (exibida após a vírgula ou ponto decimal) é representado no sistema indo-arábico decimal através da sequência de algarismos

$$\pm x_{n-1} x_{n-2} \ldots x_1 x_0, x_{-1} x_{-2} \ldots x_{-m+1} x_{-m}$$

e seu valor é dado por

$$x = \pm \left(\sum_{i=0}^{n-1} \left(x_i \times 10^i \right) + \sum_{i=-m}^{-1} \left(x_i \times 10^i \right) \right).$$

- Essas duas definições permitem representar qualquer número
  - No caso dos números irracionais, ainda que não haja um número finito de algarismos na parte fracionária, alguns são representados e os demais ficam indicados por reticências, como o número $\pi = 3{,}1415926\ldots$
- O sistema de numeração posicional indo-arábico permite, ainda, que se utilize o mesmo para representar números em outras **bases** que não a base decimal
- O surgimento de computadores ao final da década de 1940 é um exemplo do uso de sistemas de numeração posicionais com bases diferentes de 10
- Existiram computadores que operavam com números representados em base 8 (octal) e base 16 (hexadecimal), como os Burroughs B6700 e IBM Series 360, respectivamente
- Atualmente, os microprocessadores de fabricantes como Intel e AMD utilizam aritmética inteira ou real (de **ponto-flutuante**) com números expressos em base 2, ou base **binária**

- Dependendo da base usada num sistema de numeração posicional, há um conjunto finito de algarismos:
  - Binário: $\mathbb{B} = \{0,1\}$
  - Octal: $\mathbb{O} = \{0,1,2,3,4,5,6,7\}$
  - Decimal: $\mathbb{D} = \{0,1,2,3,4,5,6,7,8,9\}$
  - Hexadecimal: $\mathbb{H} = \{0,1,2,3,4,5,6,7,8,9, A, B, C, D, E, F\}$
- Note que:
  - A quantidade de elementos de cada conjunto (o **cardinal do conjunto**) é igual à base do sistema de numeração
  - O valor do maior algarismo no conjunto é sempre igual a $b - 1$, onde b é a base do sistema de numeração
  - Cada algarismo deve ocupar apenas uma casa, razão pelo qual sistemas de base maior do que 10 têm de usar letras e outros símbolos para representar os algarismos

- A representação de um número $x \in \mathbb{R}$ em base 2 é uma extensão natural daquela mostrada na Def. 2; nesse caso, teremos

$$x = \pm \sum_{i=-m}^{n-1} \left(x_i \times 2^i \right)$$

onde $x_i \in \mathbb{B} = \{0,1\}$ e n e m são definidos como anteriormente.

- Em geral, quando é possível que haja ambiguidade quanto à base na qual um número é representado, ela pode ser indicada usando-a como subscrito do número:
  - $(3{,}14)_{10} = 3 \times 10^0 + 1 \times 10^{-1} + 4 \times 10^{-2} = 3{,}14$
  - $(3{,}14)_8 = 3 \times 8^0 + 1 \times 8^{-1} + 4 \times 8^{-2} = (3{,}1875)_{10}$
  - $(11{,}101)_{10} = 1 \times 10^1 + 1 \times 10^0 + 1 \times 10^{-1} + 0 \times 10^{-2} + 1 \times 10^{-3} = 11{,}101$
  - $(11{,}101)_2 = 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = (3{,}625)_{10}$

Conversão de base

- **Conversão de base** é o nome que se dá ao processo de obtenção dos algarismos de um número expresso num sistema posicional de base b a partir dos algarismos desse mesmo número, expresso num sistema posicional de base a
- Apresentaremos aqui o processo de conversão entre as bases decimal e binária, para números inteiros e reais
- O sinal ($+$ ou $-$) que o número a ser convertido possa exibir é desconsiderado no processo, bastando adicioná-lo ao resultado

Conversão de número inteiro decimal para binário

- Seja um número $(x)_{10} \in \mathbb{Z}$ contendo n algarismos,

$$(x)_{10} = \pm(x_{n-1}x_{n-2} \dots x_1x_0)_{10}$$

onde $0 \leq x_i \leq 9$, para $0 \leq i < n$.

- Deseja-se encontrar o número equivalente a x em binário,

$$(y)_2 = \pm(y_{m-1}y_{m-2} \cdots y_1y_0)_2$$

onde $0 \leq y_i \leq 1$, para $0 \leq i < m$.

- Exemplos:
  - $(31)_{10} = (0001\ 1111)_2$
  - $(129)_{10} = (1000\ 0001)_2$

Conversão de número inteiro decimal para binário

- O processo de conversão consiste na **divisão sucessiva do número decimal por 2**, anotando o **resto da divisão** obtido a cada divisão efetuada:
  - Observe que o resto da divisão de um número decimal por 2 só pode ser **0** ou **1**!
- As divisões continuam a ser efetuadas até que o dividendo seja igual a **1**
- Após, escrevem-se os restos na **ordem inversa** à qual foram obtidos

Conversão de número inteiro decimal para binário

- Exemplo: converter $(401)_{10}$ em binário.

dividendo	quociente da divisão por 2	resto da divisão por 2
401	200	1
200	100	0
100	50	0
50	25	0
25	12	1
12	6	0
6	3	0
3	1	1
1	0	1

$(1\ 1001\ 0001)_2 = 1 \times 2^8 + 1 \times 2^7 + 1 \times 2^4 + 1 \times 2^0 = 256 + 128 + 16 + 1 = (401)_{10}$

Conversão de número real decimal para binário

- Um número real (expresso em qualquer base) pode ser dividido em duas partes:
  - **Inteira**
  - **Fracionária**
- A parte **inteira** do número real decimal pode ser convertida para binário usando o processo de conversão de número inteiro decimal para binário, já apresentado
- A parte **fracionária** do número real decimal é convertida para binário utilizando-se um processo de **multiplicação sucessiva por 2**, anotando-se a **parte inteira de cada produto** calculado e utilizando-se a **parte fracionária** desse produto como o próximo multiplicando
  - Observe que a parte inteira de cada produto só poderá ser **0** ou **1**!
  - As multiplicações continuam até que o produto seja igual a **1**
  - As partes inteiras de cada multiplicando são os algarismos binários desejados, escritos na ordem em que foram obtidos

Conversão de número real decimal para binário

- Exemplo: converter $(401{,}640625)_{10}$ em binário.

multiplicando	produto por 2	parte inteira do produto
0,640625	1,28125	1
0,28125	0,5625	0
0,5625	1,125	1
0,125	0,25	0
0,25	0,5	0
0,5	1	1

$$\begin{aligned}(0{,}1010\ 01)_2 &= 1 \times 2^{-1} + 1 \times 2^{-3} + 1 \times 2^{-6} \\ &= 0{,}5 + 0{,}125 + 0{,}015625 \\ &= (0{,}640625)_{10}\end{aligned}$$

- Somando ambas as partes inteira e fracionária em binário, temos

$$(401{,}640625)_{10} = (1\ 1001\ 0001{,}1010\ 01)_2$$

Conversão de número real decimal para binário

- Note que um número em base 10, menor do que 1 e com quantidade finita de algarismos (**número racional decimal**), pode **não ter** representação binária com número finito de algarismos!

Conversão de número real decimal para binário

- Um exemplo é o número

$$(0{,}1)_{10} = (0{,}0\ \boxed{0011}\ 0011\ 0011\ \ldots)_2$$

multiplicando	produto por 2	parte inteira do produto
0,1	0,2	0
0,2	0,4	0
0,4	0,8	0
0,8	1,6	1
0,6	1,2	1
0,2	0,4	0
0,4	0,8	0
	…	

Conversão de número inteiro binário para decimal

- Para efetuar essa conversão, basta determinar a quantidade, n, de algarismos binários presentes no número

$$(y)_2 = \pm(y_{m-1}y_{m-2} \cdots y_1y_0)_2$$

e calcular o somatório

$$(x)_{10} = \pm\sum_{i=0}^{n-1}(y_i \times 2^i)$$

Conversão de número real binário para decimal

- Para efetuar essa conversão, basta determinar as quantidades, n e m, de algarismos binários presentes nas partes inteira e fracionária do número

$$(y)_2 = \pm y_{n-1} y_{n-2} \cdots y_1 y_0, y_{-1} y_{-2} \cdots y_{-m+1} y_{-m}$$

e calcular o somatório

$$(x)_{10} = \pm \sum_{i=-m}^{n-1} \left(y_i \times 2^i \right)$$