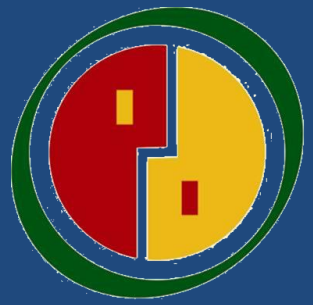


UNIVERSIDADE ESTADUAL DO RIO GRANDE DO SUL
PROF. DR. ÉDER JULIO KINAST <eder-kinast@uergs.edu.br>
MÉTODOS NUMÉRICOS – APONTAMENTOS DE AULA

01. Precisão Numérica

Versão 04 – 02/08/2019

Motivação

Conversão do Sistema Decimal para Binário

Ponto Flutuante

Motivação – erros numéricos observados em computadores

Exemplo – Somar 30.000 parcelas de valores 0,5 e 0,2 utilizando uma calculadora comum e um computador.

$$\sum_{i=1}^{30.000} 0{,}5 = 15.000$$

na calculadora e no computador.

$$\sum_{i=1}^{30.000} 0{,}2 = 6.000$$

na calculadora e

$$\sum_{i=1}^{30.000} 0{,}2 = 6.001{,}152832 \ldots$$

em um computador com precisão *float* !!!

Conversão do Sistema Decimal para Binário de Fracionários

Exemplos **Binário → Decimal**

$$(1001)_2 = (9)_{10}$$

$$1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$
$$8 + 0 + 0 + 1 = 9$$

$$(1{,}01)_2 = (1{,}25)_{10}$$

$$1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$$
$$1 + 0 + \frac{1}{4} = 1{,}25$$

Conversão do Sistema Decimal para Binário de Fracionários

Exemplos **Decimal → Binário**

$$(2{,}5)_{10} = (10{,}1)_2$$

$$1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1}$$
$$2 \quad + \quad 0 \quad + \quad \frac{1}{2} \quad = 2{,}5$$

$$(0{,}2)_{10} = (0{,}001100110011\ldots)_2$$

São necessárias infinitas casas binárias para representar exatamente o número 0,2 !!!

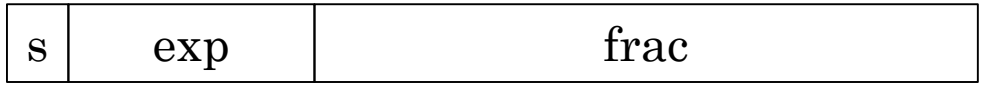
Mas um computador (binário) possui um número finito de casas para representar os números !!!

Verificar as Somas

C-soma.CPP

```
#include<iostream>

int main()
{
    float parcela=0.2,soma=0; //tentar double
    int i;

    for(i=1;i<=30000;i=i+1) soma=soma+parcela;

    printf("A soma vale %f\n\n",soma); //tentar double lf

    system("PAUSE");
    return 0;
}
```

Precisões de Ponto Flutuante em Computadores (IEEE 754)



- **float** (32 bits): exp = 8 bits, frac = 23 bits, s = 1 bit
  - Faixa de valores: 2^{-126} até 2^{127}
  - Precisão: 10^{-7}
- **double** (64 bits): exp = 11 bits, frac = 52 bits, s = 1 bit
  - Faixa de valores: 2^{-1022} até 2^{1023}
  - Precisão: 10^{-15}
- **long double** (79 bits): exp =15 bits, frac = 63 bits, s = 1 bit
  - Faixa de valores: 2^{-16382} até 2^{16383}
  - Precisão: 10^{-317}

Verificar as Precisões

C-precisao.CPP

```
#include<iostream>

int main()
{
    float A=1,S=2; //tentar double

    while(S>1) {
        A=A/10;
        S=1+A;
    }

    printf("A precisao vale %e\n\n",10*A);

    system("PAUSE");
    return 0;
}
```

Verificar as Precisões no Excel

MetNum01.xlsx

A	S	Resultado
1	2	início
0,1	1,1	S é maior que 1
0,01	1,01	S é maior que 1
0,001	1,001	S é maior que 1
0,0001	1,0001	S é maior que 1
0,00001	1,00001	S é maior que 1