Representação no Computador

INF1608 – Análise Numérica

Waldemar Celes
celes@inf.puc-rio.br

Departamento de Informática, PUC-Rio

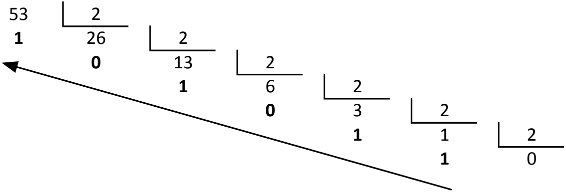
Representação binária

Conversão de decimal para binário

- $(53)_{10}$ em representação binária?

Conversão de decimal para binário

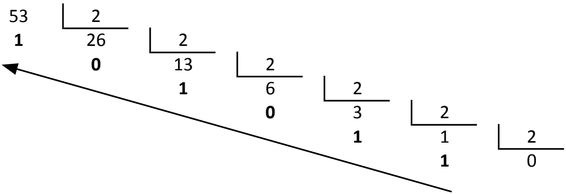
- $(53)_{10}$ em representação binária?



Representação binária

Conversão de decimal para binário

- $(53)_{10}$ em representação binária?



Logo:

$$(53)_{10} = (110101)_2$$

Representação binária

$(0.7)_{10}$?

Representação binária

$(0.7)_{10}$?

$$
\begin{aligned}
0.7 \times 2 &= 0.4 + 1 \\
0.4 \times 2 &= 0.8 + 0 \\
0.8 \times 2 &= 0.6 + 1 \\
0.6 \times 2 &= 0.2 + 1 \\
0.2 \times 2 &= 0.0 + 0 \\
0.4 \times 2 &= 0.8 + 0 \\
&\ldots
\end{aligned}
$$

Representação binária

$(0.7)_{10}$?

$$
\begin{aligned}
0.7 \times 2 &= 0.4 + 1 \\
0.4 \times 2 &= 0.8 + 0 \\
0.8 \times 2 &= 0.6 + 1 \\
0.6 \times 2 &= 0.2 + 1 \\
0.2 \times 2 &= 0.0 + 0 \\
0.4 \times 2 &= 0.8 + 0 \\
&\ldots
\end{aligned}
$$

Logo:

$$(0.7)_{10} = (0.1011001100110\ldots)_2 = (0.1\overline{0110})_2$$

Representação binária

$(0.7)_{10}$?

$$
\begin{aligned}
0.7 \times 2 &= 0.4 + 1 \\
0.4 \times 2 &= 0.8 + 0 \\
0.8 \times 2 &= 0.6 + 1 \\
0.6 \times 2 &= 0.2 + 1 \\
0.2 \times 2 &= 0.0 + 0 \\
0.4 \times 2 &= 0.8 + 0 \\
&\ldots
\end{aligned}
$$

Logo:

$$(0.7)_{10} = (0.1011001100110\ldots)_2 = (0.1\overline{0110})_2$$

E:

$$(53.7)_{10} = (110101.1011001100110\ldots)_2 = (110101.1\overline{0110})_2$$

Representação binária

Exemplo exato: $(0.25)_{10}$?

Representação binária

Exemplo exato: $(0.25)_{10}$?

col:

$$
\begin{aligned}
0.25 \times 2 &= 0.5 + 0 \\
0.5 \times 2 &= 0.0 + 1
\end{aligned}
$$

Logo:

$$(0.25)_{10} = (0.01)_2$$

Representação binária

Conversão binário para decimal

- Parte inteira:

$$(10101)_2 \ ?$$

Representação binária

Conversão binário para decimal

- Parte inteira:

$$(10101)_2 \ ?$$

$$1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (21)_{10}$$

Parte fracionária:

$$(0.1011)_2 \ ?$$

Representação binária

Conversão binário para decimal

- Parte inteira:

$$(10101)_2 \ ?$$

$$1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (21)_{10}$$

Parte fracionária:

$$(0.1011)_2 \ ?$$

$$\begin{aligned} & \frac{1}{2^1} + \frac{0}{2^2} + \frac{1}{2^3} + \frac{1}{2^4} = \\ = & \frac{1}{2} + \frac{0}{4} + \frac{1}{8} + \frac{1}{16} = \left(\frac{11}{16}\right)_2 \end{aligned}$$

Representação binária

Caso de "dízimas" binárias:

$$x = (0.\overline{1011})_2$$

Representação binária

Caso de "dízimas" binárias:

$$x = (0.\overline{1011})_2$$

$$\begin{aligned} 2^4 x &= 1011.\overline{1011} \\ x &= 0000.\overline{1011} \end{aligned}$$

Representação binária

Caso de "dízimas" binárias:

$$x = (0.\overline{1011})_2$$

$$\begin{aligned} 2^4 x &= 1011.\overline{1011} \\ x &= 0000.\overline{1011} \end{aligned}$$

$$(2^4 - 1)x = (1011)_2 = (11)_{10}$$

Logo:

$$x = \frac{11}{2^4 - 1} = \left(\frac{11}{15}\right)_{10}$$

Representação binária

Caso de “dízimas” binárias:

$$x = (0.10\overline{101})_2$$

Representação binária

Caso de "dízimas" binárias:

$$x = (0.10\overline{101})_2$$

$$y = 2^2 x = 10.\overline{101}$$
$$z = 0.\overline{101}$$

Representação binária

Caso de “dízimas” binárias:

$$x = (0.10\overline{101})_2$$

$$y = 2^2 x = 10.\overline{101}$$
$$z = 0.\overline{101}$$

Procedimento:

- Acha z
- Faz: $y = 2 + z$
- E então:

$$x = \frac{y}{2^2}$$

Representação no computador

Notação científica de números binários:

$$\pm 1.bbbb... \times 2^{eee...}$$

- A parte inteira é sempre 1
- A base é 2
- $bbbb...$ representa a mantissa
- $eee...$ representa o expoente

Representação no computador

Tipos ponto flutuante no computador

- Número de bits

	sinal	mantissa	expoente
float	1	23	8
double	1	52	11

Representação no computador

Precisão simples (float):

- Quantos dígitos decimais de precisão?

Representação no computador

Precisão simples (float):

- Quantos dígitos decimais de precisão?

$$2^{23} = 10^x$$

$$\log_2 2^{23} = \log_2 10^x$$
$$23 = x \log_2 10$$

  - Como $\log_2 10 \approx 3.322$, temos:
$$x = 6.92$$

Representação no computador

Precisão simples (float):

- Quantos dígitos decimais de precisão?

$$2^{23} = 10^x$$

$$\log_2 2^{23} = \log_2 10^x$$
$$23 = x \log_2 10$$

- Como $\log_2 10 \approx 3.322$, temos:
$$x = 6.92$$

Precisão dupla (double):

$$x = \frac{52}{\log_2 10}$$
$$x = 15.65$$

Representação no computador

Precisão dupla (double)

- Representação na máquina

$$\underbrace{s e_1 e_2 \ldots e_{11} b_1 b_2 \ldots b_{52}}_{\text{64 bits}}$$

Representação no computador

Precisão dupla (double)

- Representação na máquina

$$\underbrace{se_1 e_2 \ldots e_{11} b_1 b_2 \ldots b_{52}}_{\text{64 bits}}$$

- Representação do número 1:

$$+1.000\ldots000 \times 2^0$$

Representação no computador

Precisão dupla (double)

- Representação na máquina

$$\underbrace{s e_1 e_2 \ldots e_{11} b_1 b_2 \ldots b_{52}}_{\text{64 bits}}$$

- Representação do número 1:

$$+1.000\ldots000 \times 2^0$$

  - Menor número maior que 1:

Representação no computador

Precisão dupla (double)

- Representação na máquina

$$\underbrace{se_1 e_2 \ldots e_{11} b_1 b_2 \ldots b_{52}}_{\text{64 bits}}$$

- Representação do número 1:

$$+1.000\ldots000 \times 2^0$$

  - Menor número maior que 1:

$$+1.000\ldots001 \times 2^0 = 1 + 2^{-52}$$

Representação no computador

Precisão dupla (double)

- Representação na máquina

$$\underbrace{s e_1 e_2 \ldots e_{11} b_1 b_2 \ldots b_{52}}_{\text{64 bits}}$$

- Representação do número 1:

$$+1.000\ldots000 \times 2^0$$

  - Menor número maior que 1:

$$+1.000\ldots001 \times 2^0 = 1 + 2^{-52}$$

Epsilon da máquina

$$\epsilon_{mach} = 2^{-52}$$

Representação no computador

Arredondamento

- Se bit 53 for 0: descarta-se bits 53 em diante
- Se bit 53 for 1 e existir bit > 53 com valor 1: soma-se 2^{-52}, descarta-se bits 53 em diante
- Se bit 53 for 1 e bits > 53 for 0
  - Se bit 52 for 0: destar-se bits 53 em diante
  - Se bit 52 for 1: soma-se 2^{-52}, descarta-se demais
  - Note que ao final, bit 52 fica com valor 0

Representação no computador

Representação do expoente

- 11 bits (valores positivos): 0 a 2047

Representação no computador

Representação do expoente

- 11 bits (valores positivos): 0 a 2047
- Valores especiais: 0 e 2047

Representação no computador

Representação do expoente

- 11 bits (valores positivos): 0 a 2047
- Valores especiais: 0 e 2047
- Valor do expoente
  - Avaliação dos bits com valores: $x \in [1, 2046]$

$$e_{xp} = x - 1023$$

Representação no computador

Representação do expoente

- 11 bits (valores positivos): 0 a 2047
- Valores especiais: 0 e 2047
- Valor do expoente
  - Avaliação dos bits com valores: $x \in [1, 2046]$

$$e_{xp} = x - 1023$$

- Exemplos
  - Se $x = 1$: $e_{xp} = -1022$
  - Se $x = 2046$: $e_{xp} = 1023$
  - Se $x = 1023$: $e_{xp} = 0$

Representação no computador

Representação do expoente

- 11 bits (valores positivos): 0 a 2047
- Valores especiais: 0 e 2047
- Valor do expoente
  - Avaliação dos bits com valores: $x \in [1, 2046]$

$$e_{xp} = x - 1023$$

- Exemplos
  - Se $x = 1$: $e_{xp} = -1022$
  - Se $x = 2046$: $e_{xp} = 1023$
  - Se $x = 1023$: $e_{xp} = 0$

Em resumo:

- Para armazenar: soma-se 1023
- Para interpretar: subtrai-se 1023

Representação no computador

Números especiais

- Expoente: $(11111111111)_2 = (2047)_{10}$
  - Se mantissa diferente de zero: NaN
  - Se mantissa for zero: $\pm Inf$

$$+Inf : 011111111111000...000$$

$$-Inf : 1\underbrace{11111111111}_{11 bits}\underbrace{000...000}_{52 bits}$$

Representação no computador

Números especiais

- Expoente: $(11111111111)_2 = (2047)_{10}$
  - Se mantissa diferente de zero: NaN
  - Se mantissa for zero: $\pm Inf$

$$+Inf : 011111111111000...000$$

$$-Inf : 1\underbrace{11111111111}_{11 bits}\underbrace{000...000}_{52 bits}$$

- Expoente: $(00000000000)_2 = (0)_{10}$
  - Números sub-normais (não normalizados)

$$\pm 0.b_1 b_2 ... b_{52} \times 2^{-1022}$$

Representação no computador

Números especiais

- Expoente: $(11111111111)_2 = (2047)_{10}$
  - Se mantissa diferente de zero: *NaN*
  - Se mantissa for zero: $\pm Inf$

$$+Inf : 011111111111000...000$$

$$-Inf : 1\,\underbrace{11111111111}_{11 bits}\,\underbrace{000...000}_{52 bits}$$

- Expoente: $(00000000000)_2 = (0)_{10}$
  - Números sub-normais (não normalizados)

$$\pm 0.b_1 b_2 \ldots b_{52} \times 2^{-1022}$$

Logo:

- Menor número diferente de 0 representável

$$2^{-52} \times 2^{-1022} = 2^{-1074}$$

$$\underbrace{0}\;\underbrace{00000000000}\;\underbrace{000000...0000001}$$

Representação no computador

Observações:

- Números menores que 2^{-1074} não podem ser representados
  - Existem números representáveis ($< \epsilon_{mach}$) que, se somados a 1, não alteram seu valor

Representação no computador

Observações:

- Números menores que 2^{-1074} não podem ser representados
  - Existem números representáveis ($< \epsilon_{mach}$) que, se somados a 1, não alteram seu valor
- Números sub-normais incluem o zero: ± 0
  - -0 e $+0$ são tratados como iguais

Representação no computador

Exercício:

- Se $fl(x)$ indica a representação ponto flutuante do valor x no computador, podemos afirmar que $fl(0.2) < 0.2$?

Representação no computador

Exercício:

- Se $fl(x)$ indica a representação ponto flutuante do valor x no computador, podemos afirmar que $fl(0.2) < 0.2$?

Representação binária de 0.2:

$$
\begin{aligned}
0.2 \times 2 &= 0.0 + 0 \\
0.4 \times 2 &= 0.8 + 0 \\
0.8 \times 2 &= 0.6 + 1 \\
0.6 \times 2 &= 0.2 + 1 \\
0.2 \times 2 &= 0.0 + 0 \\
\ldots
\end{aligned}
$$

Logo: $(0.2)_{10} = (0.\overline{0011})_2$

Representação no computador

Exercício (cont):

$$(0.2)_{10} = (0.\overline{0011})_2$$

Notação científica:

$$1.1001\overline{1001} \times 2^{-3}$$

- Então, o bit 53 vale 1, com valores diferentes de 0 em seguida
  - Soma-se $2^{-52}2^{-3}$, descarta-se $(0.\overline{1001})_2 \times 2^{-52}2^{-3}$
  - Isto é:

$$\begin{aligned} fl(0.2) &= 0.2 + 2^{-55} - \frac{9}{15} \times 2^{-55} \\ &= 0.2 + (1 - 0.6) \times 2^{-55} = 0.2 + 0.4 \times 2^{-55} > 0.2 \end{aligned}$$

Representação no computador

Exercícios propostos:

- *fl*$(1.2) > 1.2$?

Representação no computador

Exercícios propostos:

- $fl(1.2) > 1.2$?
- $fl(0.3) > 0.3$?

Representação no computador

Exercício: Considere o trecho de código C abaixo

```c
#include <stdio.h>

int main (void)
{
  double a = 1.2 - 1.0 - 0.2;
  if (a == 0.0)
    printf("OK\n");
  else
    printf("ERRO\n");
  return 0;
}
```

- Qual mensagem impressa?

Representação no computador

Avaliação da expressão: $1.2 - 1.0 - 0.2$

$$fl(1.2) = ?$$

$$fl(1.0) = 1.0$$

$$fl(0.2) = 0.2 + 0.4 \times 2^{-55}$$

Representação no computador

Avaliação da expressão: $1.2 - 1.0 - 0.2$

$$\begin{aligned} fl(1.2) &= ? \\ fl(1.0) &= 1.0 \\ fl(0.2) &= 0.2 + 0.4 \times 2^{-55} \end{aligned}$$

Temos:

$$(1.2)_{10} = (1.\overline{0011})_2$$

Representação no computador

Avaliação da expressão: $1.2 - 1.0 - 0.2$

$$\begin{aligned} fl(1.2) &= ? \\ fl(1.0) &= 1.0 \\ fl(0.2) &= 0.2 + 0.4 \times 2^{-55} \end{aligned}$$

Temos:

$$(1.2)_{10} = (1.\overline{0011})_2$$

Logo:

$$fl(1.2) = 1.2 - 0.2 \times 2^{-52}$$

Representação no computador

Avaliação da expressão: $1.2 - 1.0 - 0.2$

$$fl(1.2) = ?$$
$$fl(1.0) = 1.0$$
$$fl(0.2) = 0.2 + 0.4 \times 2^{-55}$$

Temos:

$$(1.2)_{10} = (1.\overline{0011})_2$$

Logo:

$$fl(1.2) = 1.2 - 0.2 \times 2^{-52}$$

Então:

$$\begin{aligned} fl(1.2 - 1.0 - 0.2) &= 1.2 - 0.2 \times 2^{-52} - 1.0 - (0.2 + 0.4 \times 2^{-55}) \\ &= (-1.6 - 0.4) \times 2^{-55} = -2 \times 2^{-55} = -2^{-54} \\ &= -5.5511151231 \times 10^{-17} \end{aligned}$$

Representação no computador

Exercício proposto

- Qual o valor da expressão: $fl(2.3 - 2 - 0.3)$

Avaliação de erro

Como avaliar o erro de um método/procedimento?

Avaliação de erro

Como avaliar o erro de um método/procedimento?

Erro absoluto

$$|x_c - x| < \epsilon$$

onde x_c é o valor computado e x é o valor exato

Avaliação de erro

Como avaliar o erro de um método/procedimento?

Erro absoluto

$$|x_c - x| < \epsilon$$

onde x_c é o valor computado e x é o valor exato

- Pode falhar se os númeos forem pequenos

Avaliação de erro

Erro relativo

$$\frac{|x_c - x|}{|x|} < \epsilon$$

Avaliação de erro

Erro relativo

$$\frac{|x_c - x|}{|x|} < \epsilon$$

- O padrão IEEE garante: $\frac{|fl(x)-x|}{|x|} \leq \frac{1}{2}\epsilon_{mach}$

Avaliação de erro

Erro relativo

$$\frac{|x_c - x|}{|x|} < \epsilon$$

- O padrão IEEE garante: $\frac{|fl(x) - x|}{|x|} \leq \frac{1}{2}\epsilon_{mach}$
- Pode falhar quando:
  - x e x_c são zero: $=> NaN$
  - x é zero: $=> Inf$
  - x_c e x são pequenos, mas opostos em zero: $=>$ sempre falso

Avaliação de erro

Erro relativo

$$\frac{|x_c - x|}{|x|} < \epsilon$$

- O padrão IEEE garante: $\frac{|fl(x)-x|}{|x|} \leq \frac{1}{2}\epsilon_{mach}$
- Pode falhar quando:
  - x e x_c são zero: $=> NaN$
  - x é zero: $=> Inf$
  - x_c e x são pequenos, mas opostos em zero: $=>$ sempre falso

Fórmula alternativa

$$\frac{|x_c - x|}{|x_c| + |x|} < \epsilon$$