CÁLCULO NUMÉRICO

Aula: Erros de arredondamento – Parte 1.

ENG. MECÂNICA – IFPE (RECIFE)
Prof. Frederico Duarte de Menezes
Contato: fredericomenezes@recife.ifpe.edu.br

ERROS DE ARREDONDAMENTO

Representação numérica:

- Um elemento importante para a obtenção de um resultado confiável é a representação dos valores calculados.

Valores mal representados significam resultados ruins.

ERROS DE ARREDONDAMENTO

Como garantir uma boa representação?

- Cálculos manuais: quanto mais papel e caneta melhor!!!
- Utilizando calculadoras ou computadores: uma quantidade mínima de memória para armazenar os valores.

Em ambos os casos, não podemos ter em mãos folhas de papel ou memória de computador infinitas...

INSTITUTO FEDERAL
Pernambuco
Campus Recife

ERROS DE ARREDONDAMENTO

Representação Limitada

Erros de Arredondamento...

INSTITUTO FEDERAL
Pernambuco
Campus Recife

ERROS DE ARREDONDAMENTO

Como podemos entender o erro de arredondamento de forma mais simples?

NÚMEROS NO COMPUTADOR

Programas de computador podem armazenar, de forma geral, números em duas representações:

- Inteiros:
  - ...,-1000,...,-2,-1,0,1,2,...,1000,...
- Ponto flutuante:
  - ...,-2.34,...-0.55576,...,3.1415123456,...

INSTITUTO FEDERAL
Pernambuco
Campus Recife

NÚMEROS NO COMPUTADOR

De uma forma geral, entendemos que um número real pode ser representado como uma série de somas de valores tais como:

$$\frac{8}{3} = 2.6666\ldots = \left(\frac{2}{10^1} + \frac{6}{10^2} + \frac{6}{10^3} + \frac{6}{10^4} + \frac{6}{10^5} + \cdots\right) \times 10^1$$

NÚMEROS NO COMPUTADOR

Na prática, o valor é armazenado em uma quantidade finita de memória, o que leva à uma representação aproximada do valor verdadeiro:

$$\frac{8}{3} \simeq \left(\frac{2}{10^1} + \frac{6}{10^2} + \frac{6}{10^3} + \frac{6}{10^4} \right) \times 10^1 = 0.2666 \times 10^1$$

ERROS DE ARREDONDAMENTO

Como resultado, dizemos que o valor real tem sua representação “arredondada” para um valor em ponto flutuante.

Podemos dizer que um número em ponto flutuante é representado por *t* dígitos decimais, sendo ***t*** denominado de precisão do valor.

INSTITUTO FEDERAL
Pernambuco
Campus Recife

PONTO FLUTUANTE

Generalizando...

$$
\begin{aligned}
\mathrm{fl}(x) &= \pm 0.d_1 d_2 \ldots d_{t-1} d_t \times 10^e \\
&= \pm \left(\frac{d_1}{10^1} + \frac{d_2}{10^2} + \cdots + \frac{d_{t-1}}{10^{t-1}} + \frac{d_t}{10^t} \right) \times 10^e
\end{aligned}
$$

Onde ***x*** representa o valor real, ***d_i*** representa os valores em cada posição decimal e ***e*** representa o nosso expoente.

Lembrando que *t* representa o número de casas decimais em nossa representação.

PONTO FLUTUANTE

No nosso exemplo:

$$\frac{8}{3} \simeq \left(\frac{2}{10^1} + \frac{6}{10^2} + \frac{6}{10^3} + \frac{6}{10^4}\right) \times 10^1 = 0.2666 \times 10^1$$

Temos ***t=4* e *e=1***.

O valor de e não é obrigatoriamente sempre igual a 1:

$$0.2666 \times 10^1 = 0.02666 \times 10^2$$

INSTITUTO FEDERAL
Pernambuco
Campus Recife

PONTO FLUTUANTE

Olhando a nossa representação geral, fica claro que os valores de ***d_i*** devem obedecer:

$$0 \leq d_i \leq (b-1)$$

Onde b representa a base de nossa notação. No nosso exemplo **b=10**, logo:

$$0 \leq d_i \leq (10-1) \;\rightarrow\; 0 \leq d_i \leq (9)$$

PONTO FLUTUANTE

Olhando a nossa representação geral, fica claro que os valores de ***d_i*** devem obedecer:

$$0.2666 \times 10^{1}$$

Onde b represent tação. No nosso exemplo **b=10**, log

$$0 \leq d_i \leq (10-1) \rightarrow 0 \leq d_i \leq (9)$$

PONTO FLUTUANTE

Da mesma forma, o valor de ***e*** também é limitado a um número finito de valores inteiros:

$$\boldsymbol{L \leq e \leq U}$$

Onde L representa o valor limite inferior e U o limite superior de valores possíveis para o expoente.

INSTITUTO FEDERAL
Pernambuco
Campus Recife

PONTO FLUTUANTE

Como consequência principal, nossa representação em um computador genérico, sendo:

$$0.99\ldots99 \times 10^{U} \lessapprox 10^{U}$$

Maior valor real possível

$$0.10\ldots00 \times 10^{L} = 10^{L-1}$$

Menor valor real possível

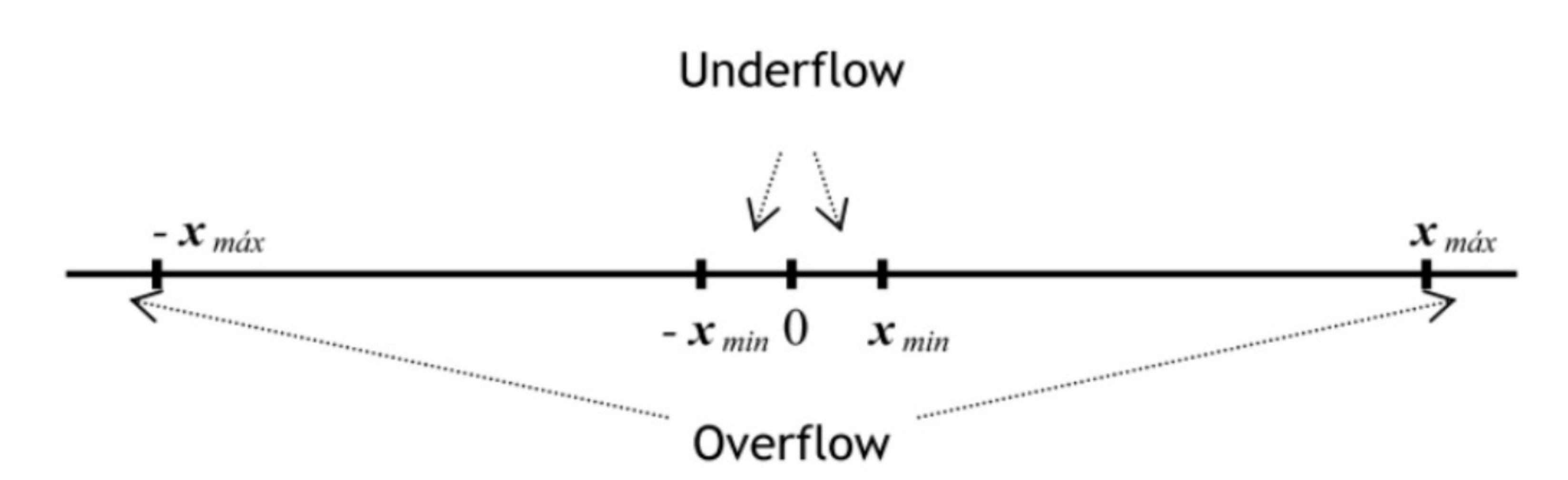
Lembrando que esses valores podem ser + ou -

ERROS DE ARREDONDAMENTO

Do ponto de vista prático, valores absolutos maiores do que a representação determina são aproximados para o maior permitido, gerando um erro de arredondamento denominado de ***OVERFLOW***.

Assim como um valor absoluto muito próximo de zero, menor do que a representação determina, será aproximado para ***ZERO***, gerando um erro de arredondamento denominado de ***UNDERFLOW***.

ERROS DE ARREDONDAMENTO

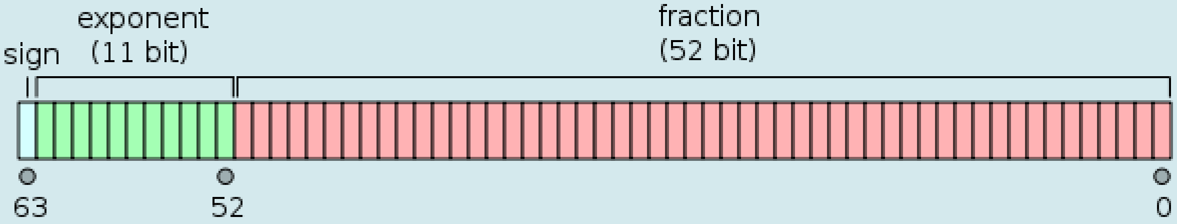


INSTITUTO FEDERAL
Pernambuco
Campus Recife

ERROS DE ARREDONDAMENTO

Como evitar o underflow e overflow??

Uma representação bastante utilizada em computação é o formato IEEE de dupla precisão, onde um valor em ponto flutuante faz uso de 64 bits (8 bytes) para sua representação.



INSTITUTO FEDERAL
Pernambuco
Campus Recife

ÉPSILON DE MÁQUINA

Entendendo o conceito de ***underflow***, fica a pergunta:

Qual o menor valor diferente de zero representado por um computador?

Em outras palavras, qual o menor valor que somado a 1 resulta em um valor diferente de 1?

INSTITUTO FEDERAL
Pernambuco
Campus Recife

ÉPSILON DE MÁQUINA

Este valor é denominado de Épsilon de Máquina, representando a unidade de arredondamento de um computador.

Para cada representação de valores em ponto flutuante, teremos um valor de Épsilon diferente.

INSTITUTO FEDERAL
Pernambuco
Campus Recife

ÉPSILON DE MÁQUINA

IEEE 754 - 2008	Nome usual	Base b	Precisão p	Épsilon de máquina[a] $b^{-(p-1)}/2$
binary16	meia precisão	2	11 (um bit implícito)	$2^{-11} = 4.88e-04$
binary32	precisão singular	2	24 (um bit implícito)	$2^{-24} = 5.96e-08$
binary64	precisão dupla	2	53 (um bit implícito)	$2^{-53} = 1.11e-16$
binary80	precisão estendida	2	64	$2^{-64} = 5.42e-20$
binary128	precisão quádrupla	2	113 (um bit implícito)	$2^{-113} = 9.63e-35$
decimal32	precisão singular decimal	10	7	5×10^{-7}
decimal64	precisão dupla decimal	10	16	5×10^{-16}
decimal128	precisão quádrupla decimal	10	34	5×10^{-34}