2.4 Representação de números reais em ponto-fixo

. No sistema de ponto-fixo o ponto binário ocupa uma posição fixa (daí o nome) - existe uma quantidade pré-definida de dígitos à esquerda e a direita do ponto. O registrador do computador é dividido em três campos: • s, sinal do número (|s| = 1 bit); • e, dígitos à esquerda do ponto binário (|e| = 8 bits, por exemplo); • d, dígitos à direita do ponto binário (|d| = 7 bits, por exemplo). Por exemplo, o número −11,75 é representado em ponto-fixo como 1 00001011 1100000 Alguns números precisam ser arredondados, como por exemplo o número 1,2: 1,2 = (1,0011001100110011...) = (1, 0011) Arredondamento para baixo: Simplesmente descartam-se os dígitos em excesso. (1,0011001100110011...)2 ≈ (1,0011001)2 Arredondamento para mais próximo: Se o próximo dígito for 0 soma-se 0 no último dígito; se o próximo dígito for 1 soma-se 1 no último dígito. (1,0011001|1)2 ≈ (1,0011001)2 + (0,0000001)2 = (1,0011010)2

A mantissa é um número binário entre *um* e *dois* e é representada como

$$M = (1, b_1\, b_2\, \cdots\, b_n)_2$$

O padrão descreve vários tipos de números, onde os mais usados são:

Tipo	n	m	BIAS	Total de bits
Meia precisão	10	4	15	1+10+5=16
Precisão simples	23	7	127	1+23+8=32
Precisão dupla	52	10	1023	1+52+11=64
Precisão quádrupla	112	14	16383	1+112+15=128

Um número em **precisão simples** pode ser armazenado no computador gravando os seguintes bits:

s	$c_7c_6c_5c_4c_3c_2c_1c_0$	$b_1b_2b_3\cdots b_{21}b_{22}b_{23}$

Por exemplo, $(-11{,}75)_{10} = (-1011{,}11)_2 = -(1{,}01111)_2 \times 2^3$.
Utilizando o sistema de ponto-flutuante precisão simples temos que

$$-(1{,}01111)_2 \times 2^3 = (-1)^1 \times (1{,}01111)_2 \times 2^{130-127},$$

portanto $s = 1$, a mantissa é 1,01111 e a característica é $130 = (10000010)_2$.
No computador este número é armazenado como

1	10000010	01111000000000000000000

O padrão IEEE754 define regras para representação de números em ponto-flutuante. A representação foi criada com base na notação científica. Por exemplo: 1234,5 = 1,2345 × 103 (1011,01)2 = (1,01101)2 × 2 3 Portanto, um número real x é representado como x = (−1)s × M × 2 C−BIAS , 1 ≤ M < 2 onde s representa o sinal, M é a mantissa, C é a característica e BIAS é o deslocamento. Os números C e BIAS são números inteiros enquanto que M é um número fracionário. A característica é um número binário inteiro e é representada como C = (cm · · · c2c1c0)2 número π com **muitas** casas, o que gastaria muita memória. Ou poderíamos calcular o número todas as vezes, o que levaria muito tempo.

4.4 Complexidade de algoritmos

A velocidade dos algoritmos usualmente é medida em número de operações matemáticas em ponto-flutuante realizadas por segundo (flops). Por exemplo:

$$5 + 4 \times 3$$

usa duas operações matemáticas. Quantas operações são usadas para calcular a multiplicação abaixo?

$$\begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} \times \begin{pmatrix} 4 & 4 & 4 \\ 5 & 5 & 5 \\ 6 & 6 & 6 \end{pmatrix}$$

Três multiplicações por elemento mais duas somas: $(3+2)9 = 45$. Se fosse uma matriz $N \times N$? O total seria:

$$(N + (N-1))(N \cdot N) = (2N-1)N^2 = 2N^3 - N^2.$$

Exemplo 4.1 *Multiplicação de matrizes*

- *Para multiplicar uma matriz* 10×10*:* $2 \cdot (10^3) - 10^2 = 2000 - 100 = 1900$ *operações*
- *Para multiplicar uma matriz* 100×100*:* $2 \cdot (100^3) - 100^2 = 2000000 - 10000 = 1990000$ *operações*
- *Para multiplicar uma matriz* 1000×1000*:* $2 \cdot (1000^3) - 1000^2 = 2000000000 - 1000000 = 1999000000$ *operações*

Utilizando informação extra (linhas iguais, colunas iguais) podemos realizar apenas 3 multiplicações e 2 somas. Em uma matriz $N \times N$, o número de operações seria $N + (N-1) = 2N - 1$.

Considere agora dois algoritmos que realizam a mesma operação em um vetor de tamanho N. Qual algoritmo é melhor?

Fórmula Geral

Menor real positivo representado em $F(\beta, p, m, M)$

$$\begin{aligned} x_m &= 0,\underbrace{10\ldots0}_{p \text{ dígitos}} \times \beta^m \\ &= \beta^{m-1} \end{aligned}$$

~~Menor~~ Maior real positivo representado em $F(\beta, p, m, M)$

$$\begin{aligned} x_M &= 0,\underbrace{(\beta-1)(\beta-1)\ldots(\beta-1)}_{p \text{ dígitos}} \times \beta^M \\ &= (1-\beta^{-p})\beta^M \end{aligned}$$