TMEC001 - CÁLCULO NUMÉRICO

CAPÍTULO 01 – MODELAGEM COMPUTACIONAL E MATEMÁTICA E ANÁLISE DE ERROS

Prof. Felipe R. Loyola

Disciplina: Cálculo Numérico

1º Semestre de 2020

1.3 – Aproximações e Erros de Arredondamento

1.3.1 – Algarismos significativos

- O conceito de algarismos significativos foi desenvolvido para designar formalmente a confiabilidade de um valor numérico. Os algarismos significativos de um número são aqueles que podem ser usados com confiança. Eles correspondem ao número de algarismos corretos mais um estimado.
- O conceito de algarismos significativos possui duas implicações importantes nos estudos de métodos numéricos.

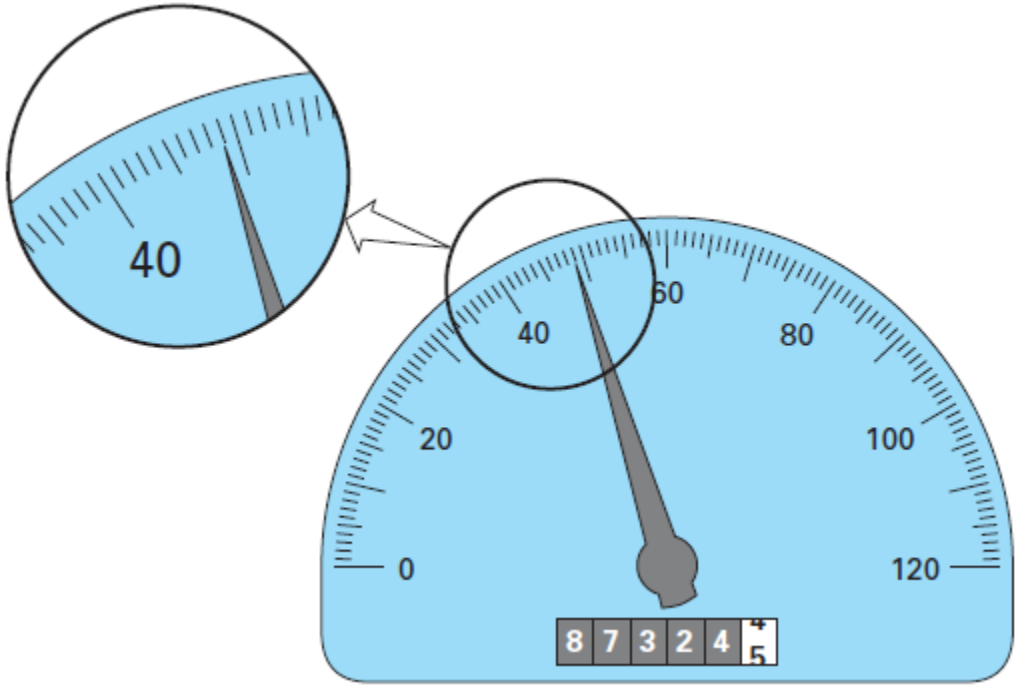


Figura 1.21

1.3.1 – Algarismos Significativos

1. Os métodos numéricos fornecem resultados aproximados. É necessário, portanto, desenvolver critérios para especificar quanta confiança se tem no resultado aproximado. Uma forma de fazer isso é em termos de algarismos significativos. Por exemplo, pode-se decidir que a aproximação é aceitável se ela for correta até quatro algarismos significativos.

1. Embora quantidades como π e $\sqrt{7}$ representem quantidades específicas, elas não podem ser expressas exatamente por um número limitado de algarismos. Por exemplo: $\pi = 3{,}14159265358979238462643 \ldots$ *ad infinitum*. Como os computadores mantem apenas um número finito de algarismos significativos. Tais números jamais podem ser representados exatamente. A omissão dos algarismos significativos remanescentes é chamada de erro de arredondamento.

1.3.2 – Acurácia e Precisão

- Os erros associados tantos aos cálculos quanto às medidas podem ser caracterizados com relação à sua acurácia e precisão. A acurácia se refere a quão próximo o valor calculado ou medido está do valor verdadeiro. A precisão se refere a quão próximos os valores individuais ou medidos estão uns dos outros.

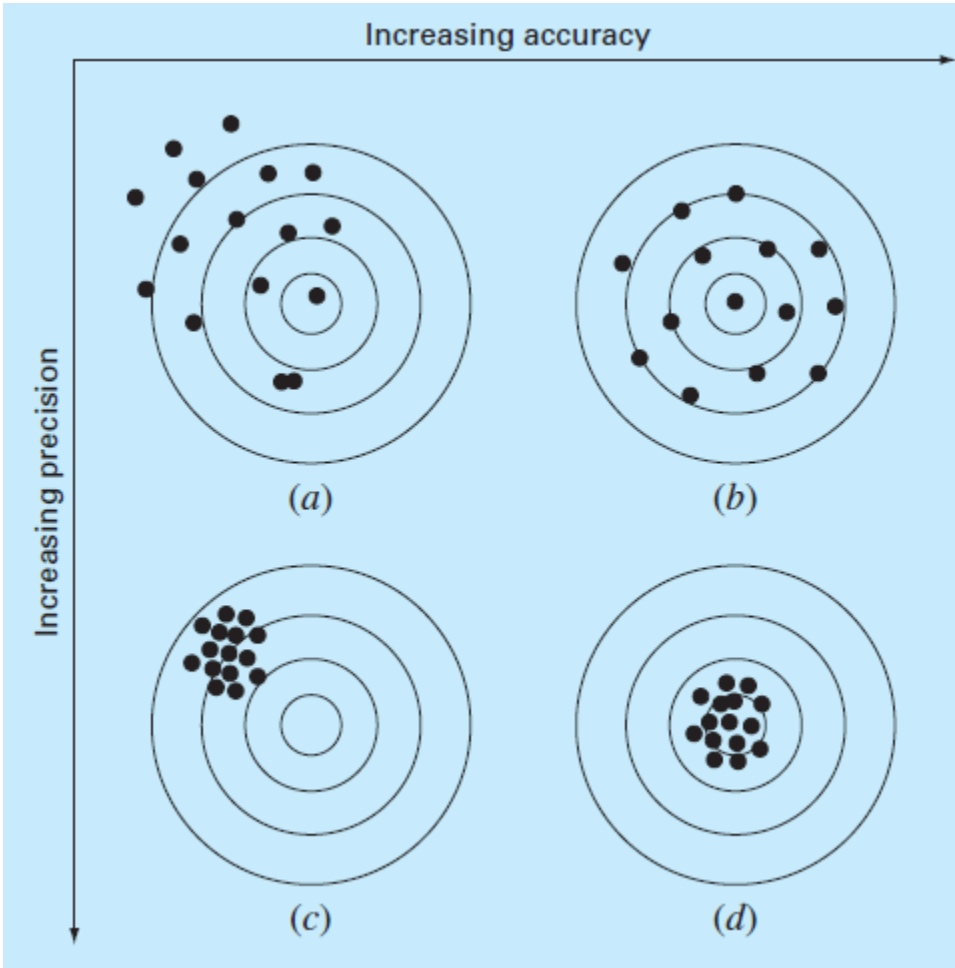


Figura 1.22

1.3.3 – Definição de Erro

- Os erros numéricos são causados pelo uso de aproximações para representar operações e quantidades matemáticas exatas. Eles incluem erros de truncamento que resultam quando são feitas aproximações para representar procedimentos matemáticos exatos, e erros de arredondamento, que aparecem quando números com uma quantidade limitada de algarismos significativos são usados para representar números exatos. Para ambos os tipos, a relação entre o resultado exato ou verdadeiro e a aproximação pode ser formulada como:

$$Valor\ verdadeiro = Aproximação + Erro \tag{32}$$

1.3.3 – Definição de Erro

- Rearranjando a equação (32), tem-se:

$$E_t = valor\ verdadeiro\ - aproximação \tag{33}$$

em que E_t designa o valor exato do erro.

- Um defeito dessa definição é que ela não leva em consideração a ordem de grandeza do valor examinado. Uma forma de considerar o valor das quantidades que estão sendo calculadas é normalizar o erro com o valor verdadeiro:

$$\varepsilon_t(erro\ relativo\ verdadeiro) = \frac{Erro\ verdadeiro}{Valor\ verdadeiro} \tag{34}$$

1.3.3 – Definição de Erro

- Em situações reais, no entanto, os valores verdadeiros não são conhecidos *a priori*. Neste caso, emprega-se uma aproximação dos valores verdadeiros ou seja:

$$\varepsilon_a = \frac{Erro\ aproximado}{Aproximação}(100\%) \qquad (35)$$

Em certos métodos numéricos, utiliza-se uma abordagem iterativa para calcular respostas. Nestes casos, uma aproximação atual é feita com base em uma aproximação prévia. Assim, o erro é frequentemente estimado como a diferença entre as aproximações prévia e corrente

$$\varepsilon_a = \frac{Aproximação\ atual - aproximação\ prévia}{Aproximação\ atual}(100\%) \qquad (36)$$

1.3.3 – Definição de Erro

- Comumente, ao se realizar cálculos, não há preocupação com o sinal do erro, mas interesse em saber se o valor absoluto percentual é menor que uma tolerância percentual pré-definida (ε_s). Nestes casos, os cálculos são repetidos até que:

$$|\varepsilon_a| < \varepsilon_s \tag{37}$$

se essa relação for válida, supõe-se que o resultado está dentro do nível aceitável pré-estabelecido ε_s.

- Também é conveniente relacionar esses erros ao número de algarismos significativos na aproximação. Pode ser mostrado que, se o seguinte critério for satisfeito, é possível ter certeza de que o resultado é correto até pelo menos *n* algarismos significativos:

$$\varepsilon_s = (0{,}5 \times 10^{2-n})(\%) \tag{38}$$

1.3.4 – Erros de Arredondamento

1.3.4.1 – Representação dos números no computador

- Os erros numéricos de arredondamento estão diretamente relacionados à maneira com os números são armazenados no computador. A unidade fundamental na qual a informação é representada é chamada palavra, entidade que consiste em uma sequencia de dígitos binários (BITS). Os números são tipicamente armazenados em uma ou mais palavras.
- Um mesmo numero pode ser expresso em diferentes bases ou sistemas numéricos, bem como em diferentes notações. Uma base numérica define o numero de caracteres básicos para a formação de qualquer numero a ser expresso naquela base. As diferentes notações definem o formato de escrita de um numero em qualquer base.

1.3.4.1 – Representação dos números no computador

- Se um mesmo numero pode ser representado em bases diferentes, é necessário o estabelecimento de um procedimento de conversão do valor de um numero de uma base para outra. O computador utiliza-se desses procedimentos para operar internamente na base 2 e se comunicar com o usuário na base 10.

- Um número em qualquer base é representado por:

$$Z_\alpha = a_m a_{m-1} a_{m-2} \dots b_1 b_2 \dots b_n \qquad (39)$$

onde $\alpha \in N$ representa a base em que o número está representado, os dígitos "a" representam a parte inteira do número e os dígitos "b" a parte fracionária.

1.3.4.1 – Representação dos números no computador

- Nota-se na equação abaixo que quando $\alpha > 10$, há necessidade de utilização de caracteres adicionais para a representação dos dígitos básicos, referente à base escolhida. Os dígitos básicos, δ, de uma determinada base formam o conjunto.

$$B = \{\delta \in \mathbb{N} | 0 \leq \delta \leq \alpha - 1\} \tag{40}$$

Um número na base α é convertido para a base 10 através da seguinte expressão:

$$Z_{10} = a_m \alpha^m a_{m-1} \alpha^{m-1} + \cdots + a_0 \alpha^0 + b_1 \alpha^{-1} + \cdots b_n \alpha^{-n} \tag{41}$$

1.3.4.1 – Representação dos números no computador

- A conversão de um número na base 10 para uma base α é feita primeiramente operando a parte inteira com a seguinte sequência de cálculos:

$$\begin{aligned} R_0 &= (Z_{10})_{INT} - q_1\alpha \\ R_1 &= q_1 - q_2\alpha \\ R_2 &= q_2 - q_3\alpha \\ &\cdots \\ R_P &= q_P - q_{P+1}\alpha \end{aligned} \tag{42}$$

onde q_1 é a parte inteira do quociente da divisão da parte inteira de Z_{10}, $(Z_{10})_{INT}$, por α, q_2 a parte inteira do quociente da divisão de q_1 por α, e assim sucessivamente até que a parte inteira do quociente $q_{P+1} < \alpha$, momento em que o procedimento se encerra. Observa-se que $0 \leq R_i \leq \alpha_{-1} (0 \leq i \leq b)$. A parte inteira do número convertido para a base α é, portanto:

$$q_{P+1} R_P R_{P-1} \cdots R_1 R_0$$

1.3.4.1 – Representação dos números no computador

- A parte fracionária de Z_{10} é convertida para a base α realizando multiplicações sucessivas conforme se segue:

$$\begin{gathered} 0, b_1, b_2 \cdots b_n \times \alpha = S_1, c_1 c_2 \cdots c_n \\ 0, c_1, c_2 \cdots c_n \times \alpha = S_1, d_1 d_2 \cdots d_n \\ 0, d_1, d_2 \cdots d_n \times \alpha = S_1, L_1 L_2 \cdots L_n \\ \cdots \end{gathered} \tag{43}$$

- As multiplicações devem prosseguir até que a parte fracionária do resultado seja 0, se o número for representável exatamente na base α. Caso contrário, o número não é exato na base α e os cálculos devem parar quando for atingido o número de dígitos desejado para o resultado. O resultado final do número convertido na base α é:

$$Z_\alpha = q_{P+1} R_P R_{P-1}, \cdots, R_1 R_0, S_1 S_2 S_3 \tag{44}$$

1.3.4.1 – Representação dos números no computador

- **<u>Exemplo</u>) Converter o número Z_{10} = 12,8 para a base $\alpha = 2$, arredondar o resultado para 5 dígitos fracionários**.

Solução: Inicialmente, deve-se converter a parte inteira, utilizando-se a equação (42):

$$q_1 = \frac{(Z_{10})_{INT}}{\alpha} = \frac{12}{2} = 6;\ R_0 = (Z_{10})_{INT} - q_1\alpha = 12 - 6 \times 2 = 0$$

$$q_2 = \frac{q_1}{\alpha} = \frac{6}{2} = 3;\ R_1 = q_1 - q_2\alpha = 6 - 3 \times 2 = 0$$

$$q_3 = \frac{q_2}{\alpha} = \frac{3}{2} = 1;\ R_2 = q_2 - q_3\alpha = 3 - 1 \times 2 = 1$$

Os cálculos param nesse ponto, uma vez que $q_3 < 2$

1.3.4.1 – Representação dos números no computador

Para a parte fracionária, utiliza-se a equação (43):

$$0{,}8 \times 2 = 1{,}6 \rightarrow S_1 = 1$$
$$0{,}6 \times 2 = 1{,}2 \rightarrow S_2 = 1$$
$$0{,}2 \times 2 = 0{,}4 \rightarrow S_3 = 0$$
$$0{,}4 \times 2 = 0{,}8 \rightarrow S_4 = 0$$
$$0{,}8 \times 2 = 1{,}6 \rightarrow S_5 = 1$$
$$\dots$$

Combinando-se os resultados anteriores, obtém-se:

$$Z_2 = q_3 R_2 R_1 R_0, s_1 s_2 s_3 s_4 s_5 \cdots$$
$$Z_2 = 1100{,}1100110011 \cdots$$

Finalmente, aproximando-se para quatro dígitos fracionários, tem-se:

$$Z_2 = 1100{,}1100 \text{ (truncamento) ou}$$
$$Z_2 = 1100{,}1101 \text{ (arredondamento)}$$

1.3.4.1 – Representação dos números no computador

- Representação Inteira:

A abordagem mais direta é chamada de método dos valores com sinal e utiliza o primeiro bit de uma palavra para indicar o sinal, com um 0 para positivo e 1 para negativo. Os bits restantes são usados para armazenar o número.

1.3.4.1 – Representação dos números no computador

- **<u>Exemplo</u>) Determinar o intervalo dos inteiros na base 10 que podem ser representados em um computador de 16 bits.**

Solução: dos 16 bits, o primeiro bit representa o sinal. Os restantes 15 bits podem conter números binários de 0 a 111111111111111. O limite superior pode ser convertido para um número decimal como

$$Z_{10} = 1 \times 2^{14} + 1 \times 2^{13} + \cdots + 1 \times 2^{1} + 1 \times 2^{0} = 32767$$

Deste modo, uma palavra no computador de 16 bits pode armazenar inteiros decimais variando de -32767 a 32767. Além disso, como o zero já foi definido como 0000000000000000, é redundante usar o número 1000000000000000 para definir um "menos zero". Portanto, ele é normalmente usado para representar um número negativo adicional: -32768, e o intervalo é de -32768 a 32767.

1.3.4.1 – Representação dos números no computador

- Representação em ponto flutuante:

As quantidade fracionárias são representadas tipicamente em computadores usando-se a forma de ponto flutuante. Nessa abordagem, o numero é expresso como uma parte fracionária, chamada mantissa ou significando, e uma parte inteira, chamada de expoente ou característica, como em:

$$m \times \alpha^e$$

onde m é a mantissa, α é a base do sistema numérico que está sendo usado, e e é o expoente:

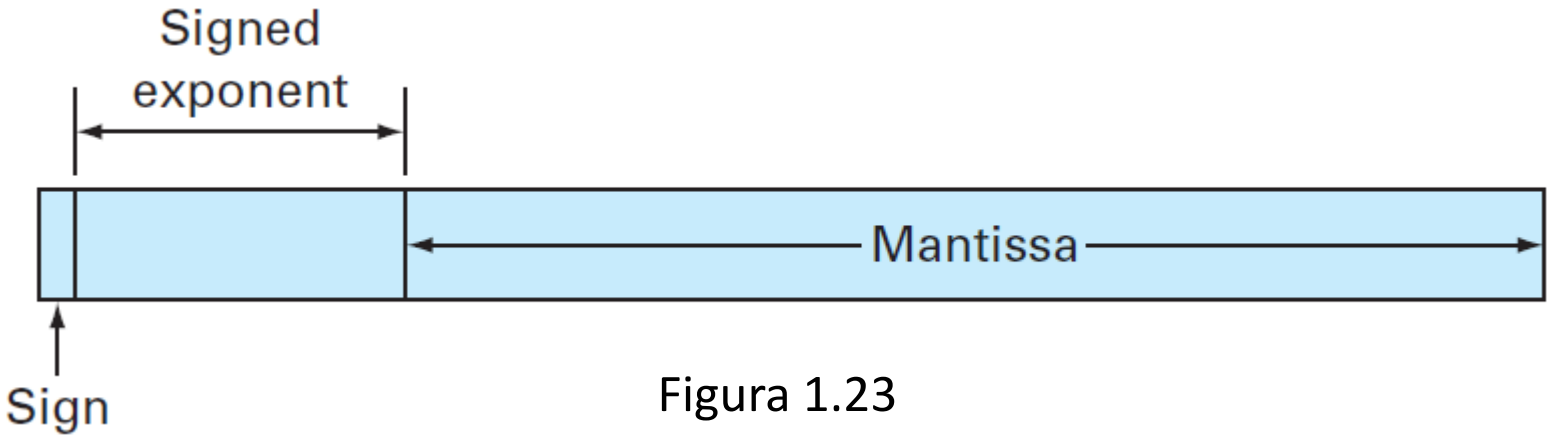


Figura 1.23

1.3.4.1 – Representação dos números no computador

A mantissa normalmente encontra-se normalizada, ou seja, apresenta o algarismo dominante nulo. Com o intuito de se evitar a inclusão de zeros inúteis, no entanto, é usual multiplicar a mantissa por um fator igual a uma potência da base, reduzindo-se então o expoente. Deste modo, o valor absoluto de m é limitado:

$$\frac{1}{\alpha} \leq m < 1 \tag{45}$$

onde α é a base. Por exemplo, em um sistema na base 10, m varia entre 0,1 e 1.

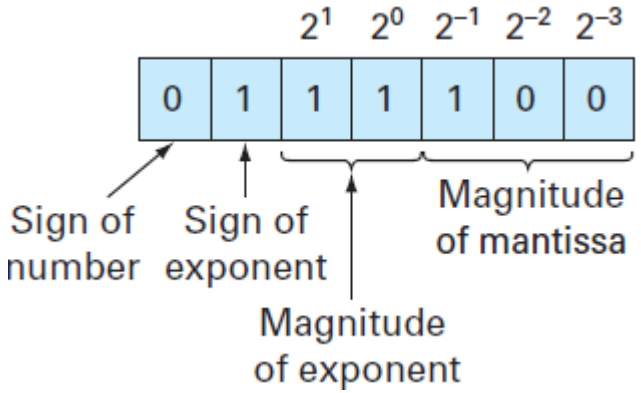
1.3.4.1 – Representação dos números no computador

A representação em ponto flutuante permite que tanto frações quanto números muito grandes sejam expressos em um computador. Ela possui, contudo, algumas desvantagens, como o fato de os números em ponto flutuante ocuparem mais memória e levarem mais tempo de processamento que os números inteiros. Mais significativo, porém, é que seu uso introduz uma fonte de erros, uma vez que a mantissa mantém apenas um número finito de algarismos significativos, sendo introduzido assim um erro de arredondamento.

1.3.4.1 – Representação dos números no computador

- **<u>Exemplo</u>) Criar um conjunto de números hipotéticos em ponto flutuante para uma máquina que armazena informação usando palavras de 7 bits. Usar o primeiro bit para o sinal do número, os próximos três para o sinal e o módulo do expoente e os três últimos para o módulo da mantissa.**

Solução: O menor número positivo é dado por:



0: sinal do número: +

1: sinal do expoente: -

11: módulo do expoente: $(11)_2 = 1x2^1 + 1x2^0 = (3)_{10}$

Assim, o expoente é 3

100: mantissa: $(100)_2 = 1x2^{-1} + 0x2^{-2} + 0x2^{-3} = (0,5)_{10}$

1.3.4.1 – Representação dos números no computador

Embora uma mantissa menor seja possível (000, 001, 010, 011), o valor 100 é usado por causa do limite imposto pela normalização (equação 45). Deste modo, o menor número positivo possível para esse sistema é +0,5x2-3, que é igual a 0,0625 no sistema decimal. Os próximos números mais altos são obtidos aumentando-se a mantissa, como em:

$$0111101 = (1x2^{-1} + 0x2^{-2} + 1x2^{-3}) \times 2^{-3} = (0,078125)_{10}$$

$$0111110 = (1x2^{-1} + 1x2^{-2} + 0x2^{-3}) \times 2^{-3} = (0,093750)_{10}$$

$$0111111 = (1x2^{-1} + 1x2^{-2} + 1x2^{-3}) \times 2^{-3} = (0,109375)_{10}$$

Observa-se que os equivalentes na base 10 são igualmente espaçados, com um intervalo de 0,015625.

1.3.4.1 – Representação dos números no computador

Neste ponto, para continuar a aumentar, é preciso diminuir o módulo do expoente para $(10)_2$, que corresponde a $(10)_2 = 1x2^1 + 0x2^0 = (2)_{10}$. A mantissa é diminuída para o seu valor (100)2. Portanto, o próximo numero é

$0110100 = (1x2^{-1} + 0x2^{-2} + 0x2^{-3}) \times 2^{-2} = (0,125000)_{10}$

Isso representa um salto de 0,125000 - 0,109375 = 0,015625. No entanto, agora quando os números mais altos forem gerados, aumentando-se a mantissa, o espaçamento terá um comprimento de 0,031250

$0110101 = (1x2^{-1} + 0x2^{-2} + 1x2^{-3}) \times 2^{-2} = (0,156250)_{10}$

$0110110 = (1x2^{-1} + 1x2^{-2} + 0x2^{-3}) \times 2^{-2} = (0,187500)_{10}$

$0110111 = (1x2^{-1} + 1x2^{-2} + 1x2^{-3}) \times 2^{-2} = (0,218750)_{10}$

1.3.4.1 – Representação dos números no computador

Esse padrão é repetido conforme quantidades maiores forem representadas até que o número máximo seja atendido:

$$0011111 = (1x2^{-1} + 1x2^{-2} + 1x2^{-3}) \times 2^3 = (7)_{10}$$

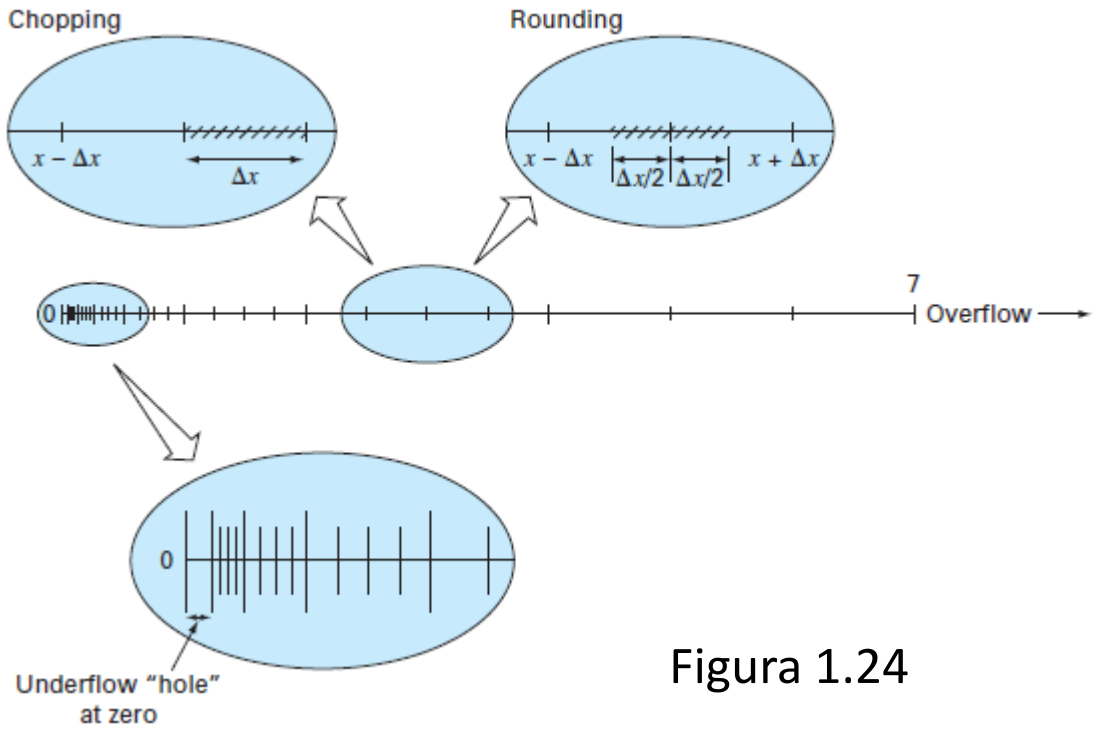


Figura 1.24

1.3.4.1 – Representação dos números no computador

Aspectos da representação em ponto flutuante que são significativos com relação aos erros de arredondamento do computador:

1. Existe um intervalo limitado de quantidades que podem ser representadas. Tentativas de usar números fora do intervalo aceitável vão resultar no que é chamado de erro de *overflow*. Entretanto, além das quantidades grandes, a representação em ponto flutuante tem a limitação adicional de que números muito pequenos também não podem ser representados. Isso é ilustrado pelo "buraco" de *underflow* entre o zero e o primeiro número positivo representável.

1.3.4.1 – Representação dos números no computador

2. Existe apenas um número finito de quantidades que podem ser representadas dentro do intervalo, portanto, o grau de precisão é limitado. Números irracionais e números racionais que não coincidam exatamente com valores existentes no conjunto não podem ser representados precisamente. Os erros introduzidos pela aproximação em ambos os casos são chamados de erros de quantização. A aproximação propriamente dita é feita por truncamento ou por arredondamento.

Supondo que o valor de $\pi = 3{,}14159265358\ldots$ deva ser armazenado em um sistema numérico na base 10, com sete algarismos significativos. Neste caso, tem-se:

$\pi = 3{,}14159265358$ (truncamento); E_t = 0,00000065...

$\pi = 3{,}14159265358$ (arredondamento); E_t = 0,00000034...

1.3.4.1 – Representação dos números no computador

3. O intervalo entre os números, Δx, aumenta quando o módulo dos números cresce. É essa característica que permite que a representação em ponto flutuante preserve os algarismos significativos. Mas ela significa, também, que o erro de quantização será proporcional ao módulo do numero que está sendo representado. Para números com ponto flutuante normalizados, essa proporcionalidade pode ser expressa, para os casos em que é empregado o truncamento como

$$\frac{|\Delta x|}{|x|} \leq \frac{\varepsilon}{2} \tag{46}$$

E para os casos nos quais o arredondamento é usado como:

$$\frac{|\Delta x|}{|x|} \leq \varepsilon \tag{47}$$

onde ε é chamado de épsilon de máquina, que pode ser calculado como

$$\varepsilon = \alpha^{1-t} \tag{48}$$

onde α é a base numérica e t é o número de dígitos na mantissa.

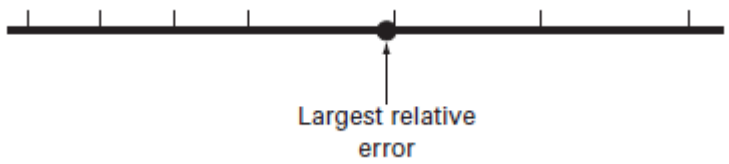
1.3.4.1 – Representação dos números no computador

- **<u>Exemplo</u>) Determinar o épsilon da máquina e verificar sua efetividade na caracterização dos erros do sistema numérico do exemplo anterior. Supor que seja usado truncamento.**

Solução: O sistema em ponto flutuante hipotético do exemplo anterior usa valores de base $\alpha = 2$ e o número de bits da mantissa t=3. Logo o épsilon da máquina seria:

$$\varepsilon = 2^{1-3} = 2^{-2} = 0{,}25$$

Consequentemente, o erro de quantização relativo deveria ser limitado por 0,25 para o truncamento. Os maiores erros relativos deveriam ocorrer para aquelas quantidades que caem imediatamente abaixo do limitante superior do primeiro intervalo entre números igualmente espaçados sucessivos. Aqueles números caindo nos maiores intervalos a seguir teriam o mesmo valor de Δx, mas um valor de x maior e, portanto, um erro relativo maior



1.3.4.1 – Representação dos números no computador

- **<u>Exemplo</u>) Empregando um valor imediatamente menor que o valor limitante do intervalo entre $(0{,}125000)_{10}$ e $(0{,}156250)_{10}$. Para este caso, o erro seria menor que**

Solução:

$$\frac{|0{,}156250 - 0{,}125000|}{|0{,}125000|} = \frac{0{,}031250}{0{,}125000} = 0{,}25$$

1.3.4.2 – Operações Aritméticas Comuns

- Utilizando-se um sistema numérico na base 10, normalizado, e um computador decimal hipotético com uma mantissa de 4 algarismos e um expoente de 1 algarismo. Adota-se, também, truncamento.
- SOMA: Exemplo: $0{,}1551 \times 10^{1} + 0{,}4381 \times 10^{-1}$

Deve-se deslocar a vírgula da mantissa do segundo termo para a esquerda um número de posições igual à diferença dos expoentes [1-(-1)]=2, como em:

$$0{,}4381 \times 10^{-1} = 0{,}004381 \times 10^{-1}$$

Somando-se os números:

$$\begin{array}{r} 0{,}155100 \times 10^{1} \\ + \quad 0{,}004381 \times 10^{1} \\ \hline 0{,}159481 \times 10^{1} \end{array}$$

E o resultado truncado é $0{,}159400 \times 10^{1}$

1.3.4.2 – Operações Aritméticas Comuns

- SUBTRAÇÃO: Supondo que se esteja subtraindo 26,86 de 36,41:

$$\begin{array}{rl} & 0{,}3641 \times 10^2 \\ - & 0{,}2686 \times 10^2 \\ \hline & 0{,}0955 \times 10^2 \end{array}$$

que deve ser normalizado, resultando em 0,9550 x10^1. Neste caso, um zero adicional não significativo é adicionado no final da mantissa. Subtraindo-se dois números muito próximos como:

$$\begin{array}{rl} & 0{,}7642 \times 10^3 \\ - & 0{,}7641 \times 10^3 \\ \hline & 0{,}0001 \times 10^3 \end{array}$$

Resulta em um caso mais dramático, em que 3 zeros não significativos são adicionais à mantissa (o resultado normalizado é 0,1000 x 10^0).

1.3.4.2 – Operações Aritméticas Comuns

- MULTIPLICAÇÃO e DIVISÃO: Na multiplicação, os expoentes são somados e as mantissas multiplicadas na divisão, os expoentes são subtraídos e as mantissas, divididas. Por exemplo,

$$(0{,}1363 \times 10^{3}) \times (0{,}6423 \times 10^{-1}) = 0{,}08754579 \times 10^{2}$$

Normalizando:

$$0{,}8754579 \times 10^{1}$$

E truncando:

$$0{,}8754 \times 10^{1}$$

1.3.4.2 – Operações Aritméticas Comuns

- Operações aritméticas com erros nas parcelas ou fatores:

Seja, x e y dois números tais que $x = \bar{x} + EA_x$ e $y = \bar{y} + EA_y$, onde EA é o erro absoluto e a barra (-) indica o valor aproximado da variável. Neste caso:

a) Adição: x+ y

$$x + y = (\bar{x} + EA_x) + (\bar{y} + EA_y) = (\bar{x} + \bar{y}) + (EA_x + EA_y)$$

Então, o erro absoluto na soma, denotado por EA_{x+y} é a soma dos erros absolutos das parcelas:

$$EA_{x+y} = EA_x + EA_y \tag{49}$$

O erro relativo será:

$$ER_{x+y} = \frac{EA_{x+y}}{\bar{x} + \bar{y}} = \frac{EA_x + EA_y}{\bar{x} + \bar{y}} = \frac{EA_x}{\bar{x}}\left(\frac{\bar{x}}{\bar{x} + \bar{y}}\right) + \frac{EA_y}{\bar{y}}\left(\frac{\bar{y}}{\bar{x} + \bar{y}}\right)$$

$$ER_{x+y} = ER_x\left(\frac{\bar{x}}{\bar{x} + \bar{y}}\right) + ER_y\left(\frac{\bar{y}}{\bar{x} + \bar{y}}\right) \tag{50}$$

1.3.4.2 – Operações Aritméticas Comuns

b) Subtração: x- y

Analogamente, tem-se

$$EA_{x-y} = EA_x - EA_y \tag{51}$$

$$ER_{x-y} = ER_x\left(\frac{\bar{x}}{\bar{x}-\bar{y}}\right) - ER_y\left(\frac{\bar{y}}{\bar{x}-\bar{y}}\right) \tag{52}$$

c) Multiplicação: xy

$$xy = (\bar{x} + EA_x)(\bar{y} + EA_y) = \bar{x}\bar{y} + \bar{x}EA_y + \bar{y}EA_x + (EA_x)(EA_y)$$

Considerando-se que $(EA_x)(EA_y)$ seja um número pequeno, pode-se desprezar tal termo na relação acima, obtendo-se:

$$EA_{xy} = \bar{x}EA_y + \bar{y}EA_x \tag{53}$$

$$ER_{xy} = \frac{\bar{x}EA_y + \bar{y}EA_x}{\bar{x}\bar{y}} = \frac{EA_y}{\bar{y}} + \frac{EA_x}{\bar{x}} = EREA_x - EA_{y_x} + ER_y \tag{54}$$

1.3.4.2 – Operações Aritméticas Comuns

d) Divisão: x / y

$$\frac{x}{y} = \frac{\bar{x} + EA_x}{\bar{y} + EA_y} = \frac{\bar{x} + EA_x}{\bar{y}} \left(\frac{1}{1 + \frac{EA_y}{\bar{y}}} \right)$$

Representando o fator $\frac{1}{1+\frac{EA_y}{\bar{y}}}$ sob a forma de uma série infinita:

$$\frac{1}{1 + \frac{EA_y}{\bar{y}}} = 1 - \frac{EA_y}{\bar{y}} + \left(\frac{EA_y}{\bar{y}} \right)^2 - \left(\frac{EA_y}{\bar{y}} \right)^3 + \cdots$$

E desprezando-se os termos com potências maiores que 1, tem-se:

$$\frac{x}{y} \approx \frac{\bar{x}}{\bar{y}} + \frac{EA_x}{\bar{y}} - \frac{\bar{x} EA_y}{\bar{y}^2}$$

Assim

$$EA_{x/y} = \frac{EA_x}{\bar{y}} - \frac{\bar{x} EA_y}{\bar{y}^2} = \frac{\bar{y} EA_x - \bar{x} EA_y}{\bar{y}^2} \quad (55)$$

e

$$ER_{x/y} \approx \left(\frac{\bar{y} EA_x - \bar{x} EA_y}{\bar{y}^2} \right) \frac{\bar{y}}{\bar{x}} = \frac{EA_x}{\bar{x}} - \frac{EA_y}{\bar{y}} = ER_x - ER_y \quad (56)$$

1.3.4.2 – Operações Aritméticas Comuns

- Perda de algarismos significativos:

Exemplo) $y = \sqrt{x^2 + 1} - 1$, com x muito pequeno

Supondo $x = 0{,}1000 \times 10^{-1}$, tem-se $x^2 = xx = 0{,}1000 \times 10^{-3}$ e $x^2 + 1 = 0{,}1000 \times 10^{1}$, que truncado resulta em $0{,}1000 \times 10^{1}$. Extraindo-se a raiz quadrada, tem-se $\sqrt{x^2 + 1} = 0{,}1000 \times 10^{1}$ e o resultado de $y = \sqrt{x^2 + 1} - 1$ é 0,000.

Manipulação de y:

$$y = \left(\sqrt{x^2 + 1} - 1\right)\frac{\left(\sqrt{x^2 + 1} + 1\right)}{\left(\sqrt{x^2 + 1} + 1\right)} = \frac{\left(\sqrt{x^2 + 1}\right)^2 - 1^2}{\sqrt{x^2 + 1} + 1} = \frac{x^2}{\sqrt{x^2 + 1} + 1}$$

Neste caso, para $x = 0{,}1000 \times 10^{-1}$, $x^2 = 0{,}1000 \times 10^{-3}$ e $x^2 + 1 = 0{,}1000 \times 10^{1}$, que truncado resulta em $0{,}1000 \times 10^{1}$. Extraindo a raiz quadrada, tem-se $\sqrt{x^2 + 1} = 0{,}2000 \times 10^{1}$. Realizando a divisão de x^2 por $\left(\sqrt{x^2 + 1} + 1\right)$, tem-se $y = 0{,}5000 \times 10^{-5}$.

Solução real: $y \approx 0{,}499988 \times 10^{-5}$