

Capítulo 3

Objetivos Específicos de Aprendizagem

Ao finalizar este Capítulo, você será capaz de:

- ▶ Determinar valores iniciais para raízes de uma equação não linear $f(x) = 0$;
- ▶ Determinar raízes α que satisfaçam uma equação $f(x) = 0$ por métodos de quebra e de linearização (iterativos);
- ▶ Determinar todas as raízes, reais e/ou complexas, para equações polinomiais, considerando os efeitos da multiplicidade;
- ▶ Avaliar a precisão do resultado obtido; e
- ▶ Utilizar os algoritmos disponibilizados.



3 Solução de Equações não Lineares a uma Incógnita

Em muitos modelos matemáticos de várias áreas, como engenharia, economia, ocorre a necessidade de determinar uma incógnita x que satisfaça uma equação $f(x) = 0$. Esse é um problema que ocupa os matemáticos há milênios. Apenas como exemplo emblemático, destacamos os papiros do escriba egípcio Ahmes (1680 a 1620 a.C.), nos quais foram encontradas aproximadamente 80 equações propostas, bem como a denominação dada por ele ao seu “estudo” sobre as equações: “Direção para saber todas as coisas obscuras”.

Para iniciar este Capítulo, vamos apresentar três definições básicas importantes:

Definição 1: **solução** ou **raiz** da equação $f(x) = 0$ é todo $\alpha \in \mathbb{C}$ (Complexos), tal que $f(\alpha) = 0$.

Definição 2: **zero** da função $f(x)$ é todo $\alpha \in \mathbb{C}$ (Complexos), tal que $f(\alpha) = 0$.

Por exemplo:

- a) $x^3 - 2x^2 + 2x = 0 \Rightarrow \alpha_1 = 0, \alpha_2 = 1 - i, \alpha_3 = 1 + i$
- b) $e^{3x^3} - 2 = 0 \Rightarrow 3x^3 \ln(e) = \ln(2) \Rightarrow \alpha = \sqrt[3]{\ln(2)/3}$
- c) $\sin(x) = 1 \Rightarrow \alpha_k = 2k\pi + \pi/2$ com $k \in \mathbb{Z}$
- d) $4\cos(x) = e^x \Rightarrow \alpha_1 = -1.5158641228050098$ e $\alpha_2 = 0.9047882178730189$
- e) $e^{2x} = -3 \Rightarrow$ não existe α

Note que nem sempre é possível solver a equação de forma explícita, ou seja, nem sempre conseguimos isolar a incógnita em um dos lados da equação para obter explicitamente uma raiz α .

Pelas equações dadas anteriormente, concluímos que:

- a) Uma equação $f(x) = 0$ pode:
 - i) não ter solução;
 - ii) ter única solução;
 - iii) ter uma quantidade finita de soluções; e
 - iv) ter uma quantidade infinita de soluções.
- b) A solução de $f(x) = 0$, pela técnica do isolamento da incógnita, pode ser simples, difícil ou até impossível de ser obtida.
- c) Resolver equações exige conhecimento de outras **metodologias**, além da forma de isolamento da incógnita.

Definição 3: um **método iterativo** obedece sempre a duas etapas em sua execução:

- a) isolamento, ou localização, de uma região do domínio da função geradora $y = f(x)$ que contenha cada solução desejada (aproximação inicial da raiz); e
- b) refinamento da solução isolada até a precisão requerida.

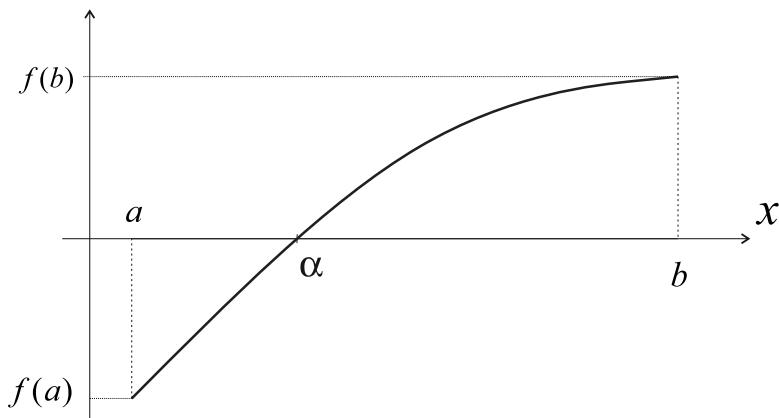
Neste livro, usaremos metodologias iterativas para obter aproximações de soluções Reais e Complexas, com destaque para a metodologia proposta por Newton, conforme encontramos nas obras de Burden e Faires (2011) e de Cheney e Kincaid (2008).



3.1 Isolamento ou localização de Soluções de $f(x) = 0$

Podemos efetivar o isolamento, ou a localização, de uma região do domínio da função geradora que contenha a solução desejada da equação $f(x) = 0$ através de:

- a) Conhecimento prático sobre o problema modelado que resultou na equação, com a consequente estimativa da solução esperada.
- b) Traçado do esboço gráfico da função geradora $y = f(x)$, uma vez que toda raiz **Real** é uma interseção desse gráfico com o eixo das abscissas x , conforme o Gráfico 3.1.

Gráfico 3.1 – Intervalo que contém uma raiz Real α de $f(x) = 0$ 

Fonte: Elaboração própria

- c) Teorema de Bolzano: numa $f(x) = 0$, se $y = f(x)$ for contínua em $[a,b]$ e $f(a) * f(b) \leq 0$ então $\exists \alpha \in [a,b] / f(\alpha) = 0$.

Note que esse teorema fornece uma condição suficiente, mas não necessária, para a existência de soluções reais, isto é, se satisfeita a condição, existirá solução, senão nada poderemos afirmar. Além disso, a solução em $[a,b]$ pode não ser única.

- d) Agrupamento das funções geradoras, $y = f(x)$, em classes com características especiais como as funções polinomiais, por exemplo, e utilização das suas propriedades algébricas, como veremos na seção 3.3.

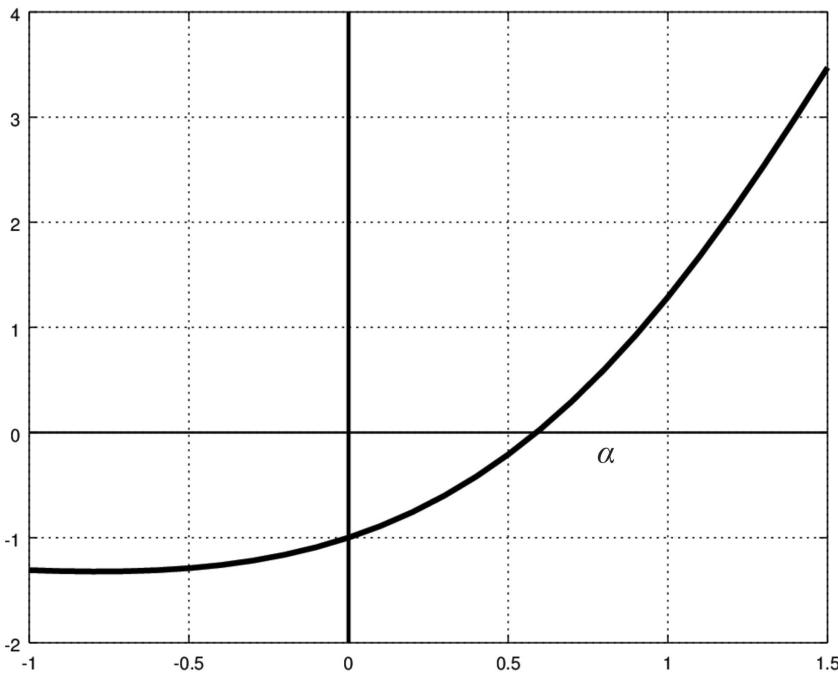
Vamos primeiramente localizar raízes de equações quaisquer, conforme

o **Exemplo 3.1**.

Exemplo 3.1: isole, ou localize, um intervalo que contenha a primeira raiz positiva α de $e^x \sin(x) - 1 = 0$.

Solução:

Observe, no Gráfico 3.2, que a raiz está localizada no intervalo $[0, 1]$ ou $[0.5, 1]$, pois $f(x) = e^x \sin(x) - 1$ corta o eixo da abcissa x (com $y = 0$) nesses intervalos:

Gráfico 3.2 – Esboço da $f(x) = e^x \sen(x) - 1$ 

Fonte: Elaboração própria

Também, como a $f(x) = e^x \sen(x) - 1$ é contínua em toda reta Real, e no intervalo $[0, 1]$ verificamos que $f(0) = -1$ e $f(1) = +1.287$, ou seja, $f(0) * f(1) < 0$, pelo teorema de Bolzano, temos assegurado que existe raiz $\alpha \in [0, 1]$.

Tratada a questão da localização de uma solução de $f(x) = 0$, vamos partir para o processo de refinamento da solução isolada.

3.2 Refinamento da Solução Isolada

Depois de isolar uma raiz α em um intervalo $[a,b]$ do domínio de $f(x)$, devemos refiná-la por meio de técnicas específicas, para melhorar a sua precisão. Essas técnicas podem ser agrupadas em três grandes famílias de métodos:

- a) **Métodos de quebra:** para os quais inicialmente temos de obter um intervalo $[a,b]$, tal que $\alpha \in [a,b]$, e o particionar em uma sequência de subintervalos menores $[a_k, b_k]$, mas que continuem contendo α , tal que $\lim_{k \rightarrow \infty} (|b_k - a_k|) = 0$. Esses métodos têm **convergência garantida**, mas normalmente são lentos.
- b) **Métodos de linearização:** para os quais obtemos uma aproximação inicial x_0 para a raiz α (x_0 valor estimado na etapa de isolamento) e posteriormente construímos uma sequência iterativa de valores:

$$\{x_k\}_{k=0}^{\infty} / \lim_{k \rightarrow \infty} (x_k) = \alpha, \text{ se a sequência for convergente.}$$

Para gerar essa sequência iterativa, linearizamos a equação não linear $f(x) = 0$, resultando em $x = F(x)$, conforme veremos na seção 3.2.2; obtemos o valor $x_1 = F(x_0)$; repetimos o processo com o $x_2 = F(x_1)$; e assim sucessivamente:

$$x_{k+1} = F(x_k)$$



Devido à sua complexidade, os métodos híbridos não fazem parte do escopo deste livro.

Esses métodos **não têm convergência garantida**, mas podem ter convergência rápida, dependendo da forma iterativa estabelecida.

- c) **Métodos híbridos:** consistem na mescla de métodos das duas metodologias citadas anteriormente, tentando associar a vantagem da convergência garantida na primeira com a alta velocidade de convergência da segunda.

A seguir, vamos apresentar três tipos de métodos da primeira família.

3.2.1 Métodos de Quebra

3.2.1.1 Método da Bissecção (ou Bipartição) para Raízes Reais

Em $f(x)=0$, se $f(x)$ for contínua em $[a,b]$ e ocorrer $f(a)*f(b) \leq 0$, então existirá uma raiz real α dentro desse intervalo, $\alpha \in [a,b]$. Logo, uma aproximação para essa raiz pode ser tomada como o valor médio $\bar{x}_1 = \frac{a + b}{2}$.

Assim, o algoritmo do método da bisseção é estabelecido com o objetivo de:

- a) reduzir o intervalo inicial $[a,b]$ a um subintervalo tão pequeno quanto necessário, porém contendo a solução α ; e
- b) obter a solução aproximada contida nesse subintervalo da seguinte forma:
 - i) obtemos o primeiro valor médio do intervalo:

$$\bar{x}_1 = \frac{a + b}{2}$$

então, temos três valores de x : a , b e \bar{x}_1 ;

- ii) redefinimos um novo intervalo $[a,b]$, com metade do comprimento inicial, considerando que:

$$\text{se } f(\bar{x}_1) = 0 \Rightarrow \alpha = \bar{x}_1$$

senão verificamos em qual subintervalo, $[a, \bar{x}_1]$ ou $[\bar{x}_1, b]$, α está:

$$\begin{cases} \text{se } f(a)*f(\bar{x}_1) < 0 \Rightarrow \alpha \in [a, \bar{x}_1], \quad b = \bar{x}_1 \text{ e } f(b) = f(\bar{x}_1) \\ \text{senão } f(\bar{x}_1)*f(b) < 0 \Rightarrow \alpha \in [\bar{x}_1, b], \quad a = \bar{x}_1 \text{ e } f(a) = f(\bar{x}_1) \end{cases}$$

Uma vez redefinido o novo intervalo, calculamos um segundo valor médio \bar{x}_2 :

$$\bar{x}_2 = \frac{a + b}{2}$$

então, temos três novos valores de x : a , b e \bar{x}_2 ;

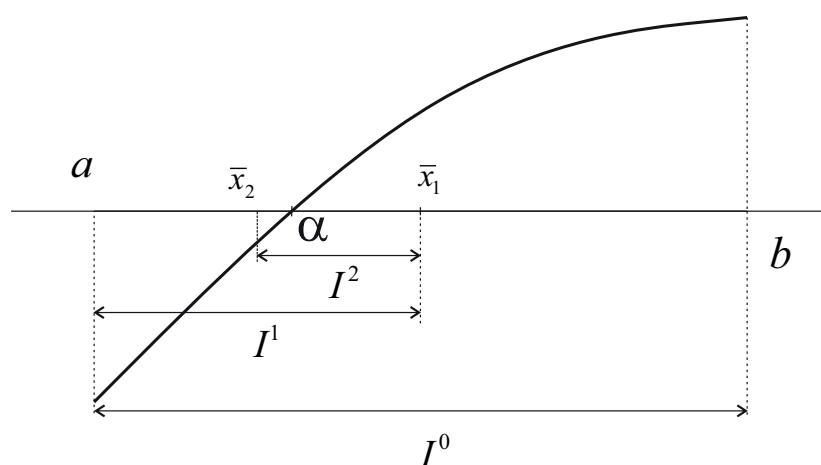
- iii) com esses novos a , b e \bar{x}_2 , voltamos ao passo (ii), considerando $\bar{x}_1 = \bar{x}_2$, enquanto algum critério de parada não for satisfeito.

No final, o último x médio, calculado sobre o último intervalo $[a,b]$, é a melhor aproximação para a raiz, $\alpha \equiv (a + b)/2$, podendo afirmar apenas que a raiz α está dentro desse último subintervalo, $\alpha \in [a,b]$, sendo o comprimento desse subintervalo uma medida da precisão da raiz desejada.

A **bisseção** é um método usado frequentemente para obter a primeira aproximação da solução. Esse método também é chamado de **método da busca binária**, ou **método da dicotomia**.

No exemplo ilustrado no Gráfico 3.3, $\alpha \in [a, \bar{x}_1] \Rightarrow b = \bar{x}_1$, o novo valor de b deve ser redefinido com o primeiro valor médio, \bar{x}_1 , para que α fique contido no primeiro intervalo $I^1 = [a, \bar{x}_1]$, que é menor do que o intervalo inicial $I^0 = [a, b]$. Assim, repetimos o processo de partições, avaliando os valores extremos e médio de cada intervalo particionado I^k , com $k = 1, 2, 3, 4, \dots$, até que algum critério de parada seja satisfeito, por exemplo, $|b - a| < \varepsilon$, com os últimos a e b , de acordo com a precisão desejada.

Gráfico 3.3 – Método da bisseção, ou bipartição, do intervalo $[a,b]$



Fonte: Elaboração própria

Neste ponto, podemos levantar uma questão fundamental: será que a sequência de valores médios \bar{x}_k converge para a raiz α , isto é,
 $\lim_{k \rightarrow \infty} (\bar{x}_k) = \alpha$?

Note que o comprimento de cada novo subintervalo é $|b_k - a_k| = \frac{|b-a|}{2^k}$, em que a e b são os valores extremos do intervalo inicial que contém a raiz α e k indica o número de partições.

Como $\alpha \in [a_k, b_k]$ e $\lim_{k \rightarrow \infty} |b_k - a_k| = \lim_{k \rightarrow \infty} \left(\frac{|b-a|}{2^k} \right) = 0 \Rightarrow$ o processo é convergente.

Então, o ponto médio do último subintervalo será uma aproximação da solução exata α com erro não superior ao comprimento desse subintervalo. Teoricamente, a raiz α será atingida quando o número k de partições tender ao infinito.

Exemplo 3.2: obtenha a primeira raiz positiva α de $e^x \sin(x) - 1 = 0$, situada em $[0,1]$, por bisseção, com $n = 5$ partições do intervalo $[0,1]$ (n denota o número total de partições).

Solução:

A partir do intervalo inicial $I^0 = [0,1]$, $k = 0$, geramos o \bar{x}_k médio inicial e fazemos a seleção para definir o primeiro intervalo particionado $I^1 = [0.5, 1]$ (em negrito), pois houve troca de sinais de $f(x) = e^x \sin(x) - 1$ entre \bar{x}_k e b . Em $k = 1$, partimos desse primeiro intervalo particionado $I^1 = [0.5, 1]$ e, a cada nova partição k , estabelecemos um novo intervalo $[a, b]$ e uma nova aproximação \bar{x}_k para a raiz $\alpha \in [a, b]$.

Tabela 3.1 – Tabela de resultados do Exemplo 3.2

k	a	$\bar{x}_k = (a+b)/2$	b	$f(a)$	$f(\bar{x}_k)$	$f(b)$	$ b-a $	2^k
0	0	0.5	1	-1	-0.21	+1.287	1.0	1
1	0.5	0.75	1	-0.21	+0.443	+1.287	0.5	2
2	0.5	0.625	0.75	-0.21	+0.093	+0.443	0.25	4
3	0.5	0.5625	0.625	-0.21	-0.064	+0.093	0.125	8
4	0.5625	0.59375	0.625	-0.064	+0.013	+0.093	0.0625	16
5	0.5625	0.578125	0.59375	-0.064	-0.02584	+0.013	0.03125	32

Fonte: Elaboração própria

Ao final de $n = 5$ partições, a raiz α está contida no intervalo final $[a, b] = [0.5625, 0.59375]$, também grifado em negrito, de comprimento $(1 - 0) / 32$. Normalmente, a melhor raiz aproximada é a média entre a e b finais da última partição, neste caso seria $\bar{x}_5 = 0.578125$. No algoritmo apresentado, consideramos uma

partição k completa quando os valores extremos do intervalo particionado $[a, b]$ e o seu valor \bar{x}_k médio estão definidos.

O critério de parada baseado no comprimento do intervalo final da partição $|b-a|$ é o mais utilizado nesse método e, neste caso, depois de 5 partições, temos $|b-a| = 0.03125$, ou seja, a raiz α é aproximada dentro de um intervalo de comprimento máximo 0.03125, que é de **ordem $O(10^{-2})$** , logo \bar{x}_5 tem **dígitos convergidos** somente até o primeiro dígito decimal, $\bar{x}_5 \approx \underline{0.5}78125$ (sublinhados).



Ordem de Grandeza dos Números

– Um número $x = m * 10^n$ em ponto flutuante é de ordem de grandeza $O(10^n)$ se o multiplicador $m \in [0, 1]$ estiver entre 10^0 e $10^{1/2}$, ou seja, se m estiver entre 1 e $\sqrt{10} \approx 3.1622\ldots$, e $x = m * 10^n$ será de ordem $O(10^{n+1})$ se $m \in [10^{1/2}, 10^1]$.

Por exemplo:

$x = 2.12345 * 10^{-6}$ é da ordem $O(10^{-6})$

$x = 5.12345 * 10^{-6}$ é considerado da ordem $O(10^{-5})$



Dígito convergido – é aquele que não mais sofrerá alteração durante as iterações.

E quando devemos parar o processo de partições sucessivas?

Podemos predefinir a quantidade n de bipartiçãoes por meio da previsão do comprimento do último subintervalo $|b^{(n)} - a^{(n)}|$ desejado, dado por:

$$|b^{(n)} - a^{(n)}| = \frac{|b - a|}{2^n} \approx \varepsilon \Rightarrow n = \ln\left(\frac{|b - a|}{\varepsilon}\right) / \ln(2) \quad (1)$$

Então, depois da n -ésima bipartição, o comprimento do subintervalo será da ordem de ε com a raiz α contida nesse último subintervalo $[b^{(n)}, a^{(n)}]$.

Exemplo 3.3: calcule o número n de bipartiçãoes necessárias para aproximar $\alpha \in [0,1]$ com precisão $\varepsilon = 10^{-10}$, ou seja, com comprimento máximo do intervalo final menor ou igual a 10^{-10} .

Solução:

$$\text{Aplicando a eq. (1): } n = \ln\left(\frac{1-0}{10^{-10}}\right) / \ln(2) = 33.2 \Rightarrow n = 34.$$

Ou seja, serão necessárias 34 partições para que o último intervalo que contenha a raiz α seja menor de 10^{-10} e que a raiz α esteja exata até o seu décimo dígito significativo, depois do ponto (vírgula).

Se a quantidade n de bipartiçãoes não for previamente determinada, quando deveremos interromper, ou truncar, o processo de bipartiçãoes para assegurar certa precisão estabelecida? Ou seja, qual é realmente o erro de truncamento existente em uma solução \bar{x}_n aproximada em n partições?

A maioria das soluções tem erros de arredondamento e de truncamento. Como já vimos nos capítulos anteriores, o arredondamento pode ser minimizado se usarmos precisão maior, por isso vamos efetuar todos os cálculos com a variável *double*, padrão real IEEE de 64 bits.

No cálculo do **erro de truncamento**, precisamos de uma solução exata “estimada” x_e para comparar com a nossa solução aproximada \bar{x}_n , obtida depois de n partições, pois as soluções exatas não estão disponíveis. Então,

$$\text{Erro truncamento estimado de } \bar{x}_n = |\bar{x}_n - x_e| \quad (2)$$



 Esse mesmo
método
deve
ter sido
testado
previamente
com alguma solução
exata conhecida.

Uma solução “exata” x_e estimada deve ter mais dígitos significativos exatos do que \bar{x}_n , e normalmente podemos obtê-la aplicando o **mesmo método** de aproximações numéricas, mas com mais repetições, então x_e pode ser:

- a) o **valor numericamente exato** de uma solução aproximada com até **infinitas partições**; ou
 - b) o **valor aproximado** com um número mínimo de partições a mais do que o do valor aproximado, como o **dobro de partições**, por exemplo.

$$a = 0.5885327439818608$$

$$b = 0.5885327439818613$$

$$\bar{x}_{51} = 0.5885327439818611 \quad \Rightarrow \quad f(\bar{x}_{51}) = 0$$

A raiz \bar{x}_{51} encontrada atinge $f(\bar{x}_{51})$ numericamente nula, com 16 dígitos significativos exatos, ou seja, é uma raiz aproximada no limite da precisão da variável *double* adotada.

A raiz \bar{x}_{10} encontrada com o dobro de iterações é $\bar{x}_{10} = 0.58837890625$, obtida com critério de parada $|b - a| = 0.0009765625$.

O erro de truncamento da nossa solução aproximada $\bar{x}_5 = 0.578125$, obtida depois de $n = 5$ partições e com critério de parada $|b - a| = 0.03125$, é o seguinte:

Erro de $\bar{x}_s = |\bar{x}_s - \bar{x}_{s1}| = |0.578125 - 0.588532743981861| = 0.010407743981861$
comparando com \bar{x}_{s1} .

Erro de $\bar{x}_5 = |\bar{x}_5 - \bar{x}_{10}| = |0.578125 - 0.58837890625| = 0.01025390625$
comparando com \bar{x}_{10} .

O erro de truncamento mais exato possível de \bar{x}_5 é 0.0104077439818611, obtido por comparação com \bar{x}_{51} , e o erro estimado por comparação com \bar{x}_{10} é 0.01025390625 e ambos são da mesma ordem de grandeza $O(10^{-2})$.

Assim, a solução aproximada $\bar{x}_5 = 0.578125$ tem erro de truncamento ≈ 0.01 , enquanto o critério de parada é maior, mas da mesma ordem $O(10^{-2})$, $|b-a|=0.03125$.

Portanto, é razoável estimar o valor “exato” de uma raiz com o mesmo método approximador fazendo o **dobro** de partições, para então calcular o erro de truncamento exato estimado:

$$\text{Erro truncamento estimado de } \bar{x}_n = |\bar{x}_n - \bar{x}_{2n}| \quad (3)$$

Observe que no **Exemplo 3.2** os critérios de parada alternativos $|f(\bar{x}_5)|=0.0258393$ ou $|\bar{x}_5 - \bar{x}_4|<0.015625$ também são de ordem de grandeza semelhante à ordem do critério de parada utilizado, $|b-a|=0.03125$, e todos são superiores ao erro de truncamento real estimado.

Normalmente, podemos usar esses critérios de parada como limite superior do erro de truncamento, pois, uma vez que eles tenham atingido valores aceitáveis, os erros de truncamento devem ser ainda menores.

Como mencionamos no Capítulo 2, para cada método discutido ao longo desta obra, vamos apresentar todos os algoritmos compactados no **Caderno de Algoritmos** disponível para **download** no link <<http://sergiopeters.prof.ufsc.br/algoritmos-livro/>>, indicando o arquivo extensão **.m** do algoritmo correspondente ao método ou exemplo apresentado, como no arquivo **Cap3MetodosParticao.m**, que contém o método da bisseção e demais métodos de partição deste Capítulo. Lembre-se sempre de testá-los no momento em que o indicamos para depois continuar a sua leitura.

Vimos no Capítulo 2 que a estimativa de um valor “exato” de uma solução também pode ser obtida adotando critério de parada com o dobro de precisão (critério de parada ε^2).



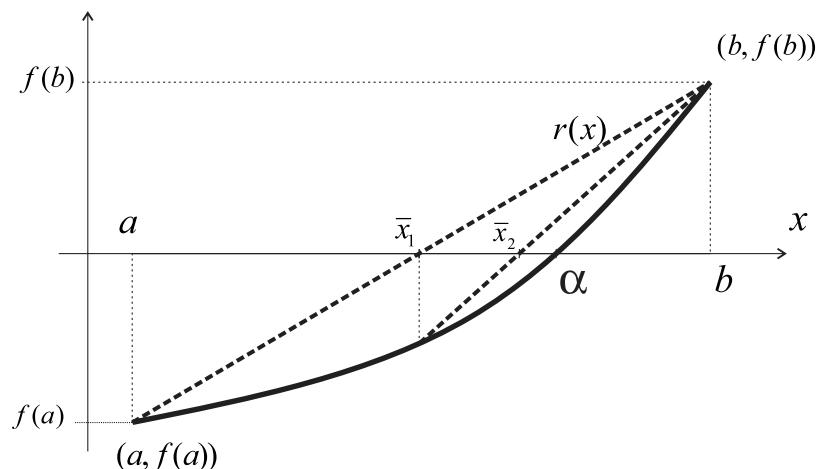
3.2.1.2 Método da Falsa Posição

Mesmo intuitivamente, percebemos que o método da bisseção é lento, pois reduz o intervalo de busca da raiz α em apenas 50% a cada partição.

Então, como particionar o intervalo $[a, b]$ para obter uma convergência mais rápida para a raiz α ?

No método da **falsa posição**, particionamos cada intervalo por meio de um novo valor de \bar{x}_k , que é a interseção entre uma reta $r(x)$ e o eixo das abscissas x , conforme o Gráfico 3.4.

Gráfico 3.4 – Método da falsa posição



Fonte: Elaboração própria

O método da falsa posição consiste nas seguintes etapas:

- Tomar um intervalo inicial $[a,b]$, com $\alpha \in [a,b]$, na etapa de localização.
- Obter os pontos extremos do intervalo $(a, f(a))$ e $(b, f(b))$.
- Definir a reta $r(x)$ que passa por esses pontos extremos:

$$r(x) = f(a) + \left(\frac{f(b) - f(a)}{b - a} \right) (x - a) \quad (4)$$

d) Calcular o valor do novo \bar{x}_k , que partitiona o intervalo, de modo que $r(\bar{x}_k) = 0$, ou seja, \bar{x}_k é a interseção entre a reta $r(x)$ e o eixo das abscissas x e não é uma média aritmética simples do intervalo, como no método da bisseção. Esse valor de \bar{x}_k é equivalente a uma média ponderada pelos valores das funções nas extremidades a e b :

$$r(\bar{x}_k) = 0 \Rightarrow \bar{x}_k = a - \frac{f(a)*(b-a)}{(f(b)-f(a))} \quad (5)$$

Alternativamente, também usamos a forma

$$\bar{x}_k = \frac{a*f(b) - b*f(a)}{f(b) - f(a)} \text{ (com uma multiplicação a mais).}$$

Então, a partir de a , b e \bar{x}_k , procedemos como no método da bisseção.

e) Se $f(\bar{x}_k) = 0 \Rightarrow \alpha = \bar{x}_k$.

Senão, pelo teorema de Bolzano, decidiremos onde a raiz α ficou, se em $[a, \bar{x}_k]$ ou se em $[\bar{x}_k, b]$,

$$\begin{cases} \text{se } f(a)*f(\bar{x}_k) < 0 \Rightarrow \alpha \in [a, \bar{x}_k], \quad b = \bar{x}_k \text{ e } f(b) = f(\bar{x}_k) \\ \text{senão } f(\bar{x}_k)*f(b) < 0 \Rightarrow \alpha \in [\bar{x}_k, b], \quad a = \bar{x}_k \text{ e } f(a) = f(\bar{x}_k) \end{cases}$$

Redefinimos o novo intervalo que contém a raiz $\alpha \in [a, b]$ e os valores das funções nesses pontos.

f) Com o novo intervalo atualizado $[a, b]$, recalculamos o novo \bar{x}_k para a próxima partição:

$$\bar{x}_k = a - \frac{f(a)*(b-a)}{(f(b)-f(a))}$$

Com a , b e \bar{x}_k , retornamos ao item (e), enquanto algum critério de parada não for satisfeito, análogo ao método da bisseção.

g) No final, o valor do \bar{x}_n calculado sobre o último intervalo, depois de n partições k do intervalo inicial, é considerado a melhor aproximação para a raiz.

O critério de parada baseado no valor do intervalo final $|b-a|$ não é indicado se uma das extremidades a ou b ficar fixa, como

apresentamos no Gráfico 3.4, no qual o ponto $x = b$ inicial fica fixo ao longo das iterações.

O **critério de parada** do método da **falsa posição** pode ser baseado no valor das diferenças entre a raiz aproximada a cada partição \bar{x}_k e o seu valor anterior \bar{x}_{k-1} , $|\bar{x}_k - \bar{x}_{k-1}| < \varepsilon$, ou no valor da função em \bar{x}_k , $|f(\bar{x}_k)| < \varepsilon$, ou uma combinação de ambas.

Veremos outros critérios de parada na seção 3.2.2.

Exemplo 3.4: obtenha a primeira raiz positiva α de $e^x \sin(x) - 1 = 0$, situada em $[0, 1]$, pelo método da falsa posição, com 5 partições, e calcule o erro de truncamento atingido.

Solução:

Tabela 3.2 – Tabela de resultados do Exemplo 3.4

k	a	$\bar{x}_k = a - f(a)(b-a) / (f(b)-f(a))$	b	$f(a)$	$f(\bar{x}_k)$	$f(b)$	$ \bar{x}_k - \bar{x}_{k-1} $
0	0	0.43719	1.0	-1.0	-0.34444	+1.28736	-
1	0.43719	0.555986	1.0	-0.34444	-0.079729	+1.28736	0.118800
2	0.555986	0.581881	1.0	-0.079729	-0.016551	+1.28736	0.0258952
3	0.581881	0.5871886	1.0	-0.016551	-0.0033553	+1.28736	0.0053073
4	0.5871886	0.5882617	1.0	-0.0033553	-0.00067690	+1.28736	0.0010731
5	0.5882617	0.5884781340	1.0	-0.00067690	-0.00013643	+1.28736	0.0002163821

Fonte: Elaboração própria

Observe que aplicamos o teorema de Bolzano para redefinir cada novo intervalo $[a, b]$, conforme a troca de sinais das funções em a , \bar{x}_k e b . Em $k = 5$ partições ponderadas do intervalo inicial, atingimos o intervalo final $[0.5882617, 1.0]$, e a raiz aproximada dentro desse intervalo deve ser $\bar{x}_5 \approx 0.5884781340$, com critério de parada $|\bar{x}_k - \bar{x}_{k-1}| < 0.0002163821$, ambos destacados em negrito.

No algoritmo apresentado, também consideramos uma partição k completa quando os valores extremos do intervalo particionado são redefinidos e o respectivo valor do \bar{x}_k do intervalo é calculado. O último \bar{x}_k é considerado a raiz aproximada e o seu erro de truncamento pode ser estimado como fizemos no método da bisseção, comparando $\bar{x}_5 = 0.5884781340$ com $\bar{x}_{10} = 0.5885327258$:

$$\text{Erro de } \bar{x}_5 = |\bar{x}_5 - \bar{x}_{10}| = |0.5884781340 - 0.5885327258| = 0.0000545918$$

Para a validação desse erro, podemos atingir o valor aproximado mais exato possível, no limite do critério de parada para 16 dígitos significativos da variável *double*, em 23 partições, $\bar{x}_{23} = 0.5885327439818610$, enquanto no método da bisseção precisamos de 51 partições.

$$\text{Erro de } \bar{x}_5 = |\bar{x}_5 - \bar{x}_{23}| = |0.5884781340 - 0.5885327439818610|$$

$$\text{Erro de } \bar{x}_5 = |\bar{x}_5 - \bar{x}_{23}| = 0.000054609981861$$

Observe que o erro de truncamento estimado com o dobro de partições também gera uma ótima estimativa para o erro de truncamento.

No método da falsa posição, o valor do critério de parada adotado $|\bar{x}_k - \bar{x}_{k-1}| < 0.00021638$ ($O(10^{-4})$) também é superior ao erro de truncamento estimado, 0.000054609981861 ($O(10^{-4})$), logo o critério de parada pode ser adotado como limite superior do erro de truncamento. Observe, ainda, que o critério $|f(\bar{x}_k)| = 0.000136426$ também é da ordem de grandeza $O(10^{-4})$.

No **Caderno de Algoritmos**, apresentamos o algoritmo do método da falsa posição no arquivo **Cap3MetodosParticao.m**.

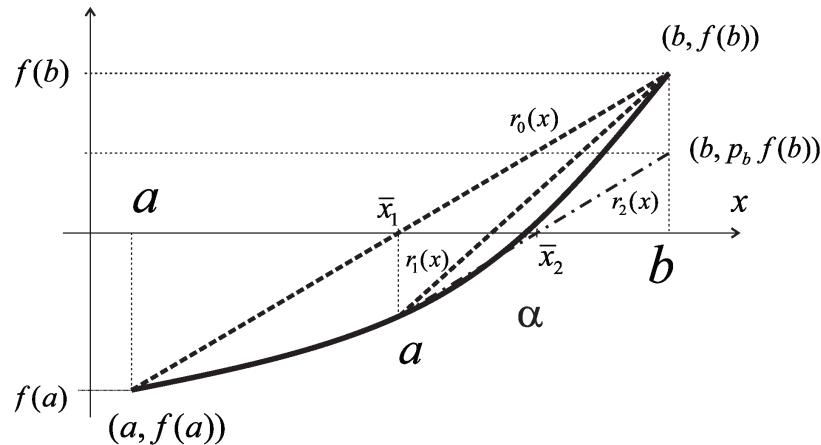
Na tabela de resultados do **Exemplo 3.4**, o ponto $b = 1.0$ ficou fixo ao longo das partições, pois $\alpha \in [\bar{x}_k, b]$, conforme esboçado no Gráfico 3.4. Esse ponto que ficou fixo, sem atualização ao longo das

partições, pode atrasar o processo de convergência. Para corrigir esse tipo de convergência unilateral, podemos destravar o ponto que ficou fixo usando o método da **falsa posição modificado**, por exemplo, ou adotar os métodos de linearização, que veremos na sequência.

3.2.1.3 Método da Falsa Posição Modificado

Para destravar algum ponto que fique fixo, seja a ou seja b , na aplicação do falsa posição, podemos adotar o método da falsa posição modificado, que impõe uma **redução na magnitude da função** aplicada ao ponto que esteja fixo, em cada partição k , conforme detalhado no Gráfico 3.5.

Gráfico 3.5 – Método da falsa posição modificado



Fonte: Elaboração própria

Assim, depois do cálculo inicial de $f(a)$, $f(b)$ e de $\bar{x}_1 = a - f(a)(b-a)/(f(b)-f(a))$, análogo ao método da falsa posição, precisamos:

- Identificar o ponto que fica fixo na partição, que é sempre aquele que continua sendo usado na redefinição do novo subintervalo; nesse caso, b fica fixo, pois $\alpha \in [\bar{x}_1, b]$, e o novo a é atualizado por \bar{x}_1 .
- Reducir o valor original da função correspondente ao ponto fixo: nesse caso $f(b)$, ao qual podemos aplicar um fator p_b

proporcional ao valor original da função no ponto não fixo, no caso a , dividido pela soma dos valores das funções neste ponto não fixo a e no ponto médio \bar{x}_1 :

$$p_b = f(a) / (f(a) + f(\bar{x}_1)) \quad (6)$$

Assim, reduzimos de $f(b)$ para $f(b) * p_b$ e usaremos esse novo valor no cálculo de \bar{x}_2 através da eq. (5).

Observe no gráfico 3.5 que a reta $r_1(x)$ corresponde à aplicação da 2ª partição do método da falsa posição usando os valores originais das funções $f(a)$ e $f(b)$, enquanto com o método da falsa posição modificado usamos a função reduzida $f(b) = f(b) * p_b$, que corresponde a nova reta $r_2(x)$, menos inclinada do que no método da falsa posição com $r_1(x)$, o que gera \bar{x}_2 mais próximo da raiz α , nesse exemplo.

Quando reduzimos o valor da função sobre o ponto fixo, aumentamos a velocidade de convergência, pois, a partir da partição seguinte, o novo valor de b é destravado e assume o \bar{x}_2 , conforme exemplo do Gráfico 3.5. Paralelamente, devemos atualizar o ponto do outro extremo do intervalo, no caso a é substituído pelo \bar{x}_k calculado e $f(a) = f(\bar{x}_k)$, mantendo sempre o teorema de Bolzano válido, conforme segue:

a) Se $f(\bar{x}_k) = 0 \Rightarrow \alpha = \bar{x}_k$

Senão, pelo teorema de Bolzano, decidiremos onde a raiz α ficou, se em $[a, \bar{x}_k]$ ou em $[\bar{x}_k, b]$:

i) quando a é o ponto fixo ($f(a)$ é reduzida):

$$\rightarrow \text{se } f(a) * f(\bar{x}_k) < 0, \quad b = \bar{x}_k, \quad f(b) = f(\bar{x}_k) \text{ e } f(a) = f(a) * p_a$$

ii) quando b é o ponto fixo ($f(b)$ é reduzida):

$$\rightarrow \text{se } f(\bar{x}_k) * f(b) < 0, \quad a = \bar{x}_k, \quad f(a) = f(\bar{x}_k) \text{ e } f(b) = f(b) * p_b$$

b) Depois das atualizações do valor extremo não fixo, do valor da sua função e do valor da função reduzida sobre o ponto

Se a é o ponto fixo, aplicamos o fator de redução p_a sobre $f(a)$, em que $p_a = f(b) / (f(b) + f(\bar{x}_k))$.



fixo, recalculamos o \bar{x}_k do intervalo como no método da falsa posição:

$$\bar{x}_k = a - \frac{f(a)*(b-a)}{(f(b)-f(a))}$$

- c) Com os novos $a, b, f(a), f(b)$ e \bar{x}_k , voltamos ao primeiro passo (a) até que algum critério de parada seja satisfeito.

Exemplo 3.5: obtenha a primeira raiz positiva α de $e^x \sin(x) - 1 = 0$, situada em $[0, 1]$, pelo método da falsa posição modificado, com 5 partições, e calcule o erro de truncamento atingido.

Solução:

Tabela 3.3 – Tabela de resultados do Exemplo 3.5

k	a	$\bar{x}_k = a - f(a)(b-a)/ (f(b)-f(a))$	b	$f(a)$	$f(\bar{x}_k)$	$f(b)$	$ \bar{x}_k - \bar{x}_{k-1} $
0	0	0.4371861	1.0	-1.0000	-0.344443	+1.28736	-
$p_b = f(a) / (f(a) + f(\bar{x}_k)) = 0.7438021478$							
1	0.4371861	0.5860805	1.0	-0.34444	-0.006117	+0.957537	0.1488943
$p_b = f(a) / (f(a) + f(\bar{x}_k)) = 0.9825494019$							
2	0.5860805	0.5887545	1.0	-0.00617	+0.00055	+0.940828	0.0026740
$p_a = f(b) / (f(b) + f(\bar{x}_k)) = 0.9994114861$							
3	0.5860805	0.5885323	0.58875447	-0.00614	-1.12e-06	+5.54e-04	2.222e-04
$p_b = f(a) / (f(a) + f(\bar{x}_k)) = 0.9998176448$							
4	0.5885323	0.588532744	0.58875447	-1.12e-6	-1.12e-06	+5.539e-4	4.464e-07
$p_a = f(b) / (f(b) + f(\bar{x}_k)) = 0.9999999013$							
5	0.5885323	0.5885327439818612	0.588532744	-1.115e-6	+2.22e-16	+5.46e-11	2.189e-11

Fonte: Elaboração própria

Observe que também aplicamos o teorema de Bolzano para redefinir cada novo intervalo $[a, b]$, conforme a troca de sinais das funções em a e b . Em $n = 5$ partições do intervalo inicial, atingimos $\bar{x}_5 = 0.5885327439818612$ com a função $|f(\bar{x}_k)| = +2.22e-16$ numericamente

nula, embora o critério de parada adotado $|\bar{x}_k - \bar{x}_{k-1}| < 2.189e-11$ ainda não seja nulo.

O erro de truncamento existente em \bar{x}_5 pode ser estimado como fizemos no método da bisseção, usando um valor de referência para comparação. Felizmente esse método é de convergência bem mais rápida do que os anteriores e atinge o limite do critério de parada para 16 dígitos significativos com apenas 6 partições, $\bar{x}_6 = 0.5885327439818612$. Assim, o erro de truncamento é numericamente nulo, conforme segue:

$$\text{Erro Truncamento estimado de } \bar{x}_5 = |\bar{x}_5 - \bar{x}_6| < O(10^{-16})$$

Também podemos dizer que o erro de truncamento é menor do que o critério de parada atingido, $|\bar{x}_k - \bar{x}_{k-1}| < 2.189e-11$ e observe que o valor da função já é numericamente nulo $|f(\bar{x}_5)| = 2.22044604925031e-16$.

Agora, confira no **Caderno de Algoritmos** o algoritmo do método da falsa posição modificado no arquivo **Cap3MetodosParticao.m**.

Comparando os três métodos de partição apresentados, obtivemos a primeira raiz positiva de $e^x \sin(x) - 1 = 0$, situada em $[0, 1]$, no limite do critério de parada para a variável *double* de 16 dígitos significativos com:

- a) 51 partições no método da bisseção;
- b) 23 partições no método da falsa posição; e
- c) 6 partições no método da falsa posição modificado.

Os três métodos convergiram para o mesmo valor de raiz α , à exceção de algum arredondamento final:

- a) 0.5885327439818611 (bissecção);
- b) 0.5885327439818610 (falsa posição);
- c) 0.5885327439818612 (falsa posição modificado).

Vamos abordar, a seguir, a família de métodos de linearização iterativa.

3.2.2 Métodos de Linearização

Nos métodos dessa família, buscamos linearizar a equação $f(x) = 0$, isolando um valor de x como **incógnita**, através de uma função do tipo $x = F(x)$, em que $F(x)$ é uma função de iteração avaliada em algum valor inicial de x .

Temos vários critérios de parada aplicáveis aos processos iterativos, alguns já vimos na seção anterior, como:

a) $|x_{k+1} - x_k| < \varepsilon$ – esse critério falha quando x_k converge lentamente;

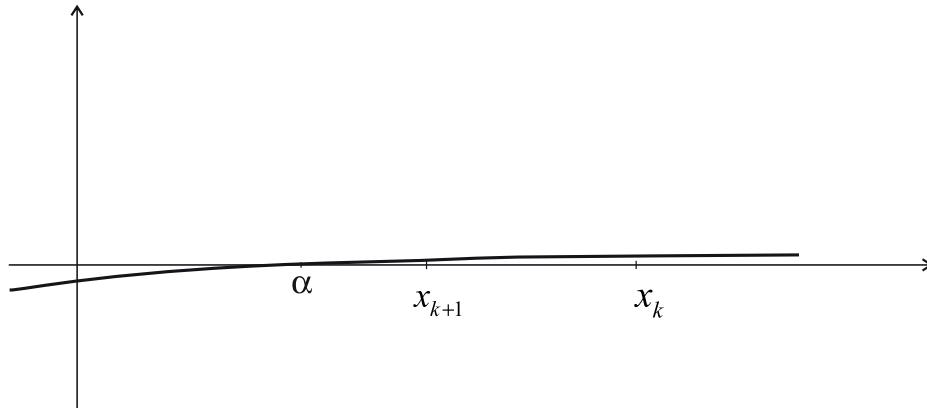
b) $\left| \frac{x_{k+1} - x_k}{x_{k+1}} \right| < \varepsilon$ – o valor relativo é mais realístico, principalmente quando x_{k+1} é muito pequeno (mas deve ser diferente de zero) ou muito grande;

c) $|f(x_{k+1})| < \varepsilon$ – esse critério falha quando $f(x)$ tem valores pequenos, conforme o Gráfico 3.6 ($f(x)$ calculado no novo x_{k+1});

d) $|f(x_{k+1})| + \left| \frac{x_{k+1} - x_k}{x_{k+1}} \right| < \varepsilon$ – é um critério composto, conservativo.

No Gráfico 3.6, observe que, apesar de x_k ainda estar longe da raiz α , o critério de parada $|f(x_{k+1})| < \varepsilon$ já pode ter sido satisfeito (quando $f(x)$ tem valores pequenos). Por outro lado, o critério $|x_k - x_{k-1}| < \varepsilon$ também pode ser satisfeito rapidamente quando x_k converge lentamente. Portanto, algum critério de parada sempre precisa ser estabelecido, mas não deve ser usado isoladamente, sendo necessário fazer uma validação do resultado aproximado obtido com algum cálculo do erro de truncamento.

Gráfico 3.6 – Processo iterativo de convergência, quando $f(x)$ tem valores pequenos



Fonte: Elaboração própria

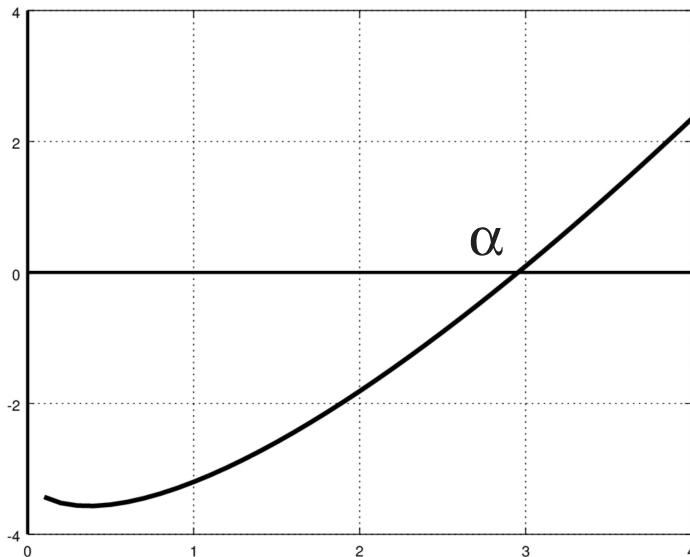
3.2.2.1 Método da Iteração Linear

O método clássico de linearização é a **iteração linear**. Embora seja normalmente lento e instável, é apresentado aqui por questões didáticas. Nesse método, obtemos a função de iteração $F(x)$ simplesmente isolando uma das variáveis x da própria função geradora $f(x)$ e aplicando inicialmente o valor estimado x_0 da etapa do isolamento da raiz. Assim,

$$\begin{aligned} x_1 &= F(x_0) \\ x_2 &= F(x_1) \\ x_3 &= F(x_2) \\ &\vdots \\ x_{k+1} &= F(x_k) \end{aligned} \tag{7}$$

em que os subíndices $k + 1$ e k indicam o nível iterativo novo e anterior, respectivamente.

Exemplo 3.6: determine, por iteração linear, a raiz de $x * \ln(x) - 3.2 = 0$, $\alpha \in [2, 3]$, com o máximo de dígitos exatos disponível em sua representação digital.

Solução:Gráfico 3.7 – Raiz de $f(x) = x * \ln(x) - 3.2 = 0$ 

Fonte: Elaboração própria

O Gráfico 3.7 mostra que a raiz está entre $a = 2$ e $b = 3$, logo podemos adotar qualquer valor inicial x_0 desse intervalo.

A seguir, vamos definir a função de iteração $F(x)$ isolando o x de duas formas possíveis:

- a) $x_0 * \ln(x) - 3.2 = 0$: isolamos como incógnita o x que aparece em $\ln(x)$, logo:

$$\ln(x) = 3.2/x_0 \Rightarrow x = e^{3.2/x_0}$$

Note que, nessa função iterativa, a solução x depende da estimativa da própria solução x_0 , portanto esperamos que esse cálculo melhore o valor de x em relação a x_0 . Assim, adotando o valor inicial médio em $[a, b]$, $x_0 = 2.5$, calculamos um novo $x = e^{3.2/2.5} = 3.59663972556928$.

Se agora adotarmos esse novo x como um novo valor inicial x_0 , então $x_0 = 3.59663972556928$, poderemos melhorar a raiz pela segunda vez, conforme a tabela iterativa a seguir:

Tabela 3.4 – Tabela de resultados do Exemplo 3.6

k	x_k	$x_{k+1} = e^{(3.2/x_k)}$	$ x_{k+1} - x_k $	$ f(x_{k+1}) $
0	2.5	3.59663972556928	1.09663972556928	1.40369884872868
1	3.59663972556928	2.43444635592001	1.16219336964927	1.03402594272603
2	2.43444635592001	3.72276714234931	1.28832078642930	1.69345547768941
3	3.72276714234931	2.36215810797272	1.36060903437659	1.16954640016981
4	2.36215810797272	3.87557252617237	1.51341441819964	2.05021252467948
:	:	:	:	:
999	1.34504630366580	10.79518554638936	9.45013924272355	22.48282865377773
1000	10.79518554638936	1.34504630366580	9.45013924272355	2.80129002385048
1001	1.34504630366580			

Fonte: Elaboração própria

Na Tabela 3.4, os critérios de parada $|x_{k+1} - x_k|$ e $|f(x_{k+1})|$ não estão diminuindo, e o valor x_{k+1} da raiz aproximada está oscilando, aumentando e diminuindo a cada iteração.

Observe que $F(x_k) = e^{3.2/x_0}$ é uma função iterativa do tipo exponencial que, neste exemplo, está amplificando o valor de x_k e consequentemente o seu erro. Então, precisamos reescrever a função iterativa $x = F(x)$.

- b) $x * \ln(x_0) - 3.2 = 0$: nessa tentativa, adotamos como incógnita o primeiro x da função, que aparece multiplicando $\ln(x)$, logo:

$$x = 3.2 / \ln x_0$$

Note que, nessa função iterativa, a solução x também depende da estimativa da solução x_0 , mas essa forma é uma função do tipo logaritmo que, neste exemplo, tende a reduzir o valor do x_0 e consequentemente o seu erro.

Adotando o mesmo $x_0 = 2.5$ como valor inicial, temos:

Tabela 3.5 – Tabela de resultados do Exemplo 3.6

k	x_k	$x_{k+1} = 3.2 / \ln(x_k)$	$ x_{k+1} - x_k $	$ f(x_{k+1}) $
0	2.5	3.492341337399333	0.992341337399333	1.167425623659166
1	3.492341337399333	2.558828299018562	0.933513038380771	0.795854258488439
2	2.558828299018562	3.405887761076910	0.847059462058348	0.973934620906707
3	3.405887761076910	2.611167118156381	0.794720642920529	0.693808868186438
4	2.611167118156381	3.334037325418966	0.722870207262585	0.814794334833144
5	3.334037325418966			
:	:	:	:	:
404	2.95416552327889	2.95416552327888	9.32587340685131e-15	8.88178419700125e-15
405	2.95416552327888	2.95416552327889	8.43769498715119e-15	8.88178419700125e-15
406	2.95416552327889			

Fonte: Elaboração própria

A raiz obtida em $n = 406$ iterações é $x_{406} = 2.95416552327889$, destacada em negrito, com critério de parada $|x_{k+1} - x_k| = 8.43769498715119e-15$.

No final das iterações, as aproximações de x_k não se alternando de uma iteração para outra e os critérios $|x_{k+1} - x_k|$, $|f(x_{k+1})|$ ficam estagnados.



Apresentamos os fatores de sub-relaxação no Capítulo 2.

Esse processo iterativo é altamente sensível à escolha da função de iteração $F(x)$, podendo gerar sequências convergentes, oscilatórias ou divergentes. Em alguns casos, podemos atenuar as oscilações usando fatores de amortecimento (**sub-relaxação**) no processo de atualização da incógnita x_{k+1} .

Para obter uma raiz com precisão máxima, é necessário que, a cada iteração, o valor calculado se aproxime da solução, ou seja, que o método converja para o valor exato da raiz.

No caso dos métodos de bisseção, falsa posição e falsa posição modificada, não precisamos nos preocupar com a convergência, que está sempre garantida, uma vez que isolamos a raiz dentro de

um dado intervalo e os métodos de quebra a mantém dentro de um subintervalo desse intervalo inicial. Já nos métodos iterativos de linearização, a convergência não é garantida. A cada iteração, podemos nos aproximar ou nos afastar da solução. Portanto, antes de resolver um problema por meio desse método, recomendamos tentar verificar se haverá ou não a convergência.

A seguir, vamos apresentar o teorema de convergência, que coloca condições suficientes, porém não necessárias, para que o método de iteração linear seja convergente.

3.2.2.1.1 Teorema de Convergência

Seja uma função $f(x)$ contínua em um intervalo $[a, b]$, α uma raiz de contida em $[a, b]$ e $F(x)$ uma função de iteração obtida a partir de $f(x) = 0$:

- a) se $F(x)$ e $F'(x)$ forem contínuas em $[a, b]$; e
- b) se $|F'(x_k)| < 1 \quad \forall x_k \in [a, b]$.

Então, $\lim_{k \rightarrow \infty} x_k = \alpha$, isto é, a sequência x_k converge para a solução α de $f(x) = 0$.

Porém, na maioria das vezes fica mais difícil obter e testar as $F'(x_k)$ do que aplicar diretamente o método e avaliar se os resultados estão convergindo ou não. No **Exemplo 3.6 (a)**, é fácil perceber que x_k aumenta e diminui alternadamente e, portanto, o método não converge. No **Exemplo 3.6 (b)**, também percebemos que x_k oscila, aumenta e diminui alternadamente, mas acaba convergindo. Para minimizar essas oscilações, podemos aplicar o fator de sub-relaxação, conforme o **Exemplo 3.7**.

Exemplo 3.7: determine a raiz de $x * \ln(x) - 3.2 = 0$, $\alpha \in [2, 3]$ com o máximo de dígitos significativos disponível em sua representação digital, usando fator de sub-relaxação para atenuar as oscilações nas iterações.

Solução:

A forma iterativa que adotamos no **Exemplo 3.6 (b)**, $x = 3.2/\ln(x_0)$, foi oscilatória mas convergente. Então, recomendamos usar um fator de amortecimento ou sub-relaxação $0 < \lambda < 1$, como adotado no método de Gauss-Seidel aplicado no Capítulo 2 para determinação iterativa de soluções x oscilatórias de sistemas de equações lineares:

$$x = x_0 + \lambda * \Delta x \quad (8a)$$

Em que $\Delta x = x - x_0$ é o incremento iterativo original do método usado, pois esse x é o valor novo dado por $x = 3.2/\ln(x_0)$.

Reescrevendo a eq. (8a), temos:

$$x = x_0 + \lambda * \Delta x = x_0 + \lambda(x - x_0) = (1 - \lambda)x_0 + \lambda * x \quad (8b)$$

Note que, se $\lambda = 1$, teremos o cálculo original.

Adotando $x_0 = 2.5$ e $\lambda = 0.5$, teremos a função iterativa:

$$x = (1 - \lambda)x_0 + \lambda(3.2 / \ln(x_0))$$

Tabela 3.6 – Tabela de resultados do Exemplo 3.7

k	x_k	$x_{k+1} = (1 - \lambda)x_k + \lambda(3.2 / x_k)$	$ x_{k+1} - x_k $	$ f(x_{k+1}) $
0	2.5	2.996170668699666	0.496170668699666	0.0878030292836272
1	2.9961706686996665	2.956163274914294	0.0400073937853724	0.0041624239393347
2	2.9561632749142941	2.95424314678781	1.920128126488e-03	1.617075723499e-04
3	2.95424314678781	2.95416850626476	7.464052305028e-05	6.214206048849e-06
4	2.95416850626476	2.95416563786223	2.868402523503e-06	2.387018933625e-07
:	:	:	:	:
10	2.95416552327889	2.954165523278883	9.325873406851e-15	4.440892098501e-16
11	2.954165523278883	2.954165523278883	0.000000000000000	0.000000000000000
12	2.954165523278883			

Fonte: Elaboração própria

Assim, obtivemos a raiz em $n = 12$ iterações, $x_{12} = 2.95165523278883$, em negrito, com critérios de parada numericamente nulos $|x_{k+1} - x_k| = 0.0$ e $|f(x_{k+1})| = 0$, portanto erro de truncamento no limite da precisão da variável *double*.

Podemos testar outros valores de λ e verificar que $\lambda = 0.5$ é mesmo o valor ótimo, permitindo a convergência no menor número de iterações. Lembre-se de que, sem o fator de amortecimento $\lambda = 0.5$, conforme o **Exemplo 3.6 (b)**, foram necessárias 406 iterações para atingir a mesma solução com 16 dígitos significativos.

Acesse o *link* <<http://sergiopeters.prof.ufsc.br/algoritmos-livro/>> para baixar o **Caderno de Algoritmos** e conferir o algoritmo do método da iteração linear no arquivo **Cap3MetodosIterativos.m**.

Também podemos implementar algoritmos puramente iterativos diretamente em **calculadoras científicas**, desde que tenham o recurso Ans (Answer) para armazenar o valor inicial, desta forma:

- digitamos o valor inicial x_0 , por exemplo 2.5, e pressionamos a tecla "=" para armazená-lo em Ans;
- digitamos a equação iterativa do novo x , com x_0 representado pelo Ans (valor da raiz anterior):

$3.2 \div \ln(\text{Ans})$, no caso do **Exemplo 3.6 (b)**, ou

$(1 - 0.5) * \text{Ans} + 0.5 * 3.2 \div \ln(\text{Ans})$, no caso do **Exemplo 3.7**; e

- pressionando a tecla "=" será gerado um novo x , que fica automaticamente armazenado em Ans e atualiza x_0 . Desse modo, a cada "=" pressionado, teremos uma iteração gerada no *display* da calculadora.

Vamos apresentar agora o método de Newton, que é mais rápido e estável do que o método da iteração linear.

3.2.2.2 Método de Newton

O método de linearização de Newton é o mais utilizado na solução de equações. Para se obter a sua função de linearização, a $f(x)$ pode ser aproximada de duas formas.

Para $f(x) = 0$ com as funções $f(x)$ e $f'(x)$ contínuas em um intervalo suficientemente próximo da solução desejada, podemos gerar uma sequência de aproximações dessa solução da seguinte forma:

- a) Tomamos uma solução inicial x_0 .
- b) Calculamos o ponto $(x_0, f(x_0))$ e o coeficiente angular $f'(x_0)$ da reta $r(x)$ tangente à $f(x)$ nesse ponto x_0 , conforme o Gráfico 3.8.

Então, essa reta será definida por:

$$r(x) = f(x_0) + f'(x_0)(x - x_0) \quad (9)$$

- c) Calculamos a interseção de $r(x)$ com o eixo das abscissas x :

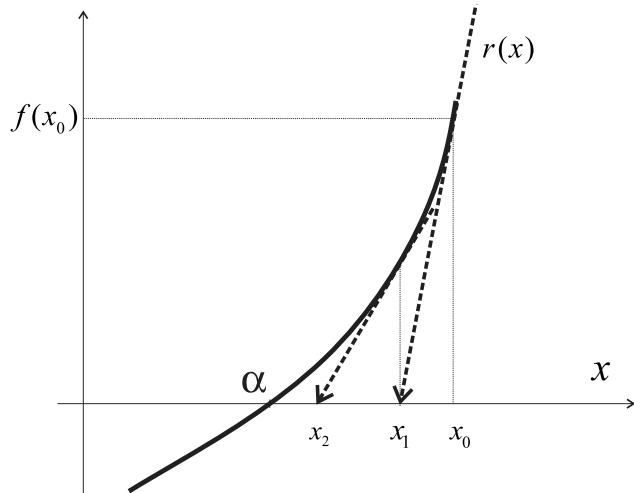
$\rightarrow r(x) = 0$, eq. (9), obtemos a função de iteração $x = F(x)$:

$$x - x_0 = -f(x_0)/f'(x_0) \quad \text{para o incremento de } x; \text{ e}$$

$$x = x_0 - f(x_0)/f'(x_0) \quad \text{para } x \quad (F(x) = x - f(x)/f'(x)).$$

Repetindo sucessivamente o passo (b), sempre usando o último valor calculado como solução anterior x_k , obtemos a expressão de recorrência:

$$x_{k+1} = x_k - f(x_k)/f'(x_k) \quad (10)$$

Gráfico 3.8 – Interpretação geométrica do método de Newton e sua reta tangente $r(x)$ 

Fonte: Elaboração própria

Note que a eq. (10) gera uma sequência de valores x_k , $k = 0, 1, 2, \dots$, conforme o Gráfico 3.8, que, se for convergente, existirá um valor final α qualquer para $\lim_{k \rightarrow \infty} x_{k+1} = \lim_{k \rightarrow \infty} x_k = \alpha$. Por consequência, aplicando esses limites na eq. (10), temos:

$$\lim_{k \rightarrow \infty} x_{k+1} = \lim_{k \rightarrow \infty} x_k - \lim_{k \rightarrow \infty} \frac{f(x_k)}{f'(x_k)} \rightarrow \lim_{k \rightarrow \infty} x_{k+1} = \lim_{k \rightarrow \infty} x_k - \frac{\lim_{k \rightarrow \infty} (f(x_k))}{\lim_{k \rightarrow \infty} (f'(x_k))}$$

Devido à continuidade de $f(x)$ e $f'(x)$, temos:

$$\lim_{k \rightarrow \infty} f(x_k) = f(\lim_{k \rightarrow \infty} x_k) \text{ e } \lim_{k \rightarrow \infty} f'(x_k) = f'(\lim_{k \rightarrow \infty} x_k),$$

então

$$\lim_{k \rightarrow \infty} x_{k+1} = \lim_{k \rightarrow \infty} x_k - \frac{f(\lim_{k \rightarrow \infty} x_k)}{f'(\lim_{k \rightarrow \infty} x_k)}$$

Como $\lim_{k \rightarrow \infty} x_{k+1} = \lim_{k \rightarrow \infty} x_k = \alpha$, temos que:

$$\alpha = \alpha - \frac{f(\alpha)}{f'(\alpha)} \rightarrow \frac{f(\alpha)}{f'(\alpha)} = 0 \text{ e } f(\alpha) = 0, \text{ desde que } f'(\alpha) \neq 0.$$

Logo, α é uma solução pois $f(x = \alpha) = 0$.

Portanto, se a sequência gerada pela eq. (10) for convergente, o seu limite será sempre uma solução. Mais ainda, sob condições

normais, a velocidade de convergência ou divergência desse método é do tipo quadrática, isto é, o incremento $\Delta x = x_{k+1} - x_k$ da aproximação de x_{k+1} é aproximadamente o quadrado do incremento da aproximação anterior, fato que não ocorre nos métodos de quebra.

Alternativamente, podemos obter o método de Newton como uma linearização da função $f(x)$ através de uma série de Taylor truncada:

- a) seja $f(x)$ uma função contínua no intervalo $[a,b]$;
- b) seja α um zero contido nesse intervalo, com função $f(x)$ e derivada $f'(x)$ contínuas em $[a,b]$, mas com $f'(\alpha) \neq 0$.

Desse modo, podemos encontrar uma aproximação x_{k+1} para a raiz de $f(x) = 0$ no intervalo $[a,b]$ utilizando a expansão de $f(x_{k+1})$ em série de Taylor em torno de um valor inicial estimado x_k , em que $\Delta x = x_{k+1} - x_k$.

Logo, $x_{k+1} = x_k + \Delta x$ e

$$f(x_{k+1}) = f(x_k + \Delta x) = f(x_k) + f'(x_k) \underbrace{\frac{\Delta x^1}{1!} + f''(x_k) \frac{\Delta x^2}{2!} + \dots}_{\text{função original, representada em série de Taylor}}$$

Fazendo $f(x_{k+1}) = 0$, temos

$$f(x_k) + f'(x_k) \frac{\Delta x^1}{1!} + f''(x_k) \frac{\Delta x^2}{2!} + \dots = 0$$

Para determinar a incógnita Δx da série de Taylor, Newton assumiu que:

- a) x_k está em um intervalo suficientemente próximo da solução;
- b) Δx é suficientemente pequeno; e a função $f(x)$ pode ser representada apenas pelos seus dois primeiros termos da série de Taylor, truncando os termos de ordem superior a 2, portanto resulta uma aproximação de 1ª ordem equivalente à reta $r(x)$ definida pela eq. (9),

$$f(x_k + \Delta x) \cong f(x_k) + f'(x_k) \Delta x = 0$$

Observe que a equação não linear original, $f(x) = 0$, foi substituída por uma equação linearizada, aproximada, cuja incógnita x está presente em $\Delta x = x - x_k$.

Então, calculamos o novo x_{k+1} aproximado por

$$\Delta x = -\frac{f(x_k)}{f'(x_k)} \quad (11a)$$

e

$$x_{k+1} = x_k + \Delta x \quad (11b)$$

Note que o valor x_{k+1} da fórmula de Newton é uma aproximação de 1ª ordem para a raiz de $f(x) = 0$, não é o seu valor exato, pois houve um erro de truncamento da série de Taylor representativa de $f(x)$ da ordem de $O(\Delta x^2)$ dado por:

$$O(\Delta x^2) = f''(x_k) \frac{\Delta x^2}{2!} + f'''(x_k) \frac{\Delta x^3}{3!} + \dots$$

Assim, em vez de resolver a $f(x) = 0$ completa, com infinitas parcelas na série, resolvemos apenas uma aproximação de $f(x) = 0$, com apenas duas parcelas:

$$\underbrace{f(x_k) + f'(x_k) \frac{\Delta x^1}{1!} + f''(x_k) \frac{\Delta x^2}{2!} + \dots = 0}_{\text{eq. original, representada em série de Taylor com infinitas parcelas}} \Rightarrow \underbrace{f(x_k) + f'(x_k) \frac{\Delta x^1}{1!} = 0}_{\text{eq. simplificada, representada com 2 parcelas}}$$

No **Exercício 3.3** deste Capítulo, propomos uma avaliação comparativa do método de Newton de 1ª ordem apresentado anteriormente, com uma extensão para um método de Newton de 2ª ordem (com os 3 primeiros termos da série de Taylor). Nesse exercício, poderemos avaliar que o número de operações aritméticas totais é menor no método de Newton tradicional de 1ª ordem.

No **Caderno de Exercícios e Respostas**, disponível no *link* <<http://sergiopeters.prof.ufsc.br/exercicios-e-respostas/>>, você encontrará os exercícios atualizados deste livro.

Agora, vamos apresentar um exemplo de aplicação do método de Newton e fazer sua comparação com o método da iteração linear.

Exemplo 3.8: determine a raiz de $x * \ln(x) - 3.2 = 0$ com $\alpha \in [2, 3]$ por Newton e compare sua eficiência em relação ao método de iteração linear.

Solução:

Dado x_0 , podemos determinar Δx e um novo x , conforme a eq. (10) ou as eqs. (11a) e (11b):

$$\begin{aligned}\Delta x &= -\frac{f(x_0)}{f'(x_0)} \\ x &= x_0 + \Delta x \\ f(x_0) &= x_0 * \ln(x_0) - 3.2 \\ f'(x_0) &= \ln(x_0) + 1\end{aligned}$$

Adotando valor inicial $x_0 = 2.5$, como no **Exemplo 3.6**, temos:

Tabela 3.7 – Tabela de resultados do Exemplo 3.8

k	x_k	$f(x_k) = x_k * \ln(x_k) - 3.2$	$f'(x_k) = \ln(x_k) + 1$	$\Delta x = -\frac{f(x_k)}{f'(x_k)}$	$x_{k+1} = x_k + \Delta x$	$ \Delta x $
0	2.500000000000000	-9.09273170314612e-01	1.91629073187416	4.74496460892097e-01	2.97449646089210	4.74e-01
1	2.97449646089210	4.24235390152670e-02	2.09007476782904	-2.02976178978191e-02	2.95419884299428	2.03e-02
2	2.95419884299428	6.94123593323326e-05	2.08322749497656	-3.33196251968217e-05	2.95416552336908	3.33e-05
3	2.95416552336908	1.87902582382549e-10	2.08321621617818	-9.01983101529758e-11	2.95416552327888	9.02e-11
4	2.95416552327888	0.00000000000000e+00	2.08321621614765	0.00000000000000e+00	2.95416552327888	0.00e00
5	2.95416552327888					

Fonte: Elaboração própria

A raiz obtida é **2.95416552327888**, destacada em negrito, com critério de parada numericamente nulo $|\Delta x| = |x_{k+1} - x_k| \approx 0$, em $n = 5$ iterações, enquanto no método de iteração linear foram necessárias $n = 406$ iterações (**Exemplo 3.6 (b)**), sem fator de sub-relaxação. Então, o erro de truncamento também está no limite da precisão da variável *double*.

Também podemos implementar esse algoritmo de Newton em **calculadora científica** com recurso Ans (Answer) desta forma:

- digitamos o valor inicial, por exemplo 2.5, e pressionamos a tecla "=" para armazená-lo em Ans;
- digitamos a equação iterativa com x_0 representado pelo Ans (valor da resposta anterior), no caso:

$$x = x_0 - \frac{f(x_0)}{f'(x_0)}$$

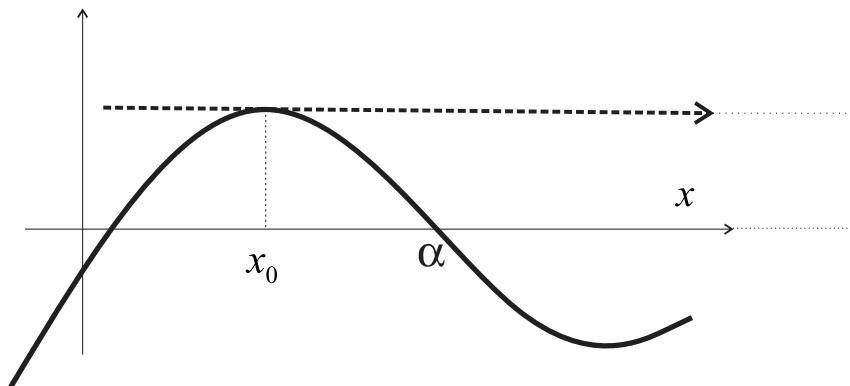
$\text{Ans} - (\text{Ans} * \ln(\text{Ans}) - 3.2) / (\ln(\text{Ans}) + 1)$, no caso do **Exemplo 3.8**; e

c) pressionando a tecla "=", sucessivamente, o valor gerado será um novo x , que já fica armazenado em Ans e atualiza x_0 . Desse modo, a cada "=" pressionado, teremos uma iteração gerada no *display* da calculadora.

Algumas considerações sobre o método de Newton:

- Note que o método de Newton tem convergência rápida, e pudemos comprovar que, sob condições normais, a convergência ou divergência desse método é do tipo quadrática, isto é, o incremento $|\Delta x|$ da iteração $(k+1)$ é aproximadamente o quadrado do incremento da iteração anterior (k) , conforme vemos no quadro de resultados do **Exemplo 3.8**, onde o expoente de Δx dobra a cada iteração.
- O novo valor para x_{k+1} é o ponto de corte da reta tangente $r(x)$ com o eixo das abscissas, pois x_{k+1} é o zero dessa reta $r(x)$ approximadora de $f(x)$.
- Pode ocorrer, durante a determinação dos x_{k+1} , que a reta tangente à $f(x)$ em $(x_k, f(x_k))$ seja paralela ao eixo da abscissa x , ou seja, $f'(x_k) = 0$, como no Gráfico 3.9 em $k=0$.

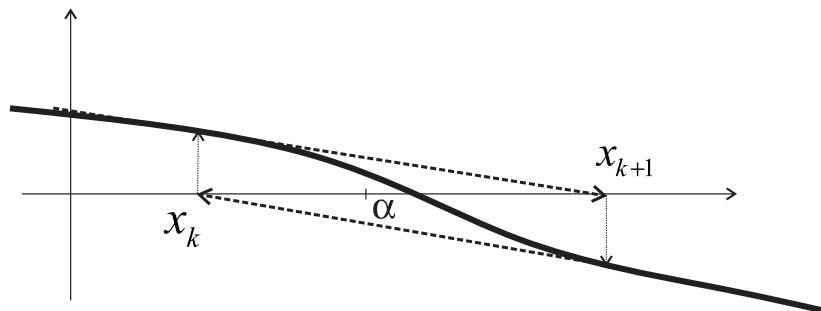
Gráfico 3.9 – Método de Newton quando o valor inicial da raiz gera derivada nula



Fonte: Elaboração própria

- d) Também pode ocorrer um ciclo repetitivo, ou *looping*, dos valores na sequência gerada pelos x_k e consequentemente o processo não converge nem diverge, conforme o Gráfico 3.10.

Figura 3.10 – Exemplo de *looping* dos valores na sequência gerada pelos x_k



Fonte: Elaboração própria

Para evitar a divisão por zero e os *loopings*, temos algumas possíveis soluções:

- i) escolher x_0 mais próximo possível de α ;
- ii) testar o uso de fatores de relaxação, conforme o **Exemplo 3.7** do método da iteração linear; e
- iii) tentar utilizar uma variante do método de Newton fixando o denominador em algum x_0 , com $f'(x_0) \neq 0$, desde que seja suficientemente próximo da raiz.

O algoritmo do método de Newton está disponível no **Caderno de Algoritmos** no arquivo **Cap3MetodosIterativos.m**.

3.2.2.3 Método da Secante

Como o método de Newton necessita da $f'(x)$ e pode ser muito difícil ou até inviável obter a sua expressão, podemos utilizar aproximações para os valores de $f'(x)$.

Por definição, temos que:

$$f'(x_0) = \lim_{\Delta x \rightarrow 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}$$

Então, se Δx for suficientemente pequeno e não nulo, poderemos usar a aproximação $f'(x_0) \approx \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}$, denominada de derivada numérica.

Assim, a derivada numérica depende de dois pontos, x_0 e $x_1 = x_0 + \Delta x$, e podemos calculá-la por:

$$f'(x_0) \approx \frac{f(x_1) - f(x_0)}{(x_1 - x_0)} \quad (12)$$

ou

$$f'(x_0) \approx \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} \quad (13)$$

Logo, aplicamos a derivada numérica dada pelas eqs. (12) ou (13) na eq. (10) e obtemos:

$$x_2 = x_0 - f(x_0) \left/ \left(\frac{f(x_1) - f(x_0)}{(x_1 - x_0)} \right) \right. \quad (14)$$

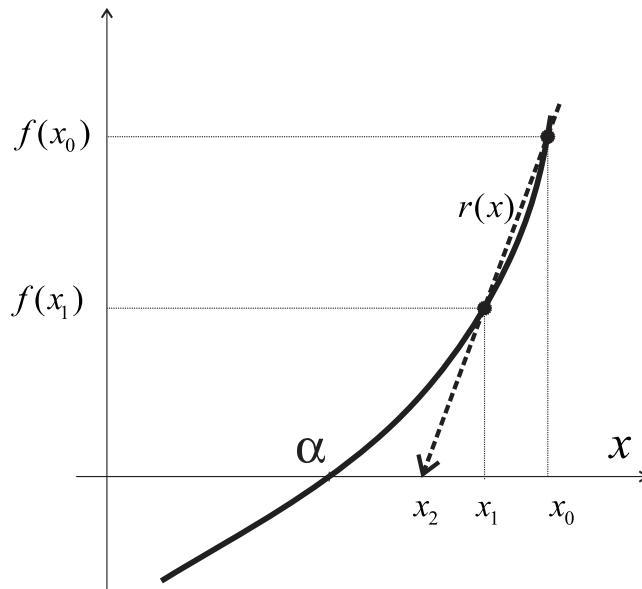
Calculado o novo valor x_2 , atualizamos os valores mais antigos x_0 e x_1 pelos valores mais atualizados disponíveis $x_0 = x_1$ e $x_1 = x_2$ e voltamos ao cálculo da derivada.

Observamos que o valor x_0 mais antigo de todos é descartado e os valores mais atualizados x_1 e x_2 são mantidos.

Se usarmos o Δx como incremento para cálculo da derivada, este assumirá naturalmente o último valor Δx calculado no método de Newton.

Repare que o x_2 é a interseção entre a reta secante, gerada pelos dois pontos anteriores $(x_0, f(x_0))$ e $(x_1, f(x_1))$, com o eixo das abcissas x , conforme o Gráfico 3.11.

Gráfico 3.11 – Método da secante



Fonte: Elaboração própria

Para algoritmização desse método, podemos:

- a) utilizar x_0 e Δx , ambos estimados inicialmente;
- b) aproximar a derivada por meio da eq. (13)

$$f'(x_0) \approx \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} ;$$

- c) calcular um novo incremento Δx e solução x de cada iteração pelo método de Newton original, eqs. (11a) e (11b);
- d) atualizar $x_0 = x$ e o incremento Δx (usando o próprio incremento do método de Newton); e
- e) voltar ao passo (b), da mesma forma que no método de Newton original.

Alternativamente, podemos aplicar diretamente a eq. (14) da seguinte forma:

- estimar x_0 e x_1 iniciais;
- calcular o novo valor x_2 , pela eq. (14);
- calcular $f(x_2)$;
- atualizar os valores mais antigos:

$$x_0 = x_1 \text{ e } f(x_0) = f(x_1) \text{ bem como } x_0 = x_2 \text{ e } f(x_0) = f(x_2);$$

- voltar ao passo (b).

Observamos que, neste último algoritmo, a função $f(x)$ é calculada apenas uma vez, no passo (c), e apenas atualizada no passo (d), portanto é uma forma mais otimizada.

A fórmula do método da secante, eq. (14), é semelhante à do método da falsa posição, eq. (5), mas aqui não é necessário um intervalo inicial que contenha a solução.

Exemplo 3.9: determine a raiz de $x * \ln(x) - 3.2 = 0$ com $\alpha \in [2, 3]$ pelo método da secante.

Solução:

Adotando $x_0 = 2.5$ e $\Delta x = 2.5 * 10^{-6}$ como valores iniciais e aplicando o método da secante, temos:

Tabela 3.8 – Tabela de resultados do Exemplo 3.9

k	x_k	$f(x_k) = x_k * \ln(x_k) - 3.2$	$f(x_k + \Delta x)$	$f'(x_k)$	$\Delta x = -\frac{f(x_k)}{f'(x_k)}$	$x_{k+1} = x_k + \Delta x$
0	2.50000000000000	-9.09273170314612e-01	-9.09268379586533e-01	1.91629123165171	4.74496337141240e-01	2.97449633714124
1	2.97449633714124	4.24232803667257e-02	1.07013647224394e+00	2.16590331986335	-1.95868762828261e-02	2.95490946085841
2	2.95490946085841	1.54987649326843e-03	-3.91936931036736e-02	2.08014636987656	-7.45080498042262e-04	2.95416438036037
3	2.95416438036037	-2.38094615578888e-06	-1.55445046612801e-03	2.08308971185042	1.14298781384402e-06	2.95416552334818
4	2.95416552334818	1.44371181676206e-10	2.38123534090917e-06	2.08321640956046	-6.93020566723873e-11	2.95416552327888
5	2.95416552327888	0.00000000000000e+00	-1.44370293497786e-10	2.08320359351340	0.00000000000000e+00	2.95416552327888
6	2.95416552327888					

Fonte: Elaboração própria

A raiz obtida é **2.95416552327888**, destacada em negrito, com critérios de parada numericamente nulos para $|\Delta x|$ e $|f(x_k)|$, em 6 iterações, enquanto o método de Newton obteve a mesma solução em 5 iterações, portanto o método da secante, no

Exemplo 3.9, exige pouco esforço computacional adicional em relação ao método de Newton e dispensa a derivada analítica. Então, neste exemplo, o erro de truncamento também chegou no limite da precisão da variável *double*, mas devemos limitar o valor de Δx a um valor mínimo, para que a diferença entre $f(x_0 + \Delta x)$ e $f(x_0)$ não seja numericamente nula, ou $f'(x_k)$ não seja nula.

Disponibilizamos as duas formas do algoritmo do método da secante no **Caderno de Algoritmos** no arquivo **Cap3MetodosIterativos.m** como método de Newton com derivadas numéricas e como método da secante.

A seguir, vamos apresentar uma adaptação do método de Newton para os casos em que a solução α de $f(x) = 0$ seja um ponto crítico.

3.2.2.4 Tratamento Geral de Funções com Derivada Nula em $x = \alpha$

Para equações de raízes simples (não repetidas), com $f'(x)$ nulo em algum dos seus valores x_k , podemos tentar outro(s) valor(es) inicial(ais) ou fixar um valor aproximado da derivada, entre outras possibilidades.

Para equações de raízes múltiplas (repetidas), a fórmula do método de Newton apresentará, além de $\lim_{x \rightarrow \alpha} f(x) \rightarrow 0$, também $\lim_{x \rightarrow \alpha} f'(x) \rightarrow 0$, o que provoca convergência lenta, isto é, perde a convergência quadrática, e gera uma indeterminação 0/0 para Δx no limite de convergência, conforme veremos na próxima seção.

Vamos primeiramente avaliar os efeitos da convergência com derivada tendendo a zero em uma função não polinomial e algumas das possíveis metodologias de soluções apresentadas na literatura pertinente.

Exemplo 3.10: obtenha uma raiz de $e^x - x - 1 = 0$ com $x_0 = 1.0$ por meio do método de Newton.

Solução:

$$f(x) = e^x - x - 1$$

$$f'(x) = e^x - 1$$

Aplicando a fórmula do método de Newton, conforme as eqs. (11a) e (11b), com $x_0 = 1$, e destacando que a raiz exata dessa equação é 0.0, temos:

Tabela 3.9 – Tabela de resultados do Exemplo 3.10

k	x_k	$f(x_k)$	$f'(x_k)$	$\Delta x = -\frac{f(x_k)}{f'(x_k)}$	$x_{k+1} = x_k + \Delta x$	$ f(x) + \Delta x $
0	1.000000e+00	7.182818e-01	1.718282e+00	-4.180233e-01	5.819767e-01	6.256190e-01
1	5.819767e-01	2.075957e-01	7.895724e-01	-2.629217e-01	3.190550e-01	3.196937e-01
2	3.190550e-01	5.677201e-02	3.758270e-01	-1.510589e-01	1.679962e-01	1.659948e-01
3	1.679962e-01	1.493591e-02	1.829321e-01	-8.164730e-02	8.634887e-02	8.548502e-02
4	8.634887e-02	3.837726e-03	9.018660e-02	-4.255317e-02	4.379570e-02	4.352636e-02
:						
22	3.389852e-07	5.750955e-14	3.389853e-07	-1.696521e-07	1.693332e-07	1.696521e-07
23	1.693332e-07	1.443290e-14	1.693332e-07	-8.523373e-08	8.409945e-08	8.523373e-08
24	8.409945e-08	3.552714e-15	8.409945e-08	-4.224420e-08	4.185525e-08	4.224420e-08
25	4.185525e-08	0.000000e+0	4.185525e-08	0.000000e+00	4.185525e-08	0.000000e+00
26	4.185525e-08					

Fonte: Elaboração própria

Em 26 iterações, geramos $x_{26} = 4.185525e-08$, destacada em negrito, com critérios de parada numericamente nulos para $|\Delta x|$ e $|f(x_k)|$. Como o Δx se tornou nulo, a partir de $x_{26} = \underline{0.00000004185525}$, os demais valores de x_k ficam constantes e com apenas 8 dígitos exatos (sublinhados).

Observe, na planilha de resultados do **Exemplo 3.10**, que:

- a) Os critérios de parada foram reduzidos a zero, mas a raiz convergida ainda não é a exata, pois deveria ser zero em todos os seus dígitos significativos. Com variável *double*, seriam 16 dígitos exatos, mas a raiz aproximada x_{26} atingiu apenas 8 dígitos exatos.

b) Ocorreu a perda da convergência quadrática, uma vez que o Δx novo foi reduzido apenas pela metade do anterior a cada iteração. A convergência quadrática, normal no método de Newton, reduz o Δx novo para o quadrado do Δx da iteração anterior.

Para resolver esse problema, Cheney e Kincaid (2008) apresentam duas possíveis soluções:

a) Uma delas usa uma correção na fórmula de cálculo do incremento das raízes pela multiplicidade M da raiz α da seguinte forma:

$$\Delta x = -\frac{f(x_k)}{f'(x_k)} \quad (15a)$$

$$x_{k+1} = x_k + M \Delta x \quad (15b)$$



Veremos como calcular a multiplicidade estimada

na seção 3.3, sobre raízes de equações polinomiais.

essa correção apresenta melhores resultados para casos como o do **Exemplo 3.10** (resultados não apresentados neste livro), mas é **necessário conhecer previamente** essa **multiplicidade** M . Nesse exemplo, como $f(x) \rightarrow 0$ e $f'(x) \rightarrow 0$, no limite da precisão digital das variáveis utilizadas (no caso *double*), podemos adotar multiplicidade equivalente $M = 2$, conforme **Propriedade 13** de polinômios, que veremos mais adiante.

b) Outra possibilidade utiliza uma correção na fórmula de cálculo do incremento da raiz quando $f(x) \rightarrow 0$ e $f'(x) \rightarrow 0$, determinando alternativamente a raiz de $g(x) = f(x)/f'(x) = 0$, pois toda raiz α que anula $f(x) = 0$ também anula $g(x) = 0$, então, aplicamos o método de Newton à função modificada $g(x) = 0$:

$$\Delta x = -\frac{g(x_k)}{g'(x_k)} = -\frac{\frac{f(x_k)}{f'(x_k)}}{\frac{[f'(x_k)]^2 - f(x_k)f''(x_k)}{[f'(x_k)]^2}} = -\frac{f(x_k)f'(x_k)}{[f'(x_k)]^2 - f(x_k)f''(x_k)} \quad (16)$$

ou

$$\Delta x = -\hat{M} \frac{f(x_k)}{f'(x_k)} \quad (17)$$

em que

$$\hat{M} = \frac{1}{g'(x_k)} = \left\{ \frac{[f'(x_k)]^2}{[f'(x_k)]^2 - f(x_k)f''(x_k)} \right\} \quad (18)$$

Com $x_{k+1} = x_k + \Delta x$, dado pela eq. (11b).

Exemplo 3.11: reaplique o método de Newton, modificando

$f(x) = 0$ para $g(x) = \frac{f(x)}{f'(x)} = 0$, conforme as eqs. (17) e (18), e

obtenha uma raiz de: $e^x - x - 1 = 0$ com $x_0 = 1.0$.

Solução:

$$f(x) = e^x - x - 1$$

$$f'(x) = e^x - 1$$

$$f''(x) = e^x$$

Tabela 3.10 – Tabela de resultados do Exemplo 3.11

k	x_k	$f(x_k)$	$f'(x_k)$	$f''(x_k)$	\hat{M} eq. (18)	$\Delta x = -\hat{M} \frac{f(x_k)}{f'(x_k)}$	$x_{k+1} = x_k + \Delta x$	$ f(x) + \Delta x $
0	1.000000e+00	7.182818e-01	1.718282e+00	2.718282e+00	2.952492e+00	-1.234211e+00	-2.342106e-01	1.259616e+00
1	-2.342106e-01	2.540578e-02	-2.088048e-01	7.911952e-01	1.855412e+00	2.257523e-01	-8.458280e-03	2.257880e-01
2	-8.458280e-03	3.567061e-05	-8.422609e-03	9.915774e-01	1.994377e+00	8.446390e-03	-1.189018e-05	8.446390e-03
3	-1.189018e-05	7.068790e-11	-1.189011e-05	9.999881e-01	1.999991e+00	1.189014e-05	-4.176666e-11	1.189014e-05
4	-4.176666e-11	0.000000e+00	-4.176670e-11	1.000000e+00	1.000000e+00	0.000000e+00	-4.176666e-11	0.000000e+00
5	-4.176666e-11							

Fonte: Elaboração própria

Observe que:

- a) em $k = 3$, o valor de \hat{M} tende a 2, com $f(x \rightarrow \alpha) \rightarrow 0$ e $f'(x \rightarrow \alpha) \rightarrow 0$, valores destacados em negrito; e
- b) em $k = 4$, $f(x_k) = 0$, logo \hat{M} se torna unitário ($\hat{M} = 1$ sempre que $f(x) = 0$), $\Delta x = 0$ e x_5 não vai mais se alterar.

Assim, $x_5 = -\underline{0.000000000417667}$, também em negrito, é uma raiz convergida com 11 dígitos exatos (sublinhados), e apesar de ser melhor do que a solução obtida no **Exemplo 3.10** (com 8 dígitos exatos), os critérios de parada também zeraram antes de chegarmos a raiz com precisão de 16 dígitos da variável *double* utilizada.

Segundo Schroder (1992), o termo $1/g'(x_k)$ dado pela eq. (18) representa uma estimativa clássica do valor da multiplicidade \hat{M} de raízes, mas essa estimativa é válida para raízes estimadas próximas da raiz exata α .

No **Exemplo 3.11**, a função e sua derivada tendem a zero, $f(x) \rightarrow 0$ e $f'(x) \rightarrow 0$, fazendo \hat{M} convergir para 2 (antes de $f(x)$ se tornar nula), conforme **Propriedade 13** de polinômios, que veremos mais adiante.

Essa correção, dada pelas eqs. (17) e (18), também apresentou melhores resultados, mas não atinge a precisão no limite da variável *double*.

Observamos que as eqs. (15) e (17) serão equivalentes à medida que a eq. (18) da multiplicidade estimada \hat{M} se torne a multiplicidade M exata, ao longo de um processo iterativo.

Em ambos os casos anteriores, usando as eqs. (15) ou (17), o incremento Δx tende a zero, mas a raiz convergida ainda fica distante do valor exato, não atingindo precisão em todos os dígitos disponíveis.

Uma terceira alternativa, mais robusta, para se determinar raízes quando $f(x) \rightarrow 0$ e $f'(x) \rightarrow 0$, pode ser a aplicação da **Regra de L'Hospital** ao cálculo de Δx para resolver a indeterminação numérica que ocorre no limite do processo de convergência, como aplicado por **Galántai e Hegedus (2010)**.



Na obra “A study of accelerated Newton methods for multiple polynomial roots”, os

autores apresentam uma abrangente revisão bibliográfica sobre aceleradores de convergência do método de Newton e formas de aproximação da estimativa do valor da multiplicidade.

Neste livro, vamos propor um algoritmo alternativo aplicando a Regra de L’ Hospital como acelerador do método de Newton, primeiramente para funções não polinomiais.

Pela Regra de L'Hospital:

sejam $f_1(x)$ e $f_2(x)$ funções contínuas e deriváveis em um intervalo I , com $f'_2(x) \neq 0$, $\forall x \in I$, e seja a um valor interno de I , tal que $f_1(a) = f_2(a) = 0$, supondo que existe $\lim_{x \rightarrow a} \left(\frac{f_1(x)}{f_2(x)} \right)$, então existe $\lim_{x \rightarrow a} \left(\frac{f_1(x)}{f_2(x)} \right)$ e $\lim_{x \rightarrow a} \left(\frac{f_1(x)}{f_2(x)} \right) = \lim_{x \rightarrow a} \left(\frac{f'_1(x)}{f'_2(x)} \right)$.

Nesse caso, vamos considerar que, no limite da convergência iterativa, teremos uma indeterminação $0/0$ na eq. (11a) de Δx , quando $f(x) \rightarrow 0$ e $f'(x) \rightarrow 0$ para a raiz $x \rightarrow a$ convergida.

A Regra de L'Hospital pode ser diretamente aplicável na equação do cálculo do incremento Δx , eq. (11a), nesse caso, com $f(x) \rightarrow 0$ e $f'(x) \rightarrow 0$, teremos

$$\lim_{x_k \rightarrow a} (\Delta x) = \lim_{x_k \rightarrow a} \left(-\frac{f(x_k)}{f'(x_k)} \right) = \lim_{x_k \rightarrow a} \left(-\frac{f'(x_k)}{f''(x_k)} \right)$$

Então, na região deconvergência, $x_k \rightarrow a$, teremos:

$$\Delta x = -\frac{f'(x_k)}{f''(x_k)} \quad (19)$$

Com x_{k+1} dado pela mesma eq. (11b).

Exemplo 3.12: reaplique o método de Newton, modificado pela Regra de L'Hospital, eq. (19), para obter uma raiz de $e^x - x - 1 = 0$ a partir de $x_0 = 1.0$.

Solução:

$$\begin{aligned} f(x) &= e^x - x - 1 \\ f'(x) &= e^x - 1 \\ f''(x) &= e^x \end{aligned}$$

Podemos aplicar a fórmula do método de Newton modificado, conforme a eq. (19), pois sabemos previamente dos **Exemplos 3.10 e 3.11** que, na região da raiz, temos $f(x) \rightarrow 0$ e $f'(x) \rightarrow 0$, então:

Tabela 3.11 – Tabela de resultados do Exemplo 3.12

k	x_k	$f'(x_k)$	$f''(x_k)$	$\Delta x = -\frac{f'(x_k)}{f''(x_k)}$	$x_{k+1} = x_k + \Delta x$	$f(x)$	$ f(x) + \Delta x $
0	1.000000e+00	1.718282e+00	2.718282e+00	-6.321206e-01	3.678794e-01	7.678842e-02	7.089090e-01
1	3.678794e-01	4.446679e-01	1.444668e+00	-3.077994e-01	6.008007e-02	1.841501e-03	3.096409e-01
2	6.008007e-02	6.192157e-02	1.061922e+00	-5.831087e-02	1.769199e-03	1.565957e-06	5.831244e-02
3	1.769199e-03	1.770765e-03	1.001771e+00	-1.767635e-03	1.564111e-06	1.223022e-12	1.767635e-03
4	1.564111e-06	1.564112e-06	1.000002e+00	-1.564110e-06	1.223322e-12	0.000000e+00	1.564110e-06
5	1.223322e-12	1.223244e-12	1.000000e+00	-1.223244e-12	7.783746e-17	0.000000e+00	1.223244e-12
6	7.783746e-17	0.000000e+00	1.000000e+00	0.000000e+00	7.783746e-17	0.000000e+00	0.000000e+00
7	7.783746e-17						

Fonte: Elaboração própria

Assim, em $k=7$ iterações, calculamos $x_7 = 7.78374589094591e-17$, com $\Delta x = 0$ e $f(x)=0$, logo $x_7 = 0.000000000000000778374589094591$ é a raiz encontrada com 16 dígitos significativos (sublinhados).

Considerações sobre o método de Newton modificado pela Regra de L'Hospital:

- a) esse processo iterativo corrige a convergência quadrática, em que Δx na iteração $k+1$ é o quadrado do Δx da iteração k anterior, conforme os valores de Δx na tabela de resultados do **Exemplo 3.12** (observe que o expoente negativo da notação científica de cada Δx dobra de valor a cada iteração); e
- b) nesse processo é necessário conhecer previamente o comportamento da função e das suas derivadas ao longo da convergência, no caso, a eq. (19) é válida somente para $f(x) \rightarrow 0$ e $f'(x) \rightarrow 0$. Se as demais derivadas também tenderem a zero, será necessário reaplicar a Regra de L'Hospital.

Vamos aprofundar a questão de "derivadas também tenderem a zero" ao longo do processo de convergência na seção 3.3.6 sobre equações polinomiais.

3.2.2.5 Rotinas de Funções Predefinidas

O método de Newton também pode ser utilizado para elaborar rotinas de cálculo de valores de funções predefinidas para máquinas digitais, conforme os exemplos a seguir.

3.2.2.5.1 Recíproco de um Número

Fazemos uso do método de Newton para elaborar uma rotina dedicada a obter o recíproco $\frac{1}{c}$ com $c \in \mathbb{C} - \{0\}$ (complexos não nulos).

Com essa rotina, desejamos determinar um x tal que

$$x = \frac{1}{c} \text{ ou } \frac{1}{x} = c \Rightarrow f(x) = \frac{1}{x} - c = 0$$

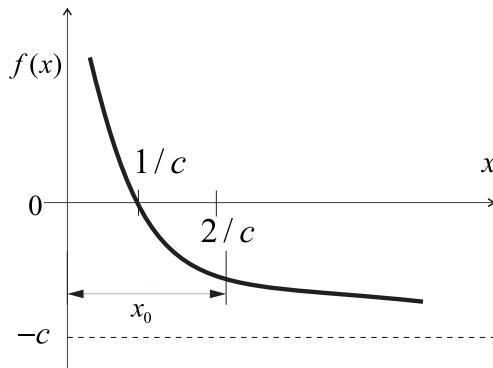
Determinando a raiz dessa equação, sem usar a divisão, teremos o recíproco de c . Aplicando Newton, conforme as eqs. (11a) e (11b), com

$$f(x) = \frac{1}{x} - c = 0 \text{ e } f'(x) = -\frac{1}{x^2} \Rightarrow \Delta x = -\frac{\frac{1}{x_k} - c}{-\frac{1}{(x_k)^2}} = (x_k)^2 \left(\frac{1}{x_k} - c \right)$$

teremos $x_{k+1} = x_k + \Delta x \Rightarrow x_{k+1} = x_k (2 - c * x_k)$.

Qual deverá ser o valor inicial x_0 para que essa expressão sempre convirja?

No Gráfico 3.12, temos a representação da função geradora $f(x) = \frac{1}{x} - c$ para $c > 1$.

Gráfico 3.12 – Representação de $f(x) = 1/x - c$ para $c > 1$ 

Fonte: Elaboração própria

Devemos estimar o valor inicial x_0 mais próximo de $\frac{1}{c}$, mas para garantirmos a convergência, o valor inicial x_0 deve estar contido no intervalo $0 < x_0 < \frac{2}{c}$. Por exemplo, se c é da ordem de $O(10^e) \Rightarrow$ o valor inicial estimado para x_0 deverá ser da ordem de $O(10^{-e})$.

Exemplo 3.13: obtenha o recíproco de $c = 425.32$ sem utilizar a operação de divisão.

Solução:

Considerando que c é da ordem de $O(10^3)$, pois $c = 0.42532e+3$, tomamos $x_0 = 0.001 = 10^{-3}$ e aplicamos a equação $\Rightarrow x_{k+1} = x_k (2 - c * x_k)$:

Tabela 3.12 – Tabela de resultados do Exemplo 3.13

k	x_k	$ \Delta x $
0	0.0010000000000000	
1	0.0015746800000000	5.7468000000000e-04
2	0.00209472925400723	5.20049254007232e-04
3	0.00232320085777991	2.28471603772682e-04
4	0.00235083814577217	2.76372879922607e-05
5	0.00235117083601085	3.32690238675701e-07
6	0.00235117088309978	4.70889325795976e-11
7	0.00235117088309978	8.67361737988404e-19

Fonte: Elaboração própria

Então, temos o valor equivalente ao exato $1 / 425.32 = 0.00235117088309978$ em 7 iterações, com critério de parada $|\Delta x| < 10^{-16}$, convergido no limite da precisão da variável *double*.

No **Caderno de Algoritmos**, confira o algoritmo de Newton para o recíproco de c no arquivo **Cap3reciproco.m**.

3.2.2.5.2 Raiz Quadrada de um Número

Fazemos uso do método de Newton para criar uma rotina que possa obter \sqrt{c} , $c \in \mathbb{R}^+$. Com essa rotina, desejamos um x tal que $x = \sqrt{c}$, o que equivale a obter a raiz positiva da equação polinomial quadrática $x^2 - c = 0$ sem usar a radiciação.

Desse modo, aplicamos o método de Newton, conforme as eqs. (11a) e (11b), com: $f(x) = x^2 - c$ e $f'(x) = 2x$,

$$\Delta x = -\frac{(x_k)^2 - c}{2 x_k}$$

$$x_{k+1} = x_k - \left(\frac{(x_k)^2 - c}{2 x_k} \right) \Rightarrow \left(x_k + \frac{c}{x_k} \right) 0.5$$

Como a $f(x) = x^2 - c$ é estritamente crescente e convexa para todo x positivo, então a expressão anterior do x_{k+1} gera uma sequência sempre convergente para raiz quadrada de c , para todo $x_0 > 0$.

A questão é: como tomar um x_0 para que essa convergência seja a mais rápida possível?

Uma boa alternativa é tomar

$$x_0 = \left(1.68 - \frac{1.29}{0.84 + m} \right) * 10^p (3.16^q) \quad (20)$$

Em que m é a mantissa fracionária de x dada por $(0 . d_1 \dots d_t)$.

Esse valor inicial é o resultado da extração da raiz quadrada de c , expresso na forma $c = (0 . d_1 \dots d_t) 10^{2p+q}$, com p inteiro e $q = 0$ ou $q = 1$, e a expressão entre parênteses sendo uma aproximação ajustada para a mantissa fracionária $m = (0 . d_1 \dots d_t)$.

Exemplo 3.14: obtenha $\sqrt{685.72} \Rightarrow c = 0.68752 * 10^3$.

Solução:

Tomamos o valor inicial:

$$\begin{cases} M = 0.68572 \text{ (mantissa)} \\ p = 1 \\ q = 1, \text{ pois } 2p + q = 3 \end{cases}$$

$$x_0 = \left(1.68 - \frac{1.29}{0.84 + 0.68572} \right) 10^1 \quad (3.16)^1 = 26.370122$$

Tabela 3.13 – Tabela de resultados do Exemplo 3.14

k	x_k	$ \Delta x / x $
0	26.370122	
1	26.18689694144919144	6.996822e-03
2	26.18625594488216635	62.447836e-05
3	26.18625593703689702	2.995949e-10
4	26.18625593703689702	0

Fonte: Elaboração própria

Então, temos o resultado equivalente a $\sqrt{685.72} = 26.18625593703689702$ em 4 iterações, com critério de parada relativo $|\Delta x / x| < 10^{-16}$,收敛ido em variável *double*.

O algoritmo de Newton para determinar a raiz quadrada de c está disponível no **Caderno de Algoritmos** no arquivo **Cap3raizquadrada.m**.

3.3 Solução de Equações Polinomiais

Vamos abordar a solução de equações da classe das **Polinomiais** devido à sua importância histórica, à frequência com que aparecem nos modelos matemáticos e às particularidades algébricas desse tipo de equações, a partir de metodologias clássicas (BURDEN; FAIRES, 2011; CHENEY; KINCAID, 2008), e uma proposta alternativa a respeito da determinação de raízes múltiplas baseada em Galántai e Hegedus (2010).

Uma equação polinomial é toda expressão do tipo $P_n(x) = 0$, em que $P_n(x) = a_1x^n + a_2x^{n-1} + \dots + a_nx + a_{n+1}$ é um polinômio de grau n se $a_1 \neq 0$.

A solução das equações $P_n(x) = 0$ tem o seguinte histórico de desenvolvimento:

$$\begin{aligned} n=1 \Rightarrow P_1(x) &= a_1x^1 + a_2 = 0 && \text{(Euclides, séc. III, a.C.)} \\ \Rightarrow \alpha &= \frac{a_2}{a_1} \end{aligned}$$

$$\begin{aligned} n=2 \Rightarrow P_2(x) &= a_1x^2 + a_2x^1 + a_3 = 0 && \text{(Sridhara/Bhaskara, séc. XII)} \\ \Rightarrow \alpha &= \frac{-a_2 \pm \sqrt{a_2^2 - 4a_1a_3}}{2a_1} \end{aligned}$$

$$n=3 \Rightarrow P_3(x) = a_1x^3 + a_2x^2 + a_3x^1 + a_4 = 0 \quad \text{(Cardano/Tartaglia, 1535)}$$

Para resolvê-la por radiciação, substituímos $x = y - a_2 / (3a_1)$ em $P_3(x) = 0 \Rightarrow y^3 + p y + q = 0$, em que $p = a_3 - a_2^2 / (3a_1)$ e $q = a_4 + (2a_2^3 - 9a_2a_3) / (27a_1)$ gera as soluções:

$$\begin{aligned} \Rightarrow \alpha_1 &= \sqrt[3]{-\frac{q}{2} + \sqrt{\frac{p^3}{27} + \frac{q^2}{4}}} + \sqrt[3]{-\frac{q}{2} - \sqrt{\frac{p^3}{27} + \frac{q^2}{4}}} \\ \Rightarrow \alpha_2 &= \sqrt[3]{-\frac{q}{2} + \sqrt{\frac{p^3}{27} + \frac{q^2}{4}}} + \sqrt[3]{-\frac{q}{2} + \sqrt{\frac{p^3}{27} + \frac{q^2}{4}}} \\ \Rightarrow \alpha_3 &= \sqrt[3]{-\frac{q}{2} - \sqrt{\frac{p^3}{27} + \frac{q^2}{4}}} + \sqrt[3]{-\frac{q}{2} - \sqrt{\frac{p^3}{27} + \frac{q^2}{4}}} \end{aligned}$$

$$n=4 \Rightarrow P_4(x) = a_1x^4 + a_2x^3 + a_3x^2 + a_4x^1 + a_5 = 0 \quad \text{(Ferrari, aprox. 1550)}$$

O método de Ferrari permite a redução de qualquer equação de grau $n = 4$ para $n = 3$.

$$n=5 \Rightarrow P_5(x) = a_1x^5 + a_2x^4 + a_3x^3 + a_4x^2 + a_5x^1 + a_6 = 0$$

Em 1824, Abel provou ser impossível solver por **radiciação** todas as $P_n(x) = 0$ para $n \geq 5$ e, em 1832, Galois classificou as possíveis e impossíveis.

Em decorrência da impossibilidade de solver por radiciação todas as polinomiais, precisamos usar outras metodologias. Neste estudo, continuaremos fazendo uso dos métodos iterativos.



Uma característica importante das equações $P_n(x) = 0$ é que sabemos previamente quantas raízes elas possuem. Para tanto, basta usar o teorema fundamental da álgebra e seus corolários, os quais provam que:

Propriedade 1: toda $P_n(x) = 0$ possui exatamente n soluções α , reais ou complexas, distintas ou repetidas (múltiplas).

Como vamos usar métodos iterativos para solver uma $P_n(x) = 0$ qualquer, vamos abordar antes algumas formas de localização das raízes Reais e Complexas, para facilitar a escolha de valores iniciais x_0 .

3.3.1 Localização de Raízes de Equações Polinomiais

Conforme vimos no início deste Capítulo, uma raiz real de $f(x) = 0$ pode ser localizada via teorema de Bolzano. Nesta seção, vamos apresentar algumas propriedades algébricas dos polinômios que ampliam as possibilidades de localização de todas as n raízes de $P_n(x) = 0$, Reais e/ou Complexas.

A determinação de intervalos $[a, b]$ e círculos, que contenham respectivamente todas as raízes Reais e Complexas, pode ser estabelecida por cotas-limite. Vamos apresentar algumas dessas cotas a seguir.

Propriedade 2 – Cota do Módulo Máximo: em uma $P_n(x) = a_1x^n + a_2x^{n-1} + \dots + a_nx + a_{n+1} = 0$, com $a_1 \neq 0$, para toda raiz $\alpha \in \mathbb{C} \Rightarrow |\alpha| < r_{\max}$, em que $r_{\max} = 1 + \frac{\max\{|a_2|, |a_3|, \dots, |a_{n+1}|\}}{|a_1|}$.

Esse limite máximo r_{\max} de todas as raízes de $P_n(x) = 0$ também é atribuído a Cauchy, embora seja normalmente menos exato do que a chamada "cota de Cauchy", que veremos na **Propriedade 3**.

Dessa forma,

- a) se α for Real, ela estará sempre contida no intervalo $(-r_{\max}, +r_{\max})$, que poderá ser rastreado por um processo de varredura usando o teorema de Bolzano para localizar subintervalos menores que contenham(m) raiz(es) Real(is); e
 - b) se a raiz for um Complexo, então estará contida no círculo C de centro $(0, 0)$ e raio r_{\max} .
-
-

Lembre-se que o módulo de um número complexo $a + b * i$ é dado por $\sqrt{a^2 + b^2}$ em que ($i = \sqrt{-1}$).

Podemos também calcular um raio mínimo r_{\min} para as raízes de $P_n(x) = 0$, ou seja, um raio de um círculo interno que não contenha raízes, conforme a Figura 3.1.

Para calcular o raio mínimo, geramos um polinômio auxiliar $P_n^{**}(x) = x^n P_n(1/x)$ cujos zeros são os recíprocos das raízes de $P_n(x) = 0$, desde que o polinômio original $P_n(x)$ não contenha zeros nulos ($a_{n+1} \neq 0$), pois os seus recíprocos gerariam divisão por zero.

Então, para $P_n(x) = 0$ com $a_{n+1} \neq 0$, temos:

$$P_n(1/x) = a_1(1/x)^n + a_2(1/x)^{n-1} + \dots + a_n(1/x) + a_{n+1} = 0$$

Então, $P_n^{**}(x) = x^n P_n(1/x)$ resulta em:

$$P_n^{**}(x) = a_1 + a_2 x + \dots + a_n x^{n-1} + a_{n+1} x^n = 0 \quad (21)$$

Então, podemos aplicar a mesma cota máxima à $P_n^{**}(x) = 0$ e determinar um raio máximo auxiliar r_2 , limite superior das raízes de $P_n^{**}(x) = 0$, via

$$r_2 = 1 + \frac{\max \{|a_1|, |a_2|, |a_3|, \dots, |a_n|\}}{|a_{n+1}|} \quad (22)$$

Observe na eq. (22) que é necessário $a_{n+1} \neq 0$, pois esse coeficiente que se tornou o de maior grau em $P_n^{**}(x)$ passa ao denominador.

O recíproco desse limite r_2 corresponde ao raio mínimo das raízes de $P_n(x) = 0$, ou seja,

$$r_{\min} = 1/r_2 \quad (23a)$$

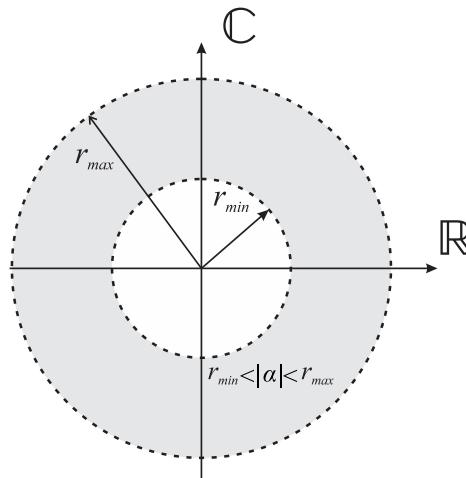
Neste caso,

$$r_{\min} = 1 \left/ \left(1 + \frac{\max \{ |a_1|, |a_2|, |a_3|, \dots, |a_n| \}}{|a_{n+1}|} \right) \right. \quad (23b)$$

Logo, a maior raiz de $P_n^{**}(x) = 0$ está dentro do limite r_2 , mas essa sua maior raiz corresponde à menor raiz de $P_n(x) = 0$.

Lembramos que a **Propriedade 2** vale para raízes Reais e Complexas, ou seja, esses limites do módulo das raízes podem ser interpretados em um gráfico cartesiano com eixo Real e Imaginário (eixo puramente Complexo), onde toda raiz α de $P_n(x) = 0$ está contida no intervalo $r_{\min} < |\alpha| < r_{\max}$, ou seja, α está dentro do anel destacado em cinza da Figura 3.13.

Figura 3.13 – Anel que contém todas as raízes α de $P_n(x) = 0$



Fonte: Elaboração própria

Exemplo 3.15: localize as raízes de $3x^6 + 4x^3 - 2x^2 - 6 = 0$ por meio das cotas-limite de **módulo máximo e mínimo**.

Solução:

Como $P_6(x) = 3x^6 + 4x^3 - 2x^2 - 6 = 0$, aplicando a **Propriedade 2**, com $a_1 = 3$ e $\max\{|0|, |0|, |4|, |-2|, |0|, |-6|\} = 6$, temos:

$$r_{\max} = 1 + \frac{6}{|3|} = 3$$

Para calcular um raio mínimo para essas raízes, podemos aplicar a eq. (23b):

$$r_{\min} = 1 \left/ \left(1 + \frac{\max\{|3|, |0|, |0|, |4|, |-2|, |0|\}}{|-6|} \right) \right.$$

$$r_{\min} = 0.6$$

Assim, as raízes de $P_6(x) = 0$ estão entre $0.6 < |\alpha| < 3.0$.

Esse resultado pode ser comprovado quando determinamos as 6 raízes de $P_6(x) = 0$ (via função *roots()* do Octave), conforme segue:

$$\begin{aligned} x_1 &= -1.373431183527068 + 0.000000000000000i \\ x_2 &= 0.580108028676416 + 1.135516978130294i \\ x_3 &= 0.580108028676416 - 1.135516978130294i \\ x_4 &= 1.035657483096530 + 0.000000000000000i \\ x_5 &= -0.411221178461147 + 0.834073854054218i \\ x_6 &= -0.411221178461147 - 0.834073854054218i \end{aligned}$$

cuja raiz de módulo:

- a) máximo é x_1 (módulo 1.373431183527068);
- b) mínimo é x_5 (módulo 0.929936584736740).

Propriedade 3 – Cota de Cauchy: dada uma equação polinomial $P_n(x) = 0$, então todas as suas raízes α Reais ou Complexas satisfazem $|\alpha| \leq r_{\max}$, sendo

$$r_{\max} = \lim_{k \rightarrow \infty} x_k$$

$$\text{com } x_{k+1} = \left(\sum_{i=2}^{n+1} \left| \frac{a_i}{a_1} \right| x_k^{n-i+1} \right)^{1/n}$$

Nessa propriedade, são aplicados k passos iterativos a partir do valor inicial $x_0 = 0$ até que algum critério de parada seja satisfeito.

Exemplo 3.16: calcule as cotas-limite máxima e mínima de Cauchy das raízes da equação polinomial $3x^6 + 4x^3 - 2x^2 - 6 = 0$, com precisão de 16 dígitos significativos.

Solução:

A partir de $x_0 = 0$ (valor inicial) e $a_1 = 3$, geramos a equação iterativa, como se o x do termo de maior grau fosse isolado, análogo a um método de iteração linear, considerando coeficientes em módulo e com valores de x_k tomados em um passo iterativo anterior, logo:

$$x_{k+1} = \left((4x_k^3 + 2x_k^2 + 6) / 3 \right)^{1/6}$$

Nesse caso, para uma cota com 16 dígitos significativos, então $|x_{k+1} - x_k| < 10^{-15}$:

Tabela 3.13 – Tabela de resultados do Exemplo 3.16

k	x_k	$ x_{k+1} - x_k $
0	0	–
1	1.12246204830937	1.12246204830937
2	1.29541512584639	0.172953077537016
3	1.34864877400131	0.0532336481549256
:		
31	1.37343118352707	6.66133814775094e-16

Fonte: Elaboração própria

O valor de x_k com 16 dígitos convergidos é o $r_{\max} = 1.37343118352707$ com $|x_{k+1} - x_k| \approx 0$.

Para calcular o raio mínimo das raízes de $P_6(x) = 3x^6 + 4x^3 - 2x^2 - 6 = 0$ pela cota de Cauchy, calculamos o raio interno de um anel que “não” contenha raízes por meio do polinômio auxiliar

$P_n^{**}(x) = x^n P_n(1/x)$, definido pela eq. (21), que contém os inversos multiplicativos das raízes de $P_6(x) = 0$:

$$P_6^{**}(x) = x^6 P_6(1/x) = 3 + 4x^3 - 2x^4 - 6x^6 = 0$$

Então, a partir de $x_0 = 0$ (valor inicial), geramos a equação iterativa a seguir:

$$x_k = \left((2x_k^4 + 4x_k^3 + 3) / 6 \right)^{1/6}$$

Depois da convergência, concluímos que a maior raiz de $P_6^{**}(x)$ está dentro do limite $r_2 = 1.11793643455463$, mas essa maior raiz de $P_6^{**}(x)$ corresponde à menor raiz de $P_6(x) = 0$. Logo, o limite inferior das raízes de $P_6(x) = 0$ é dado pela eq. (23a),

$$r_{\min} = 1 / r_2 = 0.894505241166406$$

Portanto, pela cota de Cauchy, todas as 6 raízes de $P_6(x) = 0$ também estão no intervalo $0.894505241166406 \leq \alpha \leq 1.37343118352707$.

Propriedade 4 – Cota de Kojima: dada a equação polinomial $P_n(x) = 0$ ($n \geq 2$), toda a raiz α , Real ou Complexa, satisfaz $\alpha \leq r_{\max} \mid$ e $r_{\max} = \max_1 + \max_2$. Os valores de \max_1 e \max_2 são os dois maiores valores da sequência:

$$q_i = \left\{ \left| \frac{a_{i+1}}{a_i} \right|^{\frac{1}{i}} \right\} \text{ com } i = 1, \dots, n.$$

Exemplo 3.17: calcule as cotas-limite de raízes de Kojima para o polinômio $P_6(x) = 3x^6 + 4x^3 - 2x^2 - 6 = 0$.

Solução:

Como os coeficientes são $a_1 = 3$, $a_2 = 0$, $a_3 = 0$, $a_4 = 4$, $a_5 = -2$, $a_6 = 0$ e $a_7 = -6$, então a sequência de fatores q_i é:

$$q_i = \left\{ \left| \frac{0}{3} \right|^{\frac{1}{1}}, \left| \frac{0}{3} \right|^{\frac{1}{2}}, \left| \frac{4}{3} \right|^{\frac{1}{3}}, \left| \frac{-2}{3} \right|^{\frac{1}{4}}, \left| \frac{0}{3} \right|^{\frac{1}{5}}, \left| \frac{-6}{3} \right|^{\frac{1}{6}} \right\} = \{0, 0, 1.100642, 0.903602, 0, 1.122462\}$$

Tomando os dois maiores valores: $\max_1 = 1.122462$ e $\max_2 = 1.100642$, temos

$$r_{\max} = \max_1 + \max_2 = 2.223104$$

Para determinar o raio interno do anel, podemos gerar o polinômio auxiliar dado pela eq. (21),

$$P_6^{**}(x) = -6x^6 + 0x^5 - 2x^4 + 4x^3 + 0x^2 + 0x + 3 = 0,$$

que contém os inversos multiplicativos das raízes de $P_6(x) = 0$, cuja sequência de fatores q_i de $P_6^{**}(x) = -6x^6 + 0x^5 - 2x^4 + 4x^3 + 0x^2 + 0x + 3 = 0$ é:

$$\left\{ \left| \frac{0}{-6} \right|^{\frac{1}{1}}, \left| \frac{-2}{-6} \right|^{\frac{1}{2}}, \left| \frac{4}{-6} \right|^{\frac{1}{3}}, \left| \frac{0}{-6} \right|^{\frac{1}{4}}, \left| \frac{0}{-6} \right|^{\frac{1}{5}}, \left| \frac{3}{-6} \right|^{\frac{1}{6}} \right\} = \{0, 0.577350, 0.873580, 0, 0, 0.890899\}$$

Tomando $\max_1 = 0.890899$ e $\max_2 = 0.873580$, temos

$$r_2 = \max_1 + \max_2 = 1.764479$$

então a cota mínima interna dada pela eq. (23a) é:

$$r_{\min} = 1/r_2 = 0.5667395.$$

Logo, as seis raízes de $P_6(x) = 0$ também estão no intervalo $0.5667395 \leq \alpha \leq 2.223104$.

Comparando as 3 cotas de $P_6(x) = 0$:

- a) $|\alpha| \in (0.6, 3.0)$ cota do módulo máximo – **Propriedade 2**;
- b) $|\alpha| \in [0.8945041, 1.373431]$ cota de Cauchy – **Propriedade 3**;
- c) $|\alpha| \in [0.5667395, 2.223104]$ cota de Kojima – **Propriedade 4**.

Como as três cotas garantem a inclusão de todas as raízes, então podemos tomar o menor intervalo possível, com o maior r_{\min} e o menor r_{\max} , ou seja, tomar o anel mais estreito. Entre os **Exemplos 3.15, 3.16 e 3.17**, ambos os limites do menor intervalo foram dados pela cota de Cauchy.

Dispondo das cotas máxima e mínima, podemos fazer uma varredura de possíveis raízes reais subdividindo-o em subintervalos e testando cada um com o teorema de Bolzano, no lado negativo entre $[-r_{\max}, -r_{\min}]$ e no lado positivo entre $[+r_{\max}, +r_{\min}]$, verificando as trocas de sinais da função. Essa varredura de possíveis raízes reais com o teorema de Bolzano só é válida para equações polinomiais de coeficientes Reais.

Por último, podemos completar a estimativa das n raízes com valores iniciais de raízes Complexas, distribuídos “randomicamente” dentro do anel de raio $r \in [r_{\min}, r_{\max}]$, conforme o algoritmo estabelecido no arquivo **fLocaliza.m** que apresentamos no **Caderno de Algoritmos**.

Para as raízes de $P_6(x) = 3x^6 + 4x^3 - 2x^2 - 6 = 0$, localizamos dois valores iniciais Reais, conforme o teorema de Bolzano, enquanto os 4 seguintes são estabelecidos como Complexos, para abranger possíveis raízes Reais ou Complexas.

Um possível resultado para o vetor de raízes iniciais xi gerado pela função $fLocaliza(n, a)$ seria:

$$\begin{aligned} xi_1 &= -1.365449094487721 + 0.0000000000000000i \\ xi_2 &= +1.030200914835262 + 0.0000000000000000i \\ xi_3 &= -0.937643005591872 + 0.852667185210998i \\ xi_4 &= -0.709960325540485 - 0.887984203422577i \\ xi_5 &= 0.846274902134810 + 0.716025620017412i \\ xi_6 &= 0.741095894499941 + 0.654794134744737i \end{aligned}$$

Quando não temos certeza da existência de raízes Reais, devemos sempre adotar valores iniciais Complexos, que podem convergir para raízes Complexas ou Reais, conforme o caso. Entretanto, um valor inicial real jamais convergirá para um Complexo pelo método de Newton.

Para melhorar a localização de raízes, vamos estudar a natureza das raízes de equações polinomiais, se são Reais e/ou Complexas, usando algumas propriedades específicas.

3.3.2 Natureza de Raízes de Equações Polinomiais

Nesta seção, vamos apresentar algumas propriedades que qualificam as raízes quanto à sua natureza, embora as cotas que apresentamos anteriormente já permitam uma boa localização de raízes iniciais.

Propriedade 5: as **raízes Complexas** de equações polinomiais de coeficientes **Reais** sempre ocorrem aos pares conjugados, isto é, se $\alpha = a + b * i$ for raiz, $\alpha = a - b * i$ também será raiz.

Exemplo 3.18: determine as raízes de $x^2 - x + 2 = 0$.

Solução:

$$\text{Pela solução de equações de } 2^{\text{o}} \text{ grau, temos: } \alpha = \frac{1 \pm \sqrt{1-8}}{2} \Rightarrow \alpha = \frac{1 \pm \sqrt{-7}}{2}$$

$$\alpha_1 = 0.5 + \sqrt{\frac{7}{4}} i \quad \text{e} \quad \alpha_2 = 0.5 - \sqrt{\frac{7}{4}} i$$

Consequência:

Toda $P_n(x) = 0$ de coeficientes reais com grau ímpar possui ao menos uma raiz Real.

Propriedade 6: em $P_n(x) = 0$, com $a_{n+1} \neq 0$, se n for **par** e $a_1 * a_{n+1} < 0$ \Rightarrow existirão ao menos duas raízes Reais, uma positiva e outra negativa.

Por exemplo, em $1x^{12} - 5x^2 + 3x - 2 = 0$, temos que $a_1 * a_{n+1} = (1) * (-2) < 0$, logo possui pelo menos uma raiz Real positiva e uma negativa.

Propriedade 7: em $P_n(x) = 0$, com $a_{n+1} \neq 0$, se existirão algum $k = 2, 3, \dots, n$, tal que $a_k^2 \leq a_{k+1} * a_{k-1}$ \Rightarrow existem raízes complexas.

Essa propriedade é uma condição suficiente, mas não necessária, ou seja, se não for satisfeita, nada poderemos afirmar sobre a existência de raízes Complexas.

Propriedade 8: regra dos sinais de Descartes para polinômios com **coeficientes Reais**.

Segundo essa regra, em polinômios com os termos colocados em ordem crescente ou decrescente de grau, o número de zeros Reais positivos é igual ao número de permutações de sinal dos coeficientes não nulos de $P_n(x)$ ou menor por uma diferença par. O número de zeros Reais negativos do polinômio é igual ao número de permutações de sinais dos coeficientes de $P_n(-x)$ ou menor também por uma diferença par.

Exemplo 3.19: determine o quadro de possibilidades das raízes de $x^5 + x^4 - x^3 - x^2 = 0$.

Solução:

Podemos inicialmente **fatorar** os binômios com as duas **raízes nulas**, caso existam, ou aplicar diretamente a regra de Descartes, considerando as raízes nulas no quadro de possibilidades.

$$P_5(x) = +x^5 + \underbrace{x^4 - x^3}_{1 \text{ troca}} - x^2 = (x-0)^2 (+x^3 + \underbrace{x^2 - x^1}_{1 \text{ troca}} - x^0)$$

Então, ocorre “uma” mudança de sinal entre o segundo e o terceiro coeficientes. Portanto, $P_5(x)$ possui apenas uma raiz Real positiva. Para obter as possibilidades do número de raízes Reais negativas, geramos um polinômio auxiliar, $P_5^*(x) = P_5(-x)$, cujas raízes positivas de $P_5^*(x)$, correspondem às raízes negativas do polinômio original $P_5(x)$:

$$P_5^*(x) = P_5(-x) = (-x)^5 + (-x)^4 - (-x)^3 - (-x)^2 = \underbrace{-x^5 + x^4}_{1 \text{ troca}} + \underbrace{x^3 - x^2}_{1 \text{ troca}} = 0$$

esse polinômio auxiliar $P_5^*(x)$ tem “duas” permutações de sinal (até duas raízes positivas), logo o polinômio original $P_5(x)$ possui até duas raízes negativas, mas se for menor, por uma diferença “par”, podemos não ter raízes negativas, conforme a Tabela 3.14:

Tabela 3.14 – Possibilidades de raízes reais e complexas

Nº positivas	Nº negativas	Nº Nulas	Nº Complexas (complemento para n)
1	2	2	0
1	0	2	2

Fonte: Elaboração própria

Para confirmar o resultado, observe a fatoração em binômios deste polinômio do **Exemplo 3.19**:

$$P_5(x) = +x^5 + x^4 - x^3 - x^2 = (x - (-1))^2(x - (+1))(x - 0)^2 = 0$$

Então, temos três raízes Reais distintas: -1 (aparece duas vezes, multiplicidade $M=2$), $+1$ (aparece uma vez) e uma nula (aparece duas vezes).

Sobre as propriedades que qualificam as raízes quanto à sua natureza, observamos que:

- a) podem ser incorporadas em um algoritmo de localização de raízes para definir com mais precisão os valores iniciais das raízes de $P_n(x) = 0$; e
- b) sempre que não forem localizadas raízes Reais em número suficiente, devemos usar valores iniciais Complexos, que são mais abrangentes, pois podem convergir para raízes Reais ou Complexas.

A seguir, vamos apresentar formas eficientes de calcular o valor numérico de uma função polinomial e de sua(s) derivada(s), para depois aplicar o método de Newton.

3.3.3 Cálculo Eficiente de Valores das Funções Polinomiais e suas Derivadas

Para obter valores numéricos de $P_n(x)$ e de suas derivadas, vamos mostrar qual é a forma mais eficiente de representar $P_n(x)$.

Se expressarmos $P_n(x)$ na forma padrão, considerando que as potências inteiros de x são calculadas por produtos e não por potenciações, reutilizando produtos efetuados, $x^{i+1} = \underbrace{x * x^i}_{1 \text{ multiplicação}}$, teremos que:

$$\begin{aligned} P_n(x) &= a_1 x^n + a_2 x^{n-1} + \dots + a_n x + a_{n+1} \Rightarrow \\ P_n(x) &= a_{n+1} + a_n * x + a_{n-1} * x * x + \dots + a_2 * x * x^{n-2} + a_1 * x * x^{n-1} \end{aligned}$$

Logo, $P_n(x = u)$ necessitará de $\begin{cases} n & \text{adições e} \\ 2n-1 & \text{multiplicações} \end{cases}$

Alternativamente, expressando $P_n(x)$ na forma linear de Horner, temos:

$$P_n(x) = \left(\dots \left((a_1 * x + a_2) * x + a_3 \right) * x + \dots + a_n \right) * x + a_{n+1}$$

Então, $P_n(x = u)$ necessitará apenas de $\begin{cases} n & \text{adições e} \\ n & \text{multiplicações} \end{cases}$

No **Caderno de Algoritmos**, apresentamos um algoritmo para calcular o valor numérico de um polinômio $P_n(x)$, em $x = u$, por Horner, no arquivo **fPnH1.m**.

Neste livro, vamos calcular $P_n(x)$ por meio de uma forma equivalente à de Horner, que é a divisão sintética de polinômios, ou regra de **Briot-Ruffini**, de acordo com a **Propriedade 9** a seguir.

Propriedade 9: o valor numérico de $P_n(x)$ em $x = u$ é o resto R_1 da primeira divisão de $P_n(x)$ por $(x - u)$.

Pelo algoritmo da divisão de polinômios, temos que

$$P_n(x) = (x - u)P_{n-1}(x) + R_1 \quad (24)$$

Então, fazendo $x = u$,

$$P_n(x = u) = (u - u)P_{n-1}(u) + R_1 = R_1 \rightarrow P_n(x = u) = R_1$$

Considerando o polinômio quociente dessa primeira divisão como $P_{n-1}(x) = b_1x^{n-1} + b_2x^{n-2} + \dots + b_n$, em que os coeficientes b_i podem ser obtidos da eq. (24), pela igualdade termo a termo, temos:

$$\begin{aligned} b_1 &= a_1 \\ b_i &= a_i + u * b_{i-1} \quad \forall i = 2 \text{ até } n+1 \end{aligned}$$

em que resto $R_1 = b_{n+1}$.

Então, se usarmos a divisão sintética de polinômios, o cálculo de $P_n(x = u) = R_1$ também necessitará apenas de n adições e n multiplicações.

Propriedade 10: se α é um zero de $P_n(x)$, então $P_n(x)$ será divisível por $(x - \alpha)$ e por consequência $R_1 = 0$.

Propriedade 11: o valor numérico de $P'_n(x)$ em $x = u$ é o resto R_2 da segunda divisão sucessiva de $P_n(x)$ por $(x - u)$.

Tomando a primeira divisão, temos:

$$P_n(x) = (x - u)P_{n-1}(x) + R_1$$

Efetuando uma segunda divisão sintética, via divisão de $P_{n-1}(x)$ por $(x - u)$, temos:

$$P_{n-1}(x) = (x - u)P_{n-2}(x) + R_2$$

Substituindo $P_{n-1}(x)$ em $P_n(x)$, temos:

$$P_n(x) = (x - u)((x - u)P_{n-2}(x) + R_2) + R_1$$

$$P_n(x) = (x - u)^2 P_{n-2}(x) + (x - u)R_2 + R_1$$

Derivando $P_n(x)$, reescrito na forma anterior, temos:

$$P'_n(x) = 2(x-u)^2 P_{n-2}(x) + (x-u)^2 P'_{n-2}(x) + R_2 + 0$$

Então, fazendo $x = u$,

$$P'_n(x=u) = 2(u-u)^2 P_{n-2}(u) + (u-u)^2 P'_{n-2}(u) + R_2 + 0 = R_2 \rightarrow \boxed{P'_n(x=u) = R_2}$$

A **Propriedade 11** pode ser generalizada para o cálculo dos valores de todas as n derivadas de ordem superior de uma função $P_n(x)$, conforme a **Propriedade 12**.

Propriedade 12: o valor da derivada k -ésima de $P_n(x)$ em $x = u$ é dado por $\frac{d^k P_n}{dx^k}(u) = P_n^{(k)}(u) = k! R_{k+1}$, em que R_{k+1} é o resto da $(k+1)$ -ésima divisão sucessiva de $P_n(x)$ por $(x-u)$.

Exemplo 3.20: obtenha $P_4(2)$, $P'_4(2)$, $P''_4(2)$, $P'''_4(2)$ e $P''''_4(2)$ em $P_4(x) = 2x^4 + 3x - 2$ aplicando a **Propriedade 12**.

Solução:

Primeira divisão: de $P_4(x) = 2x^4 + 3x - 2$ por $(x-2)$

$a_1 = 2$, $a_2 = 0$, $a_3 = 0$, $a_4 = 3$, $a_5 = -2$ (coeficientes do polinômio original):

$$b_1 = 2$$

$$b_2 = 0 + 2 * 2 = 4$$

$$b_3 = 0 + 4 * 2 = 8$$

$$b_4 = 3 + 8 * 2 = 19$$

$$b_5 = -2 + 19 * 2 = 36 \Rightarrow R_1 = 36$$

$$P_4(x=2) = P_4^{(0)}(x=2) = 0! * R_{0+1} = 1 * R_1 = 1 * 36 = 36$$

(também podemos obter o valor do polinômio pela **Propriedade 9**)

Segunda divisão: de $P_3(x) = 2x^3 + 4x^2 + 8x + 19$ por $(x-2)$

$a_1 = 2$, $a_2 = 4$, $a_3 = 8$, $a_4 = 19$ (esses coeficientes a_i foram atualizados pelos coeficientes b_i do quociente anterior):

$$\begin{aligned} b_1 &= 2 \\ b_2 &= 4 + 2 * 2 = 8 \\ b_3 &= 8 + 2 * 8 = 24 \end{aligned}$$

$$\begin{aligned} b_4 &= 19 + 24 * 2 = 67 \Rightarrow R_2 = 67 \\ P'_4(x=2) &= P_4^{(1)}(x=2) = 1! * R_{1+1} = 1! * R_2 = 1 * 67 = 67 \end{aligned}$$

Terceira divisão: de $P_2(x) = 2x^2 + 8x + 24$ por $(x - 2)$

$$a_1 = 2, a_2 = 8, a_3 = 24:$$

$$\begin{aligned} b_1 &= 2 \\ b_2 &= 8 + 2 * 2 = 12 \\ b_3 &= 24 + 12 * 2 = 48 \Rightarrow R_3 = 48 \\ P''_4(x=2) &= P_4^{(2)}(x=2) = 2! * R_{2+1} = 2! * R_3 = 2 * 48 = 96 \end{aligned}$$

Quarta divisão: de $P_1(x) = 2x + 12$ por $(x - 2)$

$$a_1 = 2, a_2 = 12:$$

$$\begin{aligned} b_1 &= 2 \\ b_2 &= 12 + 2 * 2 = 16 \Rightarrow R_4 = 16 \\ P'''_4(x=2) &= P_4^{(3)}(x=2) = 3! * R_{3+1} = 3! * R_4 = 3 * 2 * 1 * 16 = 96 \end{aligned}$$

Quinta divisão: do polinômio de grau $n = 0$, $P_0(x) = 2$ por $(x - 2)$

$$a_1 = 2.$$

Como não temos como fazer a divisão de um polinômio de grau $n = 0$ por um binômio $(x - 2)$ de grau $n = 1$, podemos tomar o quociente dessa quinta divisão como 0 (zero), e o resto como o próprio $P_0(x) = 2$, pois $P_0(x) = (x - 2) * 0 + 2$:

$$\begin{aligned} b_1 &= 2 \Rightarrow R_5 = 2 \\ P_4^{(4)}(x=2) &= P_4^{(4)}(x=2) = 4! * R_{4+1} = 4! * R_5 = 4 * 3 * 2 * 1 * (2) = 48 \end{aligned}$$

Podemos, ainda, validar a **Propriedade 12** comparando os resultados com as derivadas analíticas $P'_4(x) = 8x^3 + 3$, $P''_4(x) = 24x^2$, $P'''_4(x) = 48x$ e $P_4^{(4)}(x) = 48$.

Observamos que, dispondo das formas específicas de determinação de valores iniciais, do valor numérico do polinômio e de suas derivadas, podemos aplicar métodos de refinamento para as raízes de $P_n(x) = 0$. Aqui vamos aplicar o método de Newton, como mostraremos a seguir.

3.3.4 Método de Newton para $P_n(x) = 0$

A partir de um valor inicial x_0 , localizado previamente como apresentado nas seções 3.3.1 e 3.3.2, calculamos os valores de $P_n(x_0)$ e de $P'_n(x_0)$ na primeira iteração $k = 0$ pelas **Propriedades 9 e 11**:

$$\begin{aligned} P_n(x_k) &= R_1 \\ P'_n(x_k) &= R_2 \end{aligned}$$

Depois, calculamos o incremento de x dado pela eq. (11a) do **método de Newton**, conforme segue:

$$\Delta x = -\frac{P_n(x_k)}{P'_n(x_k)} = -\frac{R_1}{R_2} \quad (25)$$

Para polinomiais, o método de Newton também é chamado de Birge-Vieta.



e calculamos o valor novo $x_{k+1} = x_k + \Delta x$ dado pela eq. (11b).

Exemplo 3.21: calcule a raiz positiva de $2x^3 - x - 2 = 0$, por Newton, a partir de $x_0 = 1$, com 16 dígitos significativos exatos.

Solução:

$$P_3(x) = 2x^3 + 0x^2 - 1x - 2x^0 \quad (a_1 = +2, a_2 = 0, a_3 = -1, a_4 = -2)$$

Usando a divisão sintética de Briot-Ruffini para determinar os valores de $P_3(x)$ e $P'_3(x)$, temos:

Primeira iteração: a partir de $x_0 = 1$.

Primeira divisão: $P_3(x) = 2x^3 + 0x^2 - 1x - 2$ por $(x - x_0)$.

$$b_1 = a_1 = +2$$

$$b_2 = a_2 + b_1 * x_0 = 0 + 2 * 1 = 2$$

$$b_3 = a_3 + b_2 * x_0 = -1 + 2 * 1 = 1$$

$$b_4 = a_4 + b_3 * x_0 = -2 + 1 * 1 = -1 = R_1 = P_3(x_0 = 1)$$

Segunda divisão: $P_2(x) = 2x^2 + 2x + 1$ por $(x - x_0)$.

$$c_1 = b_1 = 2$$

$$c_2 = b_2 + c_1 * x_0 = 2 + 2 * 1 = 4$$

$$c_3 = b_3 + c_2 * x_0 = 1 + 1 * 4 = 5 = R_2 = P_3'(x_0 = 1)$$

Logo, $R_1 = -1$ e $R_2 = 5$:

$$\Delta x = -\frac{R_1}{R_2} = -\frac{(-1)}{5} = 0.2 \text{ e}$$

$$x_1 = x_0 + \Delta x = 1 + 0.2 = 1.2$$

Segunda iteração: a partir de $x_1 = 1.2$.

Primeira divisão: $P_3(x) = 2x^3 + 0x^2 - 1x - 2$ por $(x - x_1)$:

$$P_3(1.2) = R_1 = 0.256$$

Segunda divisão:

$$P_3'(1.2) = R_2 = 7.64$$

$$\Delta x = -0.256/7.64 = -0.0335078534031414 \text{ e}$$

$$x_2 = 1.2 + \Delta x = 1.16649214659686$$

Terceira iteração: a partir de $x_2 = 1.16649214659686$.

Primeira divisão: $P_3(x) = 2x^3 + 0x^2 - 1x - 2$ por $(x - x_2)$:

$$P_3(1.16649214659686) = R_1 = 0.00800874528246043$$

Segunda divisão:

$$P_3'(1.16649214659686) = R_2 = 7.16422356843290$$

$$\Delta x = -\frac{0.00800874528246043}{7.16422356843290} = -0.00111788042430008 \text{ e}$$

$$x_3 = 1.16649214659686 + \Delta x = 1.16537426617256$$

Até a terceira iteração, temos $\alpha_1 \approx 1.16537426617256$, com critério de parada $|\Delta x| \approx O(10^{-3})$.

Esse processo iterativo deve prosseguir até que algum critério de parada seja satisfeito quando os 16 dígitos significativos estiverem convergidos, o que ocorre quando o critério de parada

$|\Delta x| < 10^{-15}$ é atingido, depois de 6 iterações. Então, a primeira raiz convergida é $\alpha_1 \approx 1.165373043062415$.

Para determinar a segunda e a terceira raízes, e evitar obter novamente a primeira raiz α_1 , não basta partir de uma estimativa inicial diferente, precisamos efetuar **reduções ou deflações** do grau da seguinte forma:



Se não fizermos as reduções do grau para separar as raízes já determinadas, fatorando o polinômio, o(s) novo(s) valor(es) inicial(is) poderão convergir para raiz(es) já obtida(s).

- Tomamos $P_n(x)$ e o dividimos por $(x - \alpha_1)$, ou seja, obtemos $P_{n-1}(x)$; nesse caso, com resto R_1 numericamente nulo (residual), pois nas divisões exatas os restos seriam exatamente nulos. Esse polinômio de grau reduzido $P_{n-1}(x)$ não contém mais o zero α_1 .
- Reaplicamos uma nova estimativa inicial para esse novo zero do polinômio de grau reduzido com o mesmo método de Newton.
- Calculado o novo zero, retornamos ao item (a) até completar a redução de grau desejada, que pode ser até graus $n = 2, 1$ ou 0 , pois esses polinômios têm raízes de solução algébrica simples.

Exemplo 3.22: calcule a segunda e a terceira raiz de $2x^3 - x - 2 = 0$, sabendo que a primeira raiz é $\alpha_1 \approx 1.165373043062415$, com 16 dígitos significativos convergidos.

Solução:

Primeiramente, devemos reduzir o grau do polinômio $P_3(x)$, extraindo a 1ª raiz $\alpha_1 \approx 1.165373043062415$ já encontrada, fazendo $P_3(x) \div (x - \alpha_1)$, gerando um polinomio quociente de grau $n = 2$:

$$P_2(x) = 2x^2 + 2.33074608612483x + 1.71618865899311$$

com resto residual $R_1 = 4.44089209850063e-16$.

Logo, o polinômio original $P_3(x)$ fica fatorado da seguinte forma:

$$P_3(x) = (2x^2 + 2.33074608612483x + 1.71618865899311) * (x - \alpha_1) + 4.44089209850063e-16$$

Observe que essa divisão pode ser considerada numericamente exata, uma vez que o seu resto é apenas residual e pode ser descartado, então $P_3(x)$ fatorado se torna:

$$P_3(x) = (2x^2 + 2.33074608612483x + 1.71618865899311) * (x - \alpha_1)$$

Logo, as duas raízes restantes de $P_3(x) = 0$ estão em:

$$P_2(x) = (2x^2 + 2.33074608612483x + 1.71618865899311) = 0$$

Esse polinômio de 2º grau resultante pode ser resolvido diretamente pela sua fórmula, determinando a segunda e a terceira raiz.

Podemos também reaplicar o método de Newton a esse polinômio de 2º grau para determinar todas as raízes pelo mesmo algoritmo.

Logo, as três raízes convergidas com 16 dígitos significativos da variável *double* são:

$$\alpha_1 = +1.165373043062415 + 0.0000000000000000i$$

$$\alpha_2 = -0.582686521531207 - 0.720118564628364i$$

$$\alpha_3 = -0.582686521531207 + 0.720118564628364i$$

E o polinômio final fatorado é $P_3(x) = a_1(x - \alpha_1)(x - \alpha_2)(x - \alpha_3)$ (com $a_1 = 2$).

Na próxima seção, vamos propor um algoritmo simples, robusto e eficiente para determinar raízes múltiplas (repetidas) de equações polinomiais, de modo que possa determinar simultaneamente as suas multiplicidades.

3.3.4.1 Determinação de Raízes de Equações Polinomiais com Multiplicidade M

Vamos primeiramente estabelecer a **Propriedade 13**, que determina a multiplicidade M de uma raiz α conhecida, e comparar com o número de divisões exatas.

Propriedade 13: uma raiz α de $P_n(x) = 0$ possui multiplicidade M se, e somente se, $\frac{d^i P_n}{dx^i}(\alpha) = 0, \forall i = 0, 1, \dots, M-1$ e $\frac{d^M P_n}{dx^M}(\alpha) \neq 0$.

No **Exemplo 3.23**, vamos mostrar a equivalência da **Propriedade 13** para determinar a multiplicidade M com o número M de divisões sucessivas exatas (com restos nulos).

Exemplo 3.23: determine a multiplicidade M da raiz $\alpha = 2$ de $x^4 - 5x^3 + 6x^2 + 4x - 8 = 0$.

Solução:

Vamos primeiro calcular $P_4(x=2)$ e suas derivadas analiticamente:

$$\begin{aligned} P_4(x) &= x^4 - 5x^3 + 6x^2 + 4x - 8 && \Rightarrow P_4(2) = 0 \\ P'_4(x) &= 4x^3 - 15x^2 + 12x + 4 && \Rightarrow P'_4(2) = 0 \\ P''_4(x) &= 12x^2 - 30x + 12 && \Rightarrow P''_4(2) = 0 \\ P'''_4(x) &= 24x - 30 && \Rightarrow P'''_4(2) = 18 \end{aligned}$$

Pela **Propriedade 13**, temos que a multiplicidade de $\alpha = 2$ é $M = 3$, pois gera até derivada 2ª nula, e a derivada 3ª é não nula. Se aplicarmos as divisões sucessivas de Briot-Ruffini, teremos 3 restos nulos e $R_4 = 3$, logo a multiplicidade de $\alpha = 2$ também é $M = 3$, pois tem três divisões exatas.

A seguir, vamos definir um algoritmo sequencial para tratamento de raízes múltiplas desenvolvido através de quatro etapas, cada uma apresentando um exemplo com parte da metodologia aplicada.

Primeira etapa: vamos tentar determinar uma raiz múltipla como se fosse simples, de multiplicidade $M = 1$, usando a eq. (25) do método de Newton tradicional, mas com um polinômio exemplo cuja raiz é de multiplicidade $M = 3$.

Exemplo 3.24: determine as três raízes de $P_3(x) = x^3 - 3x^2 + 3x - 1 = 0$ com 16 dígitos exatos, considerando multiplicidade $M = 1$ (observe que a sua única raiz é $\alpha_1 = 1.0$ e tem multiplicidade $M = 3$, pois $P_3(x) = (x - 1)^3$).

Solução:

Pelos cálculos de cotas-limite, temos que todas as raízes iniciais estão entre $R_{\min} = 0.259921049896187$ e $R_{\max} = 3.84732210184363$ e:

- a) uma raiz inicial Real, localizada por varredura, pode ser:
 $x_0 = 1.1$;
- b) duas possíveis raízes iniciais Complexas, randômicas, localizadas dentro do limite das cotas, podem ser:

$$+2.36284988456718 + 2.59235983373818i$$

$$+0.91142849471678 + 1.75484834399166i$$

Aplicando o método de Newton através das eqs. (25) e (11b), a partir de $x_0 = 1.1$, teremos o seguinte processo iterativo (em *double*):

Tabela 3.15 – Tabela de resultados do Exemplo 3.24

k	$P_3(x_k)$	$P'_3(x_k)$	$\Delta x = -\frac{R(1)}{R(2)}$	$x_{k+1} = x_k + \Delta x$	$ P_3(x_k) + \Delta x $
0	9.999999999998900e-04	3.000000000000020e-02	-3.3333333332940e-02	1.066666666666670	3.4333333332930e-02
1	2.962962962960170e-04	1.33333333333540e-02	-2.22222222219780e-02	1.044444444444470	2.251851851849380e-02
2	8.779149519977380e-05	5.925925925932770e-03	-1.481481481494470e-02	1.029629629629530	1.490260631014450e-02
3	2.601229487364520e-05	2.633744855948810e-03	-9.876543209905690e-03	1.019753086419620	9.902555504779330e-03
4	7.707346629359790e-06	1.170553269302930e-03	-6.584362140092560e-03	1.013168724279530	6.592069486721920e-03
5	2.283658260271440e-06	5.202458974511440e-04	-4.389574759666210e-03	1.008779149519860	4.391858417926480e-03
:					
21	7.993605777301130e-15	1.198849464145720e-09	-6.667731034102110e-06	1.000013322676890	6.667731042095720e-06
22	2.220446049250310e-15	5.324816143570390e-10	-4.169995713244410e-06	1.000009152681170	4.169995715464850e-06
23	0.000000000000000e+00	2.513149688354590e-10	0.000000000000000e+00	1.000009152681170	0.000000000000000e+00

Fonte: Elaboração própria

Em $k=23$, observe que o critério de parada composto $|P_3(x)| + |\Delta x|$ é zerado, mas a raiz aproximada $x_{23} = 1.000009152681170$ ainda não atingiu seu valor exato. Sabemos que a raiz exata é 1.0, mas a raiz convergida x_{23} tem apenas 6 dígitos significativos exatos e o processo iterativo estagnou, pois Δx já é nulo e não vai mais alterar x nos próximos passos iterativos. Então, a primeira raiz atingida é $\alpha_1 = 1.000009152681170$, considerando multiplicidade $M = 1$.

Fazendo a redução de grau com essa primeira raiz aproximada, temos:

$$n = 2 \Rightarrow P_2(x) = 1.0x^2 - 1.99999084731883x + 0.999990847402598$$

E as demais raízes são:

$$\alpha_2 = 0.999995424 + 0.000007926i$$

$$\alpha_3 = 0.999995424 - 0.000007926i$$

Assim, as três raízes convergidas são diferentes entre si e com apenas 6 dígitos significativos exatos, ou seja, deveríamos ter obtido três raízes $\alpha_1 = 1.0$ com 16 dígitos exatos, ou seja, multiplicidade $M = 3$.

Segunda etapa: vamos agora avaliar os efeitos de uma raiz múltipla sobre os restos das divisões sucessivas de $P_n(x) = 0$ por $(x - \alpha)$.

Exemplo 3.25: determine a multiplicidade M da raiz $\alpha = 1.0$ de $P_3(x) = x^3 - 3x^2 + 3x - 1 = 0$.

Solução:

Pela divisão sintética de Briot-Ruffini, temos:

	α	a_1	a_2	a_3	a_4
1ª Divisão	1	1	-3	3	-1
			1	-2	1
2ª Divisão	1	1	-2	1	0
			1	-1	
3ª Divisão	1	1	-1	0	
			1		
4ª Divisão	1	1	0		

$$\boxed{1} \Rightarrow R_4 \text{ (não podemos efetuar a divisão de 1 por } (x-1))$$

ou usando as divisões tradicionais, temos:

$$\begin{array}{c}
 x^3 - 3x^2 + 3x - 1 \quad | \quad x - 1 \\
 \hline
 R_1 = 0 \quad 1x^2 - 2x + 1 \quad | \quad x - 1 \\
 \hline
 R_2 = 0 \quad x - 1 \quad | \quad x - 1 \\
 \hline
 R_3 = 0 \quad 1 \quad | \quad x - 1 \\
 \hline
 R_4 = 1 \quad 0
 \end{array}$$

Na quarta divisão, $P_0(x) = 1$ deve ser dividido por $(x-1)$, gerando quociente nulo e resto 1, pois $P_0(x) = 0 * (x-1) + 1$.

Então, determinamos três restos nulos, $R_1 = R_2 = R_3 = 0$ e um não nulo $R_4 = 1$, o que indica multiplicidade $M = 3$ (três divisões exatas).

Pela **Propriedade 12**, teremos as seguintes derivadas:

$$P_3(1) = 0! * R_1 = 1 * 0 = 0$$

$$P'_3(1) = 1! * R_2 = 1 * 0 = 0$$

$$P''_3(1) = 2! * R_3 = 2 * 0 = 0$$

$$P'''_3(1) = 3! * R_4 = 6 * 1 = 6$$

E pela **Propriedade 13**, a multiplicidade também é $M = 3$, pois temos até 2^a derivada nula e derivada 3^a não nula, resultando no polinômio fatorado equivalente:

$$P_3(x) = x^3 - 3x^2 + 3x - 1 = (x - 1)^3 = 0$$

Observe que, para definir a multiplicidade M de uma raiz α , precisamos conhecer o valor dessa raiz, ou seja, um resultado depende do outro.

Na próxima etapa, vamos assumir a hipótese de que apenas a multiplicidade M de uma raiz α seja previamente conhecida.

Terceira etapa: vamos assumir que a multiplicidade M é conhecida antes de conhecer a raiz correspondente.

O método de Newton para raízes não repetidas, ou seja multiplicidade $M = 1$, dado pela eq. (25), prevê que:

$$\Delta x = -\frac{P_n(x_k)}{P'_n(x_k)} = -\frac{R_1}{R_2}$$

Esse é o método de Newton clássico, mas observe que, no caso de aproximações de raízes com multiplicidade maior do que 1, R_1 e R_2 tendem a zero juntos, reduzindo a taxa de convergência (perde a taxa de convergência quadrática, como vimos nos **Exemplos 3.10** e **3.24**), podendo até gerar uma indeterminação 0/0 no limite de convergência. Então,

a) Se $\Delta x = \lim_{x \rightarrow \alpha} \left(-\frac{P_n(x)}{P'_n(x)} \right) = 0$ (para raízes não repetidas)

$$\Delta x = -\frac{P_n(x_k)}{P'_n(x_k)} = -\frac{0!R_1}{1!R_2} \quad \text{para } M = 1$$

b) Se $\Delta x = \lim_{x \rightarrow \alpha} \left(-\frac{P_n(x)}{P'_n(x)} \right) = \frac{0}{0} = ?$ (indeterminação) à medida

que x_k tende à raiz α , então poderemos aplicar a regra de L'Hospital, como vimos na seção 3.2.2.4, uma vez que:

$$\lim_{x \rightarrow \alpha} \left(\frac{P_n(x)}{P'_n(x)} \right) = \lim_{x \rightarrow \alpha} \left(\frac{P'_n(x)}{P''_n(x)} \right), \text{ então}$$

$$\Delta x = -\frac{P'_n(x_k)}{P''_n(x_k)} = -\frac{1!R_2}{2!R_3} \text{ para } M=2$$

c) Se $\Delta x = \lim_{x \rightarrow \alpha} \left(\frac{P'_n(x)}{P''_n(x)} \right) = \frac{0}{0} = ?$ à medida que x_k tende à raiz α ,

então deveremos reaplicar a regra de L'Hospital,

$$\lim_{x \rightarrow \alpha} \left(\frac{P'_n(x)}{P''_n(x)} \right) = \lim_{x \rightarrow \alpha} \left(\frac{P''_n(x)}{P'''_n(x)} \right), \text{ então}$$

$$\Delta x = -\frac{P''_n(x_k)}{P'''_n(x_k)} = -\frac{2!R_3}{3!R_4} \text{ para } M=3$$

d) Para M genérico, temos:

$$\Delta x = -\frac{P_n^{(M-1)}(x_k)}{P_n^{(M)}(x_k)} = -\frac{(M-1)!R_M}{M!R_{M+1}} = -\frac{(M-1)!R_M}{M(M-1)!R_{M+1}}$$

$$\boxed{\Delta x = -\frac{R_M}{M * R_{M+1}}} \text{ para } M \text{ qualquer}$$

(26)

Com $x_{k+1} = x_k + \Delta x$ dado pela mesma eq. (11b).

Exemplo 3.26: determine a raiz de $P_3(x) = x^3 - 3x^2 + 3x - 1 = 0$ com 16 dígitos exatos, sabendo que a multiplicidade dessa raiz é $M = 3$.

Solução:

Vamos partir do mesmo valor inicial $x_0 = 1.1$, imprimir os 4 restos das divisões sucessivas de $P_3(x)$ por $(x - x_0)$, usar $M = 3$ e aplicar a eq. (26):

$$\Delta x = -\frac{R_3}{3 * R_{3+1}}$$

Tabela 3.16 – Tabela de resultados do Exemplo 3.26

k	x_k	R_1	R_2	R_3	R_4	M	Δx (eq.26)	x_{k+1}	$ P_3(x_k) + \Delta x $
0	1.1	0.0010000	0.0300000	0.3000000	1.0	3	-0.1	1.0000	0.101
1	1.0	0	0	0	1.0	3	0.0	1.0000	0.0
2	1.0								

Fonte: Elaboração própria

Observe que obtemos a raiz exata $\alpha_1 = 1.0$ em duas iterações: na primeira, já calculamos a raiz exata; e, na segunda iteração, confirmamos essa raiz e sua multiplicidade $M_1 = 3$ gerando 3 restos nulos, mas para tal tivemos que usar um valor de multiplicidade previamente conhecido. Nesse exemplo de raiz única, se usarmos qualquer outro valor inicial da raiz, teremos o mesmo resultado convergido em 2 iterações. Perceba a dificuldade adicional de aplicar essa correção no método de Newton, dada pela eq. (26), que é a necessidade de determinar previamente o valor da multiplicidade M de uma raiz por meio de sua aproximação x_k .

Segundo o trabalho de Schroder (1992), a eq. (18) representa uma estimativa clássica do valor da multiplicidade \hat{M} que, adaptada com os restos dos polinômios representativos de f, f' e f'' , resulta em,

$$\hat{M} = \frac{(f'(x_k))^2}{(f'(x_k))^2 - f(x_k)f''(x_k)} = \frac{(R_2)^2}{(R_2)^2 - R_1 * (2 * R_3)} \quad (27)$$

Mas essa estimativa depende de vários fatores, como da proximidade da raiz x_k com a raiz exata α e do número de raízes múltiplas presentes no polinômio.

Para polinômios com única raiz múltipla, a eq. (27) gera a multiplicidade correta, independentemente do valor inicial da raiz, por exemplo:

a) para $P_3(x) = (x-1)^3$, em $x_0 = +1.1$, teremos
 $\hat{M} = 3.000000000001910$; e

b) para $P_3(x) = (x-1)^3$, em $x_0 = -110$, teremos
 $\hat{M} = 3.000000000000000$.

Para polinômios com duas ou mais raízes múltiplas, a estimativa dada pela eq. (27) é boa somente para valores de raízes nas proximidades da raiz exata, como:

a) Para $P_8(x) = (x+1)^2(x-0.9)^6$, em $x_0 = -1.001$, teremos
 $\hat{M} = 2.006315782479700$;

b) Para $P_8(x) = (x+1)^2(x-0.9)^6$, em $x_0 = +0.91$, teremos
 $\hat{M} = 6.024375198569969$; e

c) Para $P_8(x) = (x+1)^2(x-0.9)^6$, em $x_0 = +110$, teremos
 $\hat{M} = 7.999556745299021$ (\hat{M} inconsistente).

Então, para raízes Complexas ou relativamente afastadas da raiz exata, as estimativas de \hat{M} dadas pela eq. (27) falham e podem conduzir a resultados inconsistentes. Assim, precisamos de uma forma alternativa para estimar a multiplicidade \hat{M} de raízes genéricas.

Quarta etapa: vamos agora avaliar o comportamento dos restos com a aproximação de x_k com a raiz exata α .

Vamos propor neste livro uma alternativa prática para determinar a estimativa da multiplicidade \hat{M} de cada raiz x_k através de um contador do número \hat{M} de restos próximos de zero: R_1 , R_2 , $R_3, \dots, R_{\hat{M}}$, ou seja, do número de restos cujos valores em módulo fiquem abaixo de um limite mínimo estabelecido por experimentos numéricos, definido aqui pelo parâmetro R_{lim} no algoritmo proposto no **Caderno de Algoritmos**.

Quando x_k estiver convergido para a raiz α , teremos \hat{M} restos convergidos para valores “numericamente nulos (residuais)”. Então, por esse algoritmo, uma raiz aproximada x_k já é considerada previamente de multiplicidade \hat{M} , quando \hat{M} restos, ou quando a soma dos módulos desses restos, $R_1, R_2, R_3, \dots, R_{\hat{M}}$, estiverem abaixo do resto limite R_{lim} .

Segundo uma sequência de experimentos numéricos, verificamos que esse limite R_{\lim} depende basicamente da distância entre as raízes, mas que ainda não foram determinadas.

Então, considerando os coeficientes normalizados de modo que $a_1 = 1$, o limite R_{\lim} pode ser o produto $p = R_{\lim1} * R_{\lim2'}$ em que:

- a) $R_{\lim1} = 10\%$ do valor mínimo das diferenças entre todos os coeficientes em módulo, normalizado pelo maior dos coeficientes em módulo. Se essa diferença mínima for nula, adotaremos $R_{\lim1} = 0.1$; e
- b) $R_{\lim2}$ é o inverso da potência de 10 elevada ao número máximo de dígitos “fracionários” dos coeficientes (esse valor mede a influência direta da precisão das raízes).

Esse produto p deverá ser limitado a um valor mínimo para não gerar R_{\lim} nulo, então vamos adotar um mínimo 10^{-8} para considerar restos nulos sempre que ficarem menores que 10^{-8} (equivale à metade da precisão adotada, 8 dígitos significativos, para algoritmos construídos com variáveis *double* de 16 dígitos significativos), resultando na equação:

$$R_{\lim} = \max(p, 10^{-8}) \quad (28)$$

O parâmetro R_{\lim} é dependente da distância entre as raízes, do grau do polinômio e da precisão das variáveis utilizadas para o processamento numérico.

Esse parâmetro R_{\lim} captura bem raízes múltiplas espaçadas e ao mesmo tempo relativamente próximas entre si. Estas últimas precisam convergir até restos menores, por exemplo, da ordem de 10^{-8} , para serem consideradas numericamente nulas, conforme os exemplos a seguir.

Exemplo 3.27: determine as três raízes de $x^3 - 3x^2 + 3x - 1 = 0$ com 16 dígitos exatos e suas respectivas multiplicidades M (sabemos que a raiz é $\alpha = 1.0$ e a sua multiplicidade é $M = 3$).

Solução:

Tomamos $x_0 = 1.1$, conforme o **Exemplo 3.24**, calculamos o limite para os restos pela eq. (28), com:

- a) $R_{\text{lim}_1} = 0.1$ (pois a mínima diferença entre os 4 diferentes coeficientes é zero); e
- b) $R_{\text{lim}_2} = 1/10^0$ (pois o número máximo de dígitos “fracionários” dos coeficientes $\{1, -3, +3, -1\}$ é zero, uma vez que são coeficientes inteiros, sem frações decimais).

Aplicando a eq. (28), teremos o seguinte resto limite:

$$R_{\text{lim}} = \max(R_{\text{lim}_1} * R_{\text{lim}_2}, 10^{-8}) = \max(0.1 * 1, 10^{-8}) = 0.1$$

Aplicando o método de Newton modificado com a eq. (26) e estimando a multiplicidade \hat{M} a cada iteração pelo número \hat{M} de restos próximos de zero, temos:

Tabela 3.17 – Tabela de resultados do Exemplo 3.27

k	x_k	R_1	R_2	R_3	R_4	\hat{M}	Δx (eq.26)	x_{k+1}	$ P_3(x_k) + \Delta x $
0	1.100	0.0010000	0.0300000	0.3000000	1.0	2	-0.05	1.050	0.0510
1	1.050	1.2500e-04	7.5000e-03	1.5000e-01	1.0	2	-2.5e-02	1.025	2.5125e-02
2	1.025	1.5625e-05	1.8750e-03	7.5000e-02	1.0	3	-2.5e-02	1.000	2.5016e-02
3	1.000	0	0	0	1.0	3	0.0	1.000	0
4	1.000								

Fonte: Elaboração própria

Em $k = 0$ e $k = 1$, R_1 e R_2 estão abaixo de $R_{\text{lim}} = 0.1$ e a sua soma também é menor do que $R_{\text{lim}} = 0.1$ (destacados em negrito), ou seja, 2 restos são considerados suficientemente pequenos, logo a multiplicidade estimada é $\hat{M} = 2$.

Em $k = 2$, R_1 , R_2 e R_3 estão abaixo de $R_{\text{lim}} = 0.1$ e a sua soma mantém-se menor do que $R_{\text{lim}} = 0.1$ (destacados em negrito),

ou seja, 3 restos são considerados suficientemente pequenos, logo a multiplicidade estimada é $\hat{M} = 3$.

Em $k = 3$, quarta iteração, temos a confirmação da raiz convergida para 1.0 (exata) e da sua multiplicidade $M = 3$, pois 3 restos, nesse caso, são exatamente zero.

Então, os resultados desse algoritmo de Newton modificado para raízes múltiplas são exatos, $x_1 = 1.0$, e $M_1 = 3$, conforme o **Exemplo 3.27**, enquanto os resultados obtidos com multiplicidade $M = 1$ geram resultados com apenas 6 dígitos significativos exatos (análogos aos resultados da função roots() do Octave), conforme o **Exemplo 3.24**.

Exemplo 3.28: determine as raízes de

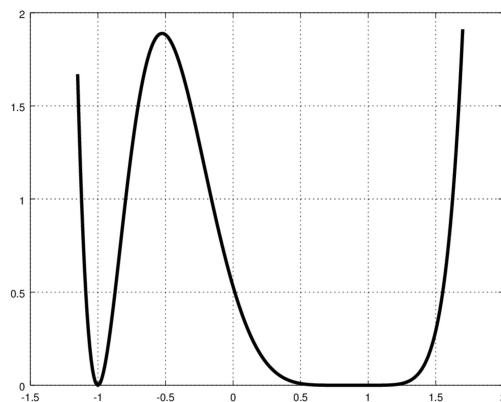
$$P_8(x) = 1x^8 - 3.4x^7 + 2.35x^6 + 4.32x^5 - 7.1685x^4 + 1.56006x^3 + 3.287061x^2 - 2.480058x + 0.531441 = 0$$

com 16 dígitos exatos e suas respectivas multiplicidades (polinômio fatorado equivalente: $P_8(x) = (x+1)^2 * (x-0.9)^6 = 0$).

Solução:

Nenhuma raiz Real foi localizada pelo teorema de Bolzano, pois ambas as raízes são de multiplicidade par e $P_8(x)$ não produz trocas de sinais, conforme o Gráfico 3.13.

Gráfico 3.13 – Representação da função $P_8(x)$



Fonte: Elaboração própria

As raízes iniciais complexas são geradas randomicamente pela função $fLocaliza(n,a)$ apresentada no **Caderno de Algoritmos**. O limite para os restos é determinado por:

- a) $R_{\text{lim}_1} = 0.00157548999093255$, pois a mínima diferença entre todos os coeficientes ocorre entre $|-3.4|$ e $|+3.287061|$, que foi normalizada pelo maior coeficiente em módulo $|-7.1685|$; e
- b) $R_{\text{lim}_2} = 1/10^6$, pois o número máximo de dígitos “fracionários” dos coeficientes de $P_8(x)$ é 6 (são coeficientes com no máximo 6 dígitos depois do ponto (vírgula)).

Aplicando a eq. (28), teremos o seguinte resto limite:

$$R_{\text{lim}} = \max(R_{\text{lim}_1} * R_{\text{lim}_2}, 10^{-8}) = \max(0.00157548999093255 * 10^{-6}, 10^{-8}) = 10^{-8}$$

Note que, nesse exemplo, os restos só serão considerados suficientemente pequenos quando ficarem abaixo de $R_{\text{lim}} = 10^{-8}$, ou seja, quando a raiz aproximada estiver próxima da raiz convergida.

Assim, os resultados obtidos são duas raízes exatas com suas respectivas multiplicidades:

$$\begin{aligned}x_1 &= +0.900000000000000 + 0.000000000000000i \text{ e } M_1 = 6 \\x_2 &= -1.000000000000000 + 0.000000000000000i \text{ e } M_2 = 2\end{aligned}$$

enquanto os resultados obtidos com multiplicidade $M = 1$ (análogo à função $roots()$ do Octave) geram resultados com apenas 3 a 6 dígitos significativos exatos.

Para cada raiz de multiplicidade M , devem ser efetuadas M reduções de grau para então determinar as próximas raízes.

Nos exemplos testados, se a soma dos módulos dos restos, $R_1, R_2, R_3, \dots, R_{\hat{M}}$, for menor do que R_{lim} , consideraremos que

a multiplicidade estimada da raiz x_k será \hat{M} . Desse modo, podemos acelerar o processo de convergência, usando a eq. (26), por meio de sucessivas estimativas de \hat{M} , que devem ser confirmadas no final, na última iteração, quando todos os restos até R_M tornam-se valores residuais. Consequentemente, R_{M+1} será naturalmente diferente de zero, e a eq. (26) será aplicável até o limite da precisão digital das variáveis envolvidas, obtendo raízes com a precisão máxima possível para a variável utilizada, no caso *double*.

Quando as raízes estão próximas entre si, os restos para valores de x nessa região diminuem muito, como se as raízes próximas fossem uma única raiz múltipla, então precisamos de um valor R_{\lim} menor do que um R_{\lim} para raízes mais espaçadas, a fim de eliminar uma possível falsa multiplicidade. Nesses casos, temos de fazer o processo iterativo chegar mais próximo da raiz exata para então assumir que essa raiz aproximada tenha multiplicidade M .

São aspectos importantes sobre o comportamento dos restos:

- a) Em $P_3(x) = x^3 - 3x^2 + 3x - 1 = (x-1)^3$, temos três raízes iguais a 1.0, $M = 3$ e 3 restos convergidos são nulos. No processo iterativo, os restos se reduzem gradativamente, mas sempre em ordem crescente $R_1 < R_2 < R_3$, até o quarto resto R_4 , que é sempre igual ao primeiro coeficiente a_1 .
- b) No exemplo testado por Rall (1966) para o polinômio, $P_5(x) = (x-1+\varepsilon)(x-1)(x-1-\varepsilon)(x-2)(x-3)$ com $\varepsilon = 10^{-2}$ e 10^{-3} , temos três raízes próximas a 1.0 e todas com multiplicidade $M = 1$. Nesse caso, com $\varepsilon = 10^{-2}$ e para a primeira raiz encontrada, os restos convergem para R_1 da ordem de $O(10^{-15})$, R_2 na ordem de $O(10^{-3})$, R_3 na ordem de $O(10^{-2})$ e o quarto resto é da ordem da unidade. Esses valores de restos poderiam gerar uma falsa multiplicidade $M = 3$ se o R_{\lim} estabelecido fosse, por exemplo, da ordem de $O(10^{-2})$, mas aplicamos o R_{\lim} proposto pela eq. (28), que assumiu o

valor mínimo 10^{-8} , ou seja, considerou apenas o R_1 como resto nulo e manteve a multiplicidade correta $M = 1$. Se fosse o caso de uma multiplicidade $M = 3$ verdadeira, os restos desse polinômio decresceriam até ficarem todos menores do que $R_{\lim} = 10^{-8}$ e teríamos obtido $M = 3$.

c) Para outro polinômio com 5 raízes próximas como

$$P_{10}(x) = -x^{10} - 7x^9 + 20.95x^8 - 34.75x^7 + 34.5004x^6 - 20.5012x^5 + \\ + 6.7512x^4 - 0.9504x^3 - 0x^2 - 0x + 0$$

($P_{10}(x) = (x-0)^3(x-0.8)(x-0.9)(x-1.0)^3(x-1.1)(x-1.2)$), pelo método de Newton modificado para raízes múltiplas são geradas as seguintes raízes:

$$x_1 = 0 \Rightarrow M_1 = 3$$

$$x_2 = 0.8000000000143045 \Rightarrow M_2 = 1$$

$$x_3 = 0.8999999993704603 \Rightarrow M_3 = 1$$

$$x_4 = 1.0000000000113123 \Rightarrow M_4 = 3$$

$$x_5 = 1.1000000003988399 \Rightarrow M_5 = 1$$

$$x_6 = 1.2000000000078299 \Rightarrow M_6 = 1$$

(resultados com no mínimo 10 dígitos significativos exatos)

Enquanto pelo método de Newton tradicional teremos raízes múltiplas calculadas como raízes distintas e com apenas 4 dígitos significativos exatos:

$$\begin{aligned} & 1.19999999821068 + 0.000000000000000i \\ & 1.100000002521077 + 0.000000000000000i \\ & 1.000159100194941 + 0.000275675116168i \\ & 1.000159100194941 - 0.000275675116168i \\ & 0.999681795585784 + 0.000000000000000i \\ & 0.900000001769856 + 0.000000000000000i \\ & 0.79999999912334 + 0.000000000000000i \\ & 0.000000000000000 + 0.000000000000000i \\ & 0.000000000000000 + 0.000000000000000i \\ & 0.000000000000000 + 0.000000000000000i \end{aligned}$$

(resultados análogos aos obtidos pela função *roots()* do *Octave* ou pelo *Wolfram Alpha*).

d) Para polinômios de grau elevado e multiplicidade também elevada, o algoritmo de Newton modificado para raízes múltiplas proposto neste livro também foi eficiente. Por exemplo, para a equação polinomial $P_{20}(x)=(x-1)^{20}=0$ expandida, temos:

- i) uma única raiz $x_1 = 1.0$ com multiplicidade $M_1 = 20$ (20 raízes iguais a x_1); e
- ii) com algoritmo de Newton original, sem a correção de multiplicidade ou com a função *roots()* do *Octave*, temos 20 raízes distintas e imprecisas como as duas exemplificadas a seguir:

$$1.322055777843924 + 0.055758394623173i$$

$$0.754913929640771 + 0.000000000000000i$$

Por fim, obtivemos os parâmetros limite da eq. (28) por meio de diversos testes numéricos com diferentes polinômios, que devem ser novamente experimentados para especificidades de novas famílias de polinômios. Há também uma dependência natural das raízes em relação ao seu valor inicial, que foi minimizada por um algoritmo de localização precisa de raízes iniciais, conforme *fLocaliza(n,a)*.

Nos casos de raízes muito próximas entre si, é recomendada a utilização de técnicas como a determinação dos autovalores da matriz companheira usando o método QR, conforme Serre (2002).

Na próxima seção, vamos detalhar o processo de purificação das raízes determinadas nos polinômios de grau reduzido visando minimizar o efeito do acúmulo de erros de arredondamentos decorrentes dessas reduções de grau.

3.3.4.2 Purificação de Raízes Obtidas com Redução de Grau

Considere a equação polinomial:

$$P_4(x) = x^4 - 11.101x^3 + 11.1111x^2 - 1.0111x + 0.001 = 0$$

que tem as seguintes raízes exatas:

$$\alpha_1 = 0.001$$

$$\alpha_2 = 0.1$$

$$\alpha_3 = 1$$

$$\alpha_4 = 10$$

Considere que a quarta raiz aproximada pelo método de Newton seja $\alpha_4 = 10.000005$, com multiplicidade $M = 1$, e que tenha sido encontrada em $P_1(x)=0$ pelas reduções sucessivas de grau:

α_1 é obtida em $P_4(x)=0$

α_2 é obtida em $P_3(x)=0$

α_3 é obtida em $P_2(x)=0$

α_4 é obtida em $P_1(x)=0$

Então, α_4 contém erros devido ao método de Newton e devido às reduções sucessivas de grau, que implicam arredondamentos nos coeficientes dos sucessivos quocientes. Isso acumula erros de arredondamento e pode gerar instabilidade numérica.

Uma solução para minimizar esse problema pode ser o chamado processo de **purificação** dos resultados parciais, aplicado já a partir da segunda raiz α_2 , porque é a primeira raiz a ser obtida em um polinômio que já sofreu a redução de grau e arredondamentos nos seus coeficientes.

Assim, para purificar α_4 , por exemplo, reaplicamos o método de Newton, como se fôssemos obter a primeira raiz, usando:

- a) o polinômio original, $P_4(x)=0$, que é o único “puro”, sem erros de arredondamento nos seus coeficientes, pois não sofreu redução de grau; e

- b) o valor inicial como cada **raiz aproximada**, obtida nos polinômios de grau reduzido.

O resultado da purificação de $\alpha_4 = 10.000005$ em $P_4(x)=0$ (original) é a quarta raiz purificada $\alpha_4 = 10.00000000000002$, com precisão de 16 dígitos significativos (sublinhados).



Note que essas raízes aproximadas através do polinômio de grau reduzido já são próximas das raízes exatas, então podem ser usadas como valores iniciais no polinômio original diminuindo o risco de ocorrer convergência para uma raiz já obtida.

Antes de continuar a sua leitura, volte ao **Caderno de Algoritmos** e confira o algoritmo de Newton completo para localização, determinação e purificação de todas as raízes de equações polinomiais, Reais ou Complexas, de qualquer multiplicidade M , no arquivo **Cap3NewtonPolinomios.m**.

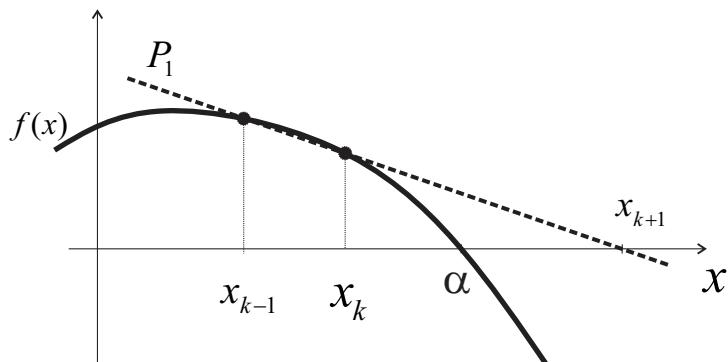
Considere que esse algoritmo de Newton completo:

- apresenta uma alternativa para determinação de raízes múltiplas de equações polinomiais, espaçadas e/ou próximas entre si, convergindo até ao limite da precisão digital disponível (no caso *double*);
- recupera a taxa quadrática de convergência otimizando o número total de operações aritméticas para obtenção de raízes múltiplas;
- tem princípios matemáticos simples; e
- utiliza parâmetros limite que podem ser estendidos a outros tipos de polinômios.

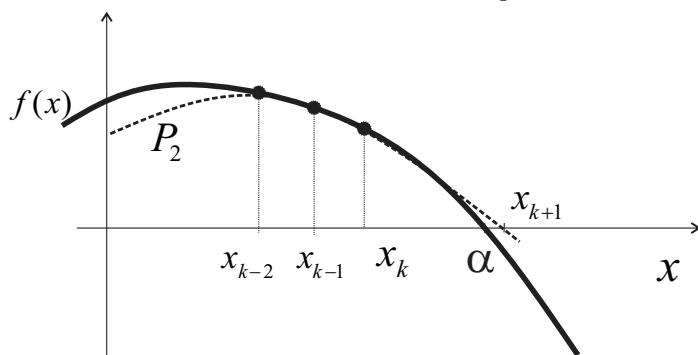
3.3.5 Determinação de Raízes de Equações pelo Método de Müller

O método de Muller para a determinação de raízes de equações é uma extensão do método da secante que busca minimizar alguns problemas presentes nesse método e no de Newton, como não encontrar diretamente raízes Complexas, a menos que seja dado um valor inicial Complexo e seja utilizada a aritmética complexa na sua aplicação.

Para apresentar o método de Müller (BURDEN; FAIRES, 2011), vamos analisar os Gráficos 3.14 e 3.15, que comparam a resolução de uma equação $f(x)=0$ pelos métodos da secante e de Müller.

Gráfico 3.14 – Método da secante (P_1 reta)

Fonte: Elaboração própria

Gráfico 3.15 – Método de Müller (P_2 parábola)

Fonte: Elaboração própria

No método da secante, calculamos o zero da **função aproximadora linear** que passa por dois pontos da curva de $f(x)=0$, conforme o Gráfico 3.14. Já no método de Müller, calculamos o zero da **função aproximadora quadrática** que passa por três pontos da curva de $f(x)=0$ a ser resolvida, conforme o Gráfico 3.15. Seria equivalente a estendermos a aproximação de 1^a ordem da série de Taylor para gerar o método de Newton, mas usando agora uma aproximação de 2^a ordem da série de Taylor, conforme propomos no **Exercício 3.3 do Caderno de Exercícios e Respostas** disponível no link <<http://sergiopeters.prof.ufsc.br/exercicios-e-respostas/>>.

Supondo que três aproximações iniciais, x_{k-2} , x_{k-1} e x_k , são estimadas na região da raiz de $f(x)=0$, a nova aproximação x_{k+2} seria obtida de um zero do polinômio quadrático $P_2(x)$ que passa por essas três aproximações iniciais de $f(x)$, logo

$$P_2(x) = a(x - x_k)^2 + b(x - x_k) + c \quad (29)$$

Aplicando as condições $P_2(x_{k-2}) = f(x_{k-2})$, $P_2(x_{k-1}) = f(x_{k-1})$ e $P_2(x_k) = f(x_k)$, obtemos o sistema de equações lineares:

$$\begin{cases} P_2(x_{k-2}) = a(x_{k-2} - x_k)^2 + b(x_{k-2} - x_k) + c = f(x_{k-2}) \\ P_2(x_{k-1}) = a(x_{k-1} - x_k)^2 + b(x_{k-1} - x_k) + c = f(x_{k-1}) \\ P_2(x_k) = a(x_k - x_k)^2 + b(x_k - x_k) + c = f(x_k) \end{cases} \quad (30)$$

que, resolvido diretamente, explicita os coeficientes a , b e c :

$$a = q * f(x_k) - q (1+q) f(x_{k-1}) + q^2 * f(x_{k-2}) \quad (31a)$$

$$b = (2q+1) f(x_k) - (1+q)^2 f(x_{k-1}) + q^2 * f(x_{k-2}) \quad (31b)$$

$$c = (1+q) f(x_k) \quad (31c)$$

onde

$$q = \frac{x_k - x_{k-1}}{x_{k-1} - x_{k-2}} \quad (32)$$

Agora, encontramos a nova aproximação x_{k+1} resolvendo a equação de 2º grau $P_2(x_{k+1}) = 0$:

$$x_{k+1} = x_k - (x_k - x_{k-1}) \left[\frac{2c}{b \pm \sqrt{b^2 - 4ac}} \right] \quad (33)$$

Note que, na eq. (33), temos duas possibilidades de obter x_{k+1} , a depender do sinal escolhido para o radicando, mas esse sinal deve ser escolhido de modo que o valor do módulo do denominador seja sempre o maior possível, isto é, o sinal para o radicando deve ser o mesmo sinal de b . Essa escolha gera um denominador maior em magnitude, evita possibilidades de denominador nulo, minimiza

arredondamentos e gera um valor de x_{k+1} mais próximo de x_k , dando mais estabilidade ao método numérico.

Assim, escolhemos a seguinte expressão para atualizar x_{k+1} ,

$$x_{k+1} = x_k - (x_k - x_{k-1}) \left[\frac{2c}{b + sign(b)\sqrt{b^2 - 4ac}} \right] \quad (34)$$

Então, aplicamos iterativamente a fórmula dada na eq. (34) até que um critério de parada seja satisfeito. Note que, se o radicando gerar números Complexos, podemos obter raízes Complexas de $f(x)=0$ diretamente a partir de valores iniciais Reais, o que é impossível de ocorrer com o método de Newton ou da secante.

Exemplo 3.29: determine a raiz de $x * ln(x) - 3.2 = 0$ pelo método de Müller com o máximo de dígitos significativos disponível.

Solução:

Escolhido o primeiro valor inicial $x_0 = 2.5$ para $k = 0$ em $[2, 3]$ e gerado outros dois valores anteriores, x_{k-2} e x_{k-1} , nessa região de proximidade da raiz, temos:

$$x_0 = 2.5$$

$$x_{k-1} = x_0 * 0.99 = 2.475$$

$$x_{k-2} = x_0 * 0.98 = 2.450$$

Tabela 3.18 – Tabela de resultados do Exemplo 3.29

k	x_{k-2}	x_{k-1}	x_k	x_{k+1}	$ x_{k+1} - x_k $
0	2.450	2.475	2.500	2.95286713407486	0.45557167
1	2.475	2.500	2.95286713407486	2.95416246266100	0.0013017
2	2.500	2.95286713407486	2.95416246266100	2.95416552326093	3.0606e-06
3	2.95286713407486	2.95416246266100	2.95416552326093	2.95416552327888	1.7952e-11
4	2.95416246266100	2.95416552326093	2.95416552327888	2.95416552327888	8.8818e-16
5	2.95416552326093	2.95416552327888	2.95416552327888	2.95416552327888	4.4409e-16

Fonte: Elaboração própria

$$x_6 = 2.95416552327888$$

Observe que encontramos a raiz x_6 em 6 iterações com 16 dígitos significativos de precisão, assim como no método da secante,

ao passo que com o método de Newton foram necessárias 5 iterações, para este mesmo exemplo.

Veremos agora um exemplo de polinômio com raízes Complexas.

Exemplo 3.30: determine uma raiz de $x^4 - 2x^3 + 6x^2 - 8x + 8 = 0$ com o máximo de dígitos significativos disponível pelo método de Müller (sabendo que as raízes são: $1+i$, $1-i$, $2i$ e $-2i$).

Solução:

Vamos escolher o primeiro valor inicial $x_k = 2.0$ para $k=0$, dentro das cotas-limite, e gerar os outros dois valores, x_{k-2} e x_{k-1} , nessa região de proximidade da raiz. Desse modo, temos:

$$x_k = 2.0$$

$$x_{k-1} = x_k * 0.99 = 1.98$$

$$x_{k-2} = x_k * 0.98 = 1.96$$

Tabela 3.19 – Tabela de resultados do Exemplo 3.30

k	x_{k-2}	x_{k-1}	x_k	x_{k+1}	$ x_{k+1} - x_k $
0	$1.96000 + 0.00000i$	$1.98000 + 0.00000i$	$2.00000 + 0.00000i$	$1.31997 + 0.66667i$	0.95231
1	$1.98000 + 0.00000i$	$2.00000 + 0.00000i$	$1.31997 + 0.66667i$	$1.13016 + 0.84954i$	0.26357
2	$2.00000 + 0.00000i$	$1.31997 + 0.66667i$	$1.13016 + 0.84954i$	$1.11082 + 1.04137i$	0.19281
3	$1.31997 + 0.66667i$	$1.13016 + 0.84954i$	$1.11082 + 1.04137i$	$0.99908 + 1.01479i$	0.11485
:					
24	$1.00000 + 1.00000i$	$1.00000 + 1.00000i$	$1.00000 + 1.00000i$	$1.00000 + 1.00000i$	0.00000

Fonte: Elaboração própria

$$x_{25} = 1.00000 + 1.00000i$$

Observe que na 1^a iteração, efetuada em $k=0$, já geramos uma 1^a aproximação x_{k+1} Complexa para a raiz e encontramos a raiz convergida em 25 iterações com 16 dígitos significativos de precisão.

Podemos determinar as demais raízes pelo processo de redução de grau, como no próximo exemplo.

Exemplo 3.31: determine uma raiz de $x^3 - 3x^2 + 3x - 1 = 0$ com o máximo de dígitos significativos disponível pelo método de Müller (sabendo que a raiz é 1.0 e tem multiplicidade $M=3$).

Solução:

Escolhemos um primeiro valor inicial $x_k = 2.0$ para $k=0$, dentro das cotas-limite, e geramos os outros dois valores, x_{k-2} e x_{k-1} , nessa região de proximidade da raiz:

$$2.0$$

$$x_{k-1} = x_k * 0.99 = 1.98$$

$$x_{k-2} = x_k * 0.98 = 1.96$$

Depois de 180 iterações, atingimos o critério de parada

$$|x_{k+1} - x_k| = 3.57470423575603e-16$$

e a raiz:

$$x = 1.00000266769999 + 5.00575188652281e-06i$$

com precisão de apenas 6 dígitos significativos, como acontece no método de Newton clássico sem a devida correção da multiplicidade.

Na sua forma original, o método de Müller não foi adequado para determinar raízes múltiplas, não atingindo resultados no limite da precisão digital disponível, como também ocorreu no método de Newton clássico.

Agora, confira o algoritmo de Müller no arquivo **Cap3Muller.m** no seu **Caderno de Algoritmos**.

Por fim, para aprofundar os estudos de cada Capítulo, disponibilizamos um **Caderno de Exercícios e Respostas** para *download* no *link* <<http://sergiopeters.prof.ufsc.br/exercicios-e-respostas/>>.

3.4 Conclusões

Neste Capítulo, cumprindo seu objetivo principal, tratamos da solução numérica de equações a uma variável $f(x)=0$ quaisquer. Conforme alertamos no início da abordagem desse problema, esta não é uma tarefa simples, especialmente devido à inexistência de um método ou mesmo de uma metodologia geral que seja efetiva em todas as equações.

Apresentamos três metodologias, que não são as únicas existentes, e analisamos detalhadamente o comportamento algébrico, gráfico, numérico e computacional de pelo menos dois métodos típicos de cada uma.

Também tratamos minuciosamente da questão da influência da multiplicidade nas soluções de equações polinomiais $P_n(x)=0$, disponibilizando uma alternativa para obter todas as raízes de equações polinomiais com a máxima precisão possível.

