UNIVERSIDADE FEDERAL DE SANTA CATARINA - UFSC
CENTRO TECNOLÓGICO - CTC
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA - INE

Cálculo Numérico em Computadores:

Capítulo 2

Resolução de Equações não lineares

Autores: Prof. Sérgio Peters
Acad. Andréa Vergara da Silva

e-mail: sergio.peters@ufsc.br

Florianópolis, 2013.

Capítulo 2 - Solução de Equações a uma Variável

2.0 - Preliminares

Nos modelos matemáticos de problemas de várias áreas (engenharia, economia, etc.) ocorre a necessidade de se resolver uma equação f(x) = 0.

<u>Definição 1</u>: Raiz de f(x) = 0 é todo $\alpha \in$ C (Complexos), tal que $f(\alpha) = 0$. Ou seja, um valor qualquer α é raiz de f(x) = 0, ou zero da função f(x) se zerar a função f(x).

Exemplos:
i) $x^3 - 2x^2 + 2x = 0 \Rightarrow \alpha_1 = 0; \alpha_2 = 1 + i ; \alpha_3 = 1 - i$
ii) $e^{3x^3} - 2 = 0 \Rightarrow 3x^3 \ln e = \ln 2 \Rightarrow x^3 = \frac{\ln 2}{3} \Rightarrow \alpha = \sqrt[3]{\frac{\ln 2}{3}}$
iii) sen (x) = 1 $\Rightarrow \alpha_k = 2k\pi + \pi/2$, com $k \in \aleph$
iv) $4.\cos(x) = e^x \Rightarrow \alpha = ?$
v) $e^{2x} = -3 \Rightarrow \nexists\, \alpha$

Note-se que nem sempre é possível explicitar a variável da equação, ou seja, nem sempre se consegue isolar a variável em um dos membros da equação para se obter uma raiz α.

Pelas equações do exemplo anterior, conclui-se que:

1ª) Uma equação f(x)=0 pode:
- não ter solução;
- ter uma única solução;
- ter uma quantidade finita de soluções;
- ter uma quantidade infinita de soluções;

2ª) A solução de f(x) = 0 pode ser muito difícil, ou até impossível, de ser obtida pela técnica de isolamento da variável;

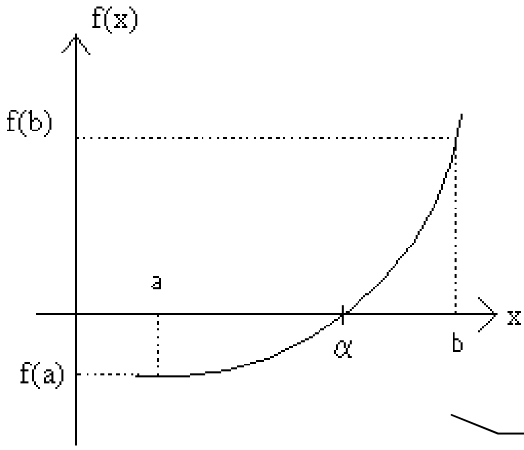
3ª) Resolver equações exige conhecimento de outras metodologias, além da forma de isolamento da variável. Aqui, usaremos a metodologia iterativa (proposta por Newton no século XVII) para a solução das equações.

<u>Definição 2</u>: Um método iterativo obedece sempre à duas etapas na sua execução:
1ª). Isolamento da solução desejada (aproximação inicial da solução);
2ª). Refinamento da solução isolada até a precisão requerida.

2.1 - Isolamento de Soluções de f(x) = 0

Para equações f(x) = 0 esta etapa é a mais difícil por ser a mais subjetiva . Ela pode ser efetivada através de:
1º). Conhecimento teórico sobre o problema modelado que gerou a equação.

2º). Fazer esboço gráfico de y = f(x).



Limita-se à(s) raíz(es) reais (α) pois a abcissa é definida em $\Re$.

3º) Utilizar propriedades algébricas da função geradora y = f(x). Dentre estas propriedades destacam-se:

Propriedade 1 (Teorema de Bolzano): Em f(x) = 0, se y=f(x) for contínua em [a,b] e $\rightarrow$
$f(a) * f(b) < 0 \Rightarrow \exists\ \alpha \in [a,b] \ / \ f(\alpha) = 0$.

4º) Agrupar as funções geradoras, y = f(x), em classes com características especiais.

Exemplo:
As funções geradoras do tipo $p_n(x) = a_1x^n + a_2x^{n-1} + ... + a_{n-1}$, com $\alpha_i \in \Re$, são $\rightarrow$denominadas polinômios de grau n e as equações polinomiais, $p_n\ (x) = 0$, possuem várias propriedades que facilitam o isolamento das soluções.

Propriedade 2: Toda equação polinomial de grau n possui n raízes reais ou complexas, distintas ou repetidas.

Propriedade 3: Para equações polinomiais de coeficientes reais tem-se que, se o complexo a + bi for raiz, seu conjugado a - bi também é raiz.

2.2 - Refinamento da solução isolada

Após isolada a raiz de f(x) = 0 ela será refinada através de técnicas específicas. Estas técnicas podem ser agrupadas em três grandes famílias de métodos:

1ª) Métodos de quebras: Obter $[a,b] \ / \ \alpha \in [a,b]$ e particioná-lo em subintervalos $[a_i,b_i]$ contendo α e, tais que, $\lim_{i \to \infty} |a_i - b_i| = 0$.

Estes métodos têm convergência garantida mas normalmente são lentos.

2^a) Métodos de linearização: Obtém-se uma aproximação inicial x^0 (valor estimado) para o α e posteriormente constrói-se uma seqüência iterativa $\{ x^i \}_{i=0}^{\infty} \;/\; \lim_{i \to \infty} x^i = \alpha$.

Para definir a seqüência iterativa, lineariza-se a equação não linear f(x) = 0, gerando uma equação linear recursiva $x^{i+1} = F(x^i)$.
Estes métodos não têm convergência garantida, mas, dependendo da forma recursiva estabelecida F(x), têm convergência rápida.

3^a) Métodos híbridos: consistem na mescla das duas metodologias, tentando associar as suas vantagens: convergência garantida e alta velocidade de convergência.

2.2.1 - Métodos de Quebra

2.2.1.1 - Método da Bisseção (ou Bipartição)

Em f(x) = 0 se f(x) for contínua em [a,b] e ocorrer f(a) * f(b) < 0, então uma raiz $\alpha \in$ [a,b] pode ser obtida como segue:

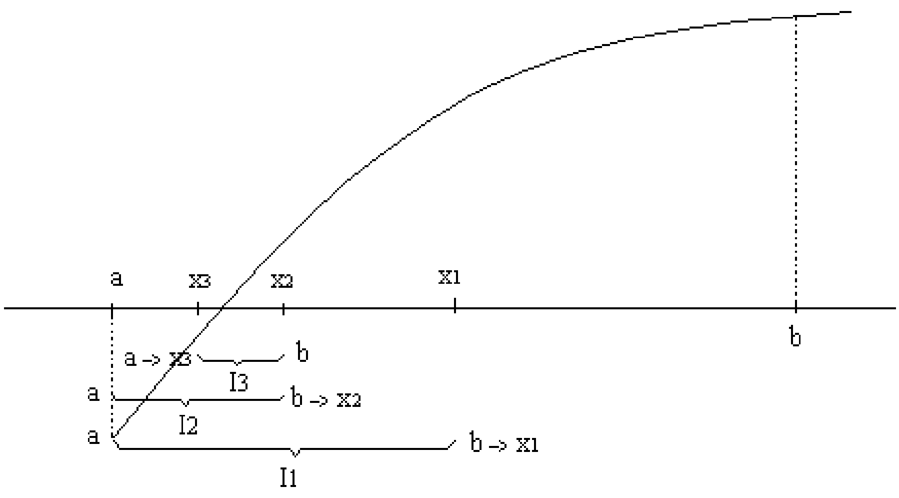
i) obter $x^1 = \dfrac{a + b}{2}$

ii) se $f(x^1) = 0 \Rightarrow x^1 = \alpha$

senão devemos verificar em qual subintervalo de [a,b] α está:

$$\begin{cases} \text{se } f(a)\,.\,f(x^1) < 0 \Rightarrow \alpha \in [a, x^1] \text{ e } b \leftarrow x^1 \\ \text{se } f(x^1)\,.\,f(b) < 0 \Rightarrow \alpha \in [x^1, b] \text{ e } a \leftarrow x^1 \\ \text{Redefinindo o novo intervalo } [a,b] \text{ retorna-se ao passo (i)} \end{cases}$$

No exemplo ilustrado abaixo $\alpha \in [a,x^1] \Rightarrow b = x^1$, ou seja, o valor de b é redefinido, para que α fique contido em um intervalo menor. Assim, repete-se o processo iterativo sucessivamente, avaliando x^k, com k = 2, 3, 4, ..., até que | a - b | < ε.



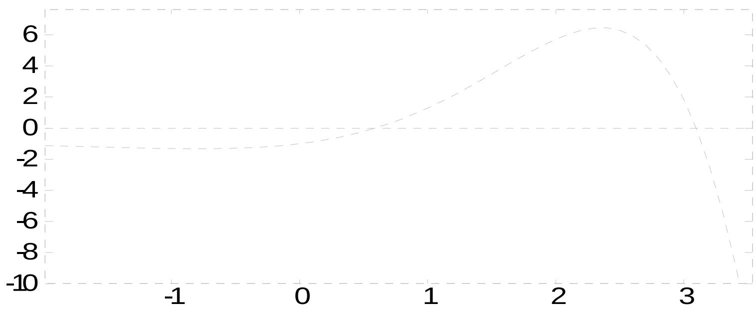
Neste ponto podemos levantar uma questão fundamental: será que $\lim\limits_{k \to \infty} x^k = \alpha$?

Note-se que o comprimento de cada novo subintervalo é $\dfrac{|b-a|}{2^k}$, onde *a* e *b* são os valores limites do intervalo inicial e *k* indica o nível iterativo, ou seja, número de bipartições.

Como: $\lim\limits_{k \to \infty} \dfrac{|b-a|}{2^k} = 0 \quad \Rightarrow \quad \lim\limits_{k \to \infty} x^k = \alpha$

Então, teoricamente, a raiz α será atingida quando o número *k* de bipartições tende ao infinito.

Exemplo:
1º) Obtenha um raiz α de e^x sen(x) = 1, situada em [0,1], por bisseção com 4 partições.



Temos que $f(x) = e^x \, sen(x) - 1$ é uma função contínua em toda reta Real e verifica-se que f(0) = -1 e f(1) = 1,287, como f(0) * f(1) < 0, $\exists\, \alpha \in [0,1]$, que pode ser comprovado pelo gráfico acima. Então partimos para o processo de refinamento. Como:

K	a	x_M	b	f(a)	f(x_M)	f(b)
0	0	0,5	1	-1	-0,21	1,287
1	0,5	0,75	1	-0,21	0,443	1,287
2	0,5	0,625	0,75	-0,21	0,093	0,443
3	0,5	0,5625	0,625	-0,21	-0,064	0,093
4	0,5625	0,594	0,625	-0,064	0,014	0,093

Logo a raiz $\alpha \cong 0{,}594$ com Erro=|a - b| = 0,062

Considerações:

i) Quando truncar o processo de bipartições para assegurar uma precisão ε estabelecida?
1º) Podemos predefinir a quantidade de partições através de:
O comprimento de cada subintervalo é dado por:

$$\frac{|b-a|}{2^k} \Rightarrow \frac{|b-a|}{2^k} \cong \varepsilon \Rightarrow k = \left(\frac{\ln\left(\frac{|b-a|}{\varepsilon}\right)}{\ln 2} \right) \qquad (1)$$

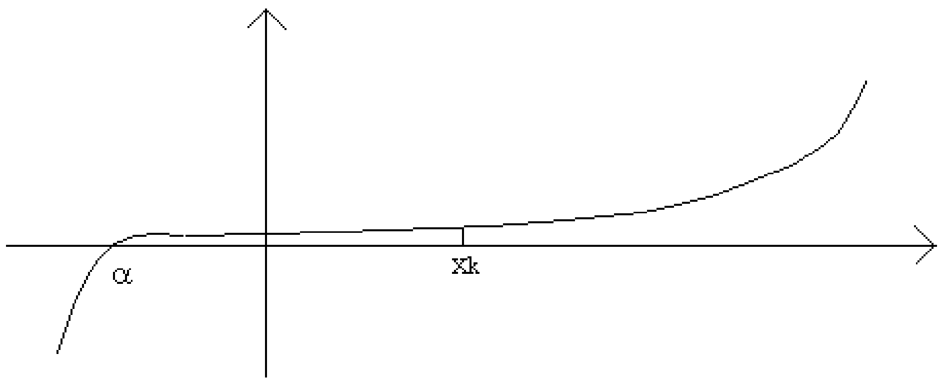
Então, após a k-ésima bipartição o comprimento do subintervalo deverá ser da ordem da precisão estabelecida ε**.**

Exemplo: Para $\sqrt{x} - 5.e^{-x} = 0$ obtenha o número de partições necessárias para aproximar $\alpha \in [1,2]$ com precisão $\varepsilon = 10^{-10}$.

Aplicando (1) $\Rightarrow k = \dfrac{\ln\left(\frac{1}{10^{-10}}\right)}{\ln 2} = 33{,}2 \quad \Rightarrow \quad k = 34$

2º) Ou podemos repetir as bipartições até que um critério de parada seja satisfeito.

a) $\left|f(x_{M_K})\right| < \varepsilon$



Observação: Note que apesar de x^k ainda estar longe da raiz α, o critério de parada estabelecido $|\ f(x^k)\ | < \varepsilon$ pode já ter sido satisfeito. Portanto, é um critério que deve ser estabelecido, mas não serve como decisão final, é necessário associar uma outra condição de parada, como a do item b).

b) $|\ b - a\ | < \varepsilon$

c) $|\ x^k - x^{k-1}| < \varepsilon$ ——— $|x^k - x^{k-1}|$ é pequeno mas $|f(x^k)|$ ainda é grande

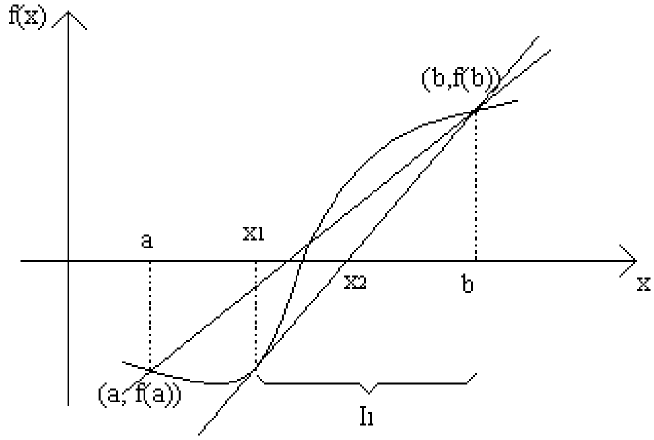
d) $\left|\dfrac{x^k - x^{k-1}}{x^k}\right| < \varepsilon$ ——— Valor relativo do critério de parada é mais realístico, principalmente quando x^k é muito pequeno ou muito grande

e) $\left|f(x^k)\right| + \left|\dfrac{x^k - x^{k-1}}{x^k}\right| < \varepsilon$ ——— É recomendável usar um critério combinado, que considere mais de um efeito.

I.2.1.2 - Método da Falsa Posição (ou das Cordas, ou Regula Falsi)

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

Como particionar o intervalo [a , b] de forma a obter uma convergência mais rápida?



O método da falsa posição consiste nas seguintes etapas:

i) Toma-se [a , b] com $\alpha \in$ [a , b];

ii) Obtém-se os pontos (a, f(a)) e (b,f(b));

iii) Define-se a reta r(x) que passa por estes pontos:

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

iv) O valor de x_k é obtido para $r(x_k) = 0$ (interseção de r(x) com o eixo das abcissas)

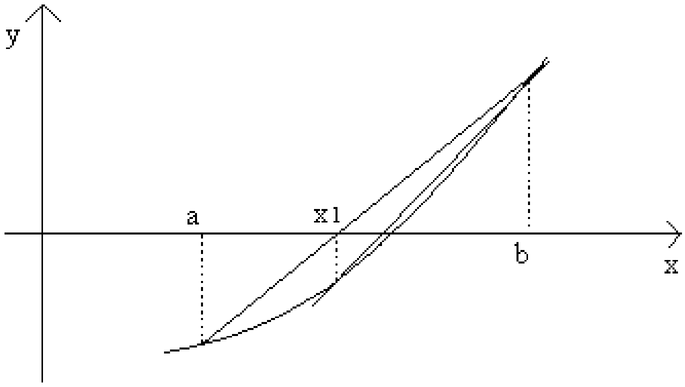
$$r(x_k) = 0 \Rightarrow x_K = a - \frac{f(a).(b-a)}{f(b) - f(a)} = \frac{a.f(b) - b.f(a)}{f(b) - f(a)}$$

v) Pela propriedade 1 decide-se onde ficou a raiz α, se em [a , x_k], ou se em [x_k , b].

vi) Redefine-se o intervalo e retorna-se ao item (ii) até se atingir a precisão desejada.

<u>Considerações</u>:

Os métodos de quebra podem apresentar, em algumas equações, a seguinte situação:



Note-se que uma das extremidades pode ficar fixa, sem atualização iterativa, atrasando o processo de convergência. Assim, é muito **importante alterar o critério de parada**, o critério de cálculo do erro, que não pode ser mais baseado na diferença entre 'a' e 'b', $|b-a|<\varepsilon$, e deve ser baseado no valor do zero x ou da f(x), $|x^k - x^{k-1}| < \varepsilon$ ou $\left|f(x^k)\right| < \varepsilon$.

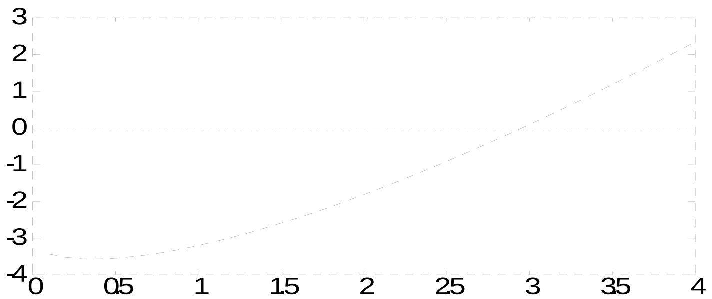
Podemos ainda corrigir a convergência unilateral, adotando-se o método de Pégaso, o método da falsa posição modificado ou os métodos de linearização, que veremos a seguir.

2.2.2 - Métodos de Linearização:

Nestes métodos busca-se linearizar a equação f(x) = 0 escolhendo apenas uma das incógnitas x como variável e todas a outras incógnitas são tratadas como se fossem conhecidas, inicialmente assumindo valores estimados x^*. Assim,

$$f(x) = 0 \Rightarrow x = F(x^*) \text{ ou } x_k = F(x_{k-1})$$

Exemplo: Resolver $x.\ln x - 3{,}2 = 0$ $\qquad \alpha \in [2,3]$



i) $x.\ln x^* - 3{,}2 = 0$

$$x = \frac{3{,}2}{\ln x^*}$$

Adotando $x^o = 3$

k	x^*	$x=3{,}2/\ln(x^*)$
0	3,000	2,913
1	2,913	2,993
2	2,993	2,919
3	2,919	...
...	...	...
53	2,954	2,954

A raiz obtida é 2,954 com Erro=$|x^k - x^{k-1}| < 10^{-3}$

ii)

$$x^* \ln x - 3{,}2 = 0$$

$$\ln x = \frac{3{,}2}{x^*}$$

$$x = e^{\frac{3{,}2}{x^*}}$$

k	x^*	$x = e^{\left(\frac{3{,}2}{x^*}\right)}$
0	3	2,906
1	2,906	3,008
2	3,008	2,897
3	2,897	3,17
...	...	...
...	∞	∞

Está divergindo.

Considerações:

1. Este processo iterativo é altamente sensível à escolha da forma iterativa $x = F(x^*)$, podendo gerar seqüências iterativas convergentes, seqüências oscilatórias ou seqüências divergentes. Em alguns casos podemos atenuar as oscilações usando fatores de amortecimento (sub-relaxação) no processo de atualização da variável x_k (vide capítulo 4).

2. Para se obter um resultado coerente e com exatidão, é necessário que a cada iteração a resposta se aproxime mais e mais da solução real, ou seja, que o método convirja para o valor real da raiz. No caso do método da bisseção, nós não precisamos nos preocupar com a convergência, pois nesse método ela está sempre garantida, já que isolamos a raiz dentro de um dado intervalo e nunca deixamos esse intervalo inicial. Já no método da iteração linear, a convergência não é garantida a priori. A cada iteração podemos nos aproximar ou nos afastar da solução real. Portanto, antes de resolver um problema através desse método é recomendável verificar se haverá ou não a convergência. O seguinte Teorema coloca condições suficientes, porém não necessárias para que o método de iteração linear seja convergente.

Teorema:

Seja uma função *f(x)* contínua em um intervalo [*a,b*] e α uma raiz de *f*(*x*) contida em [*a,b*]. Seja F*(x)* uma função de iteração obtida a partir de *f(x).*
Se:
i). F*(x) e* F*'(x)* forem contínuas em [*a,b*];
ii). |*F'(x*)| < 1 (para todo) $\forall\, x \in [a,b]$;
iii). $x^* \in [a,b]$.
Então: $\lim_{k\to\infty} x^k = \alpha$

Nesse caso, a sequência x^k converge para a raiz α de f(x)=0.

2.2.3.1) Método de Newton (Raphson)

Seja f(x) uma função contínua no intervalo [a,b] e seja α um zero contido neste intervalo, com derivadas f´(x) e f´´(x) contínuas, também em [a,b], mas com f´(x) ≠ 0. Podemos encontrar uma aproximação x_k para a raiz α de f(x)=0 no intervalo [a,b], utilizando a sua expansão em série de Taylor em torno de um valor inicial x^* *ou* x_{k-1} estimado.

Qualquer função f(x) pode ser representada, de forma exata, por série de Taylor e a equação f(x)=0 pode ser representada como segue, fazendo $x = x^* + \Delta x$:

$$f(x) = f(x^* + \Delta x) = \underbrace{f(x^*) + f'(x^*).\frac{\Delta x^1}{1!} + f''(x^*).\frac{\Delta x^2}{2!} + \ldots}_{\text{função original, com } f(x) \text{ representado em série de Taylor}}$$

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

Para resolvermos a equação representada em série de Taylor, teríamos que resolver uma equação polinomial em Δx com grau infinito, o que seria impossível.

Então Newton assumiu que uma região próxima da raiz α os valores de Δx são suficientemente pequenos para que se possa representar a função f(x) apenas pelos seus dois primeiros termos da série de Taylor, truncando os demais termos considerados de ordem inferior.

Assim,

$$f(x^* + \Delta x) \cong f(x^*) + f'(x^*).\Delta x = 0$$

Observa-se acima que a equação não linear f(x) = 0 foi substituída por uma equação linearizada aproximada, onde a única variável x está presente em $\Delta x = x - x^*$. Todas as incógnitas nos demais termos são assumidas como conhecidas, ou seja, inicialmente são valores estimados e são atualizados por um processo iterativo.

Então, resolvemos

$$f(x^*) + f'(x^*).\Delta x = 0$$

$$\Delta x = -\frac{f(x^*)}{f'(x^*)} \Rightarrow x = x^* + \Delta x$$

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

Aproximação de Newton para a raiz de f(x)=0, que será atingida quando Δx tende a zero.

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

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

Para melhorar a aproximação x da raiz de f(x) = 0, podemos adotá-la como um novo valor estimado x^* para uma segunda avaliação, ou seja, uma segunda iteração, e assim por diante:

$$x^{k+1} = x^k - \frac{f(x^k)}{f'(x^k)}$$

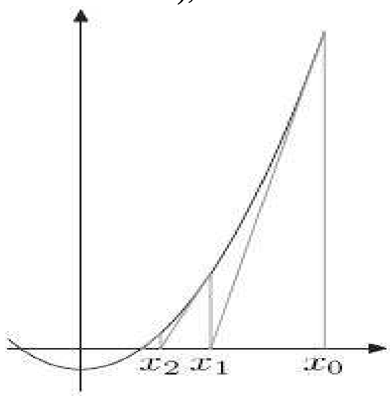
de modo que o erro de truncamento $O(\Delta x^2) = f''(x^*).\frac{\Delta x^2}{2!} + f'''(x^*).\frac{\Delta x^3}{3!} + \dots$ vai sendo reduzido a cada iteração.

<u>Características</u>:

- Intuitivamente nota-se que o método de Newton tem convergência rápida. Pode-se provar que a convergência deste método é quadrática, isto é, o erro da próxima iteração (k+1) é aproximadamente o quadrado do erro da próxima iteração k.

- Newton pode não convergir:

  i) se x^0 é inadequado (onde a reta tangente, representativa de f(x) em x^0, não corta o eixo x nas proximidades de x^0);



No exemplo acima, a reta tangente corta adequadamente o eixo x.

  ii) se pegar um ponto crítico (onde a derivada vale zero);
  iii) se ocorrer um *looping* .

Soluções para estes problemas:
i) escolher x^0 o mais próximo possível de α;

ii) para se evitar a divisão por zero e *loopings* pode-se utilizar um variante do método de Newton:

$$x^{k+1} = x^k - \frac{f(x^k)}{f'(x^k)}$$

fixando o denominador em $f'(x_0)$:

$$x^{k+1} = x^k - \frac{f(x^k)}{f'(x^0)} \quad \text{——— Newton modificado}$$

Pode ser muito difícil, ou até impossível, para o usuário obter a derivada explícita de f′(x). Como, então, refinar a raiz pelo método de Newton sem explicitar a f′(x)?

Por definição, temos que:

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

se *h* for pequeno

$$f'(x) \cong \frac{f(x+h) - f(x)}{h}$$

Podemos adotar, do lado direito da expressão de f'(x*), $x = x^*$, $x + h = x^* + h$ *e* $h = x - x^* = \Delta x^*$**, ou seja,** Δx^* **também é um valor estimado inicialmente:**

$$f'(x^*) \cong \frac{f(x^* + \Delta x^*) - f(x^*)}{\Delta x^*}$$ **onde** Δx^* **pode ser inicialmente estimado e depois recalculado a cada iteração, conforme a formula normal de Newton-Raphson:**

$$\Delta x = -\frac{f(x^*)}{f'(x^*)} \Rightarrow x = x^* + \Delta x$$

, onde x é novo valor da raiz aproximada de f(x)=0.

Alternativamente, pode-se adotar $f'(x^*)$ baseado em dois pontos anteriores x_{k-1} e x_k para depois substituir-se na expressão do método de Newton:

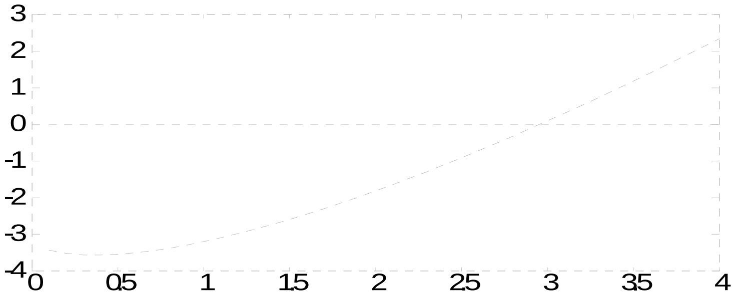
$$f'(x^*) = \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}} \quad \rightarrow \quad x^{k+1} = x^k - \frac{f(x^k)}{f'(x^k)}$$

$$x_{k+1} = x_k - \frac{f(x_k)}{\dfrac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}}$$

$$x_{k+1} = x_k - \frac{(x_k - x_{k-1}).f(x_k)}{f(x_k) - f(x_{k-1})} \quad \text{——— MÉTODO DA SECANTE}$$

$$x_{k+1} = \frac{x_{k-1}.f(x_k) - x_k.f(x_{k-1})}{f(x_k) - f(x_{k-1})}$$

Exemplo: Resolver com Newton-Raphson $x.\ln x - 3{,}2 = 0$ $\quad \alpha \in [2,3]$



$f(x^*) = x.\ln x - 3{,}2$

$f'(x^*) = \ln x + 1$

$\Delta x = -\dfrac{f(x^*)}{f'(x^*)} \quad \rightarrow \quad x = x^* + \Delta x$

Adotando $x^o = 3$ como valor inicial

k	x^*	$f(x^*) = x^*.\ln x^* - 3{,}2$	$f'(x^*) = \ln x^* + 1$	$x = x^* + \Delta x$	$Erro = \lvert \Delta x \rvert$
0	3	0.09584	2.0986	2.9543	0.04567
1	2.9543	3.4935e-004	2.0833	2.9542	1.6769e-004
2	2.9542	4.7594e-009			

A raiz obtida é 2,9542 com Erro=$|x^k - x^{k-1}| < 10^{-3}$, em apenas 2 cálculos (iterações).

- O que ocorre se a raiz α de f(x) = 0 tiver multiplicidade maior que um?

Tendo uma raiz α para f(x) = 0 $\Rightarrow$ Podemos reescrever f(x) da seguinte forma:

$$f(x) = (x - \alpha).g(x),$$

onde g(x) é o quociente da divisão de f(x) por (x - α) , que contém todas as outras raízes de f(x), exceto α. Derivando esta expressão para f(x) tem-se:

$$f'(x) = (x - \alpha).g'(x) + g(x).1$$

Se α anular também a função quociente g(x) $\Rightarrow$ g(x) = 0 (α é raiz de g(x)), então, f′(x) $\Rightarrow$ f′(α) = 0. Pois $f'(\alpha) = \underbrace{\cancel{(\alpha - \alpha)}}_{0}.\, g'(x) + \underbrace{\cancel{g(x)}}_{0} \Rightarrow f'(\alpha) = 0$.

Logo, com raízes repetidas a fórmula do Método de Newton apresenta f(x) $\rightarrow$ 0 e f'(x) $\rightarrow$ 0 também.

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \begin{array}{l} \rightarrow\ 0 \\ \rightarrow\ 0 \end{array}$$

O que torna a convergência lenta.

Exemplo: Obtenha por Newton uma raiz de: $e^x - x - 1 = 0$ com $x^0 = 1$.

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

f′(x) = e^x - 1

Aplicando a fórmula do método de Newton vem:
$x_0 = 1$
$x_1 = 0{,}58198$
$x_2 = 0{,}31906$
.
.
.
$x_4 = 0{,}08635$
.
.
.
$x_7 = 0{,}01107$
$x_8 = 0{,}005545$
.
.
.
$x_{11} = 6{,}9 \text{ x } 10^{-4}$
.
.
.
$x_{13} = 1{,}7416 \text{ x } 10^{-4}$
$x_{14} = 8{,}8041 \text{ x } 10^{-5}$ --> 0

O processo se tornou muito lento, perdeu a convergência quadrática.

Solução dada a este problema:
Dado f(x) = 0, pode-se tomar uma função g(x) modificada a partir de f(x):

$$g(x) = \frac{f(x)}{f'(x)}$$

Agora, aplica-se à função modificada g(x) pois toda raiz α que anula f(x) também anula g(x)

$$\Rightarrow \quad x_{k+1} = x_k - \frac{g(x_k)}{g'(x_k)} = x_k - \frac{\dfrac{f(x_k)}{f'(x_k)}}{\dfrac{[f'(x_k)]^2 - f(x_k).f''(x_k)}{[f'(x_k)]^2}}$$

$$x_{k-1} = x_k - \frac{f(x_k).f'(x_k)}{[f'(x_k)]^2 - f(x_k).f''(x_k)}$$

Convergirá mais rápido em casos de raízes múltiplas.

Ou aplicamos a **Regra de L'Hospital**: Sejam *f* e *g* funções deriváveis num intervalo aberto *I*, exceto possivelmente, em um ponto $a \in I$. Suponhamos que $g'(x) \neq 0$ para todo $x \neq a$ em *I*.

(i) Se $\lim_{x\to a} f(x) = \lim_{x\to a} g(x) = 0$, então

$$\lim_{x\to a}\left(\frac{f(x)}{g(x)}\right) = \lim_{x\to a}\left(\frac{f'(x)}{g'(x)}\right) = L$$

Logo,

$$\lim_{x\to a}\left(\frac{f(x)}{f'(x)}\right) = \lim_{x\to a}\left(\frac{f'(x)}{f''(x)}\right) = L$$, para o caso de $f(x) \to 0 \quad e \quad f'(x) \to 0$, resultando

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

Exemplo: Reaplicando o Método de Newton, modificado pela Regra de L'Hospital, para obter uma raiz de: $e^x - x - 1 = 0$ com $x^0 = 1$.
f '(x) = e^x - 1
f ''(x) = e^x
Aplicando a fórmula do método de Newton vem:
x_0 = 1
x_1= 0.367879
x_2= 0.0600801 --> O processo recuperou a rápida convergência quadrática.
x_3= 1.7692e-003
x_4= 1.5641e-006 --> 0

- Newton pode ser utilizado para se elaborar rotinas de funções pré-definidas para as máquinas digitais.

Exemplo: Elabore uma rotina para obter o recíproco ($1/c$) de c $\in \Re$ - {0}

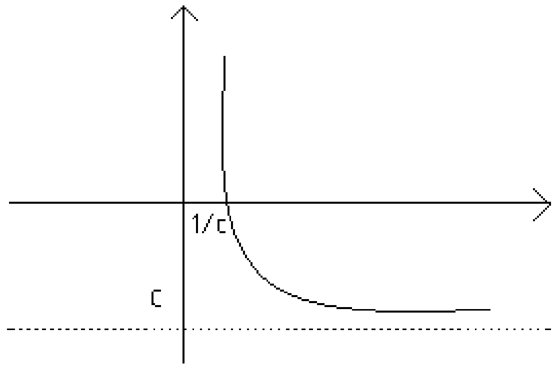
Deseja-se um x tal que x = $1/c \Rightarrow 1/x$ = c $\Rightarrow$ que pode ser escrita conforme a equação:
$f(x) = \frac{1}{x} - c$. Aplicando Newton com f '(x) = $-\frac{1}{x^2}$.

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

$$\Rightarrow \boxed{x_{k+1} = (2 - c.x_k).x_k}$$

Será que esta expressão acima sempre converge?

Gráfico da função geradora:



Devemos estimar o valor inicial x_k o mais próximo de $1/c$ para, assim, garantirmos a convergência. Se c = 0,... x $10^E \Rightarrow 1/c = \mathcal{O}(10^{-E})$

Exemplos:

1) Obtenha o recíproco de c = 425,32 e c = 0,42532 x 10^3 sem utilizar a operação de divisão.

Resposta: tomando $x_0 = 0.001\ (10^{-3}) \Rightarrow x_6 = 0{,}00235117$

2) Crie uma rotina para obter $\sqrt{c}$, $c \in \Re^+$.

Solução: Deseja-se x tal que $x = \sqrt{c} \Rightarrow x^2 - c = 0$

Aplicar Newton: $f(x) = x^2 - c$

$f`(x) = 2x$ $\qquad x_{k+1} = x_k + \Delta x = x_k + \left(-\dfrac{f(x_k)}{f'(x_k)} \right)$

$$x_{k+1} = x_k - \frac{(x_k^2 - c)}{2.x_k} \Rightarrow (x_k + c/x_k).0{,}5$$ -> Sempre converge para todo $x_0 > 0$.

Como estimar um x_0 adequado?

Hero sugere que:

$$x_0 = (1{,}68 - \frac{1{,}29}{0{,}84 + M}).10^p.3{,}16^q$$

Mantissa

Para $c = 0{,}d_1 \ldots d_t * 10^{2p+q}$, com q = 0 ou q =1 de modo que p seja inteiro.

3) Obtenha $\sqrt{685{,}72} \Rightarrow c = 0{,}68752\ .\ 10^3$

Solução:

$$\begin{cases} M = 0{,}68572 \\ p = 1 \\ q = 1 \end{cases} \quad \Rightarrow x_0 = (1{,}68 - \frac{1{,}29}{0{,}84 + 0{,}68572}) \;.\; 10^1 \;.\; 3{,}16^1$$

$$\Rightarrow x_0 = 26{,}370122$$

Aplicando Newton $\Rightarrow x_3 = 26{,}186255$

Exercícios:

1) Elabore uma rotina para efetuar a divisão sem utilizar a operação de divisão.
2) Crie uma rotina para obter $1/\sqrt{c}$, sem o uso da divisão e da radiciação.
3) Crie rotinas para obter: $\sqrt[3]{c}$ e $\sqrt[n]{c}$.

2.2.4 - Métodos Híbridos para f(x) = 0
Poli-algoritmos

Existe atualmente uma forte tendência na literatura a se mesclar os métodos das duas famílias anteriores para se criar métodos abrangentes e que minimizem as desvantagens dos submétodos envolvidos.

Exemplo: Obtenha a raiz próxima de zero de $e^x + x - 2 = 0$ com $\varepsilon = 10^{-5}$
Solução:
1º) Bisseção com [0;1] $\Rightarrow X_{M1}=0{,}5$
.
.
.
$X_{M17} = 0{,}44286$

- Lento
- Não obtém todos os $\alpha \in \Re$
- Necessita de $[a,b] \supset \alpha$

2º) Falsa Posição com [0;1] $\Rightarrow X_{F1} =$
$X_{F2} =$
.
.
.
$X_{F7} = 0{,}44285$

- Lento (um pouco menos)
- Não obtém todos os $\alpha \in \Re$
- Necessita de $[a,b] \supset \alpha$

3º) Newton com $X_0 = 0$
$X_1 = 0{,}5$
.
.
.
$X_3 = 0{,}44285$

- Necessita de f ` (x)
- Pode não convergir
- Pode convergir para um α já obtido

4º) Secante com $X_0 = 0$
$X_1 = 1$
$X_2 = 0{,}3788$

- Lento (comparando-se com Newton)
- Necessita de duas estimativas iniciais
- Menos abrangente
- Nem sempre converge.

$$\vdots$$

$X_5 = 0{,}44286$

2.2.4.a) Poli-Algoritmo Newton & Bisseção

Dado f(x) = 0 este poli-algoritmo permite obter todos os $\alpha \in \Re$ que estão em [a,b], seguindo-se os seguintes passos:

i) Toma-se o universo [a,b] e subdivide-o em k (livre) subintervalos de comprimento $h = {(b-a)}/{k}$;
ii) Rastrear cada subintervalo $[x_i ; x_{i+1}]$ onde $x_{i+1} = x_i + h$ e $x_0 = a$ para verificar onde existe $f(x_i) * f(x_{i+1}) < 0$;
iii) Refinar $[x_i ; x_{i+1}]$ por bisseção e Newton associados.

Observação: Note-se que ainda o método de Newton pode convergir para um α já obtido. Para evitar esta possibilidade devemos retirar, algebricamente, as raízes já determinadas através do seguinte procedimento:
→ Dados f(x) e α_1 tal que $f(\alpha_1) = 0$. Então, $f(x) = (x - \alpha).g(x)$, com o quociente da divisão entre f(x) e $(x - \alpha)$ não nulo ($g(x) \neq 0$). Toda raiz α_j de g(x) será também raiz de f(x). Este processo de divisão gera uma função g(x) com uma raiz α a menos que f(x). Este processo é usualmente denominado de deflação, na resolução de equações polinomiais.

→ Reescrevendo a expressão anterior para g(x), tem-se:

$$g(x) = \frac{f(x)}{x - \alpha_1}$$

→ Agora, aplica-se Newton em g(x) = 0

$$x_{k+1} = x_k - {g(x_k)}/{g'(x_k)}$$

$$x_{k+1} = x_k - \frac{f(x_k)}{(x_k - \alpha_1)} \cdot \frac{(x_k - \alpha_1)^2}{f'(x_k)(x_k - \alpha_1) - f(x_k)}$$

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k) - [f(x_k)/(x_k - \alpha_1)]}$$

Com este procedimento, evita-se o risco de Newton convergir novamente para α_1.
Para retirar j raízes:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k) - [\sum_{i=1}^{j} f(x_k)/(x_k - \alpha_i)]}$$

2.2.4.b) Poli-Algoritmo Dekker-Brent

Criado na década de 70, faz combinações judiciosas entre bisseção, falsa posição e secante.

2.3 - Solução de Equações por Classes
Solução de Equações Polinomiais

a) **Preliminares**

Devido à importância histórica e a freqüência com que aparecem nos modelos matemáticos, vamos aqui abordar a solução das equações polinomiais.

Uma equação polinomial é toda equação do tipo $p_n(x) = 0$, onde
$p_n(x) = a_1x^n + a_2x^{n-1} + ... + a_nx^1 + a_{n+1}$, com $a_i \in \Re$ e $a_1 \neq 0 \Rightarrow$ polinômio de grau n.

Como resolver $p_n(x) = 0$?

$n = 1 \rightarrow a_1x + a_2 = 0 \Rightarrow \alpha = -a_2/a_1$

$n = 2 \rightarrow a_1x^2 + a_2x + a_3 = 0 \Rightarrow \alpha = \dfrac{-a_2 \pm \sqrt{a_2^2 - 4a_1a_3}}{2a_1}$

Resolvido pelos:

- Babilônios em 1800a.C. (algebricamente)
- Gregos
- Hindus no século I d.C. (Báscara)

$n = 3 \rightarrow a_1x^3 + a_2x^2 + a_3x + a_4 = 0 \Rightarrow$ fazendo $x = y - a_2/ 3a_1$ teremos:

$y^3 + py + q = 0 \Rightarrow$

$$\alpha = \sqrt[3]{-q/2 + \sqrt{p^3/27 + q^2/4}} + \sqrt[3]{-q/2 - \sqrt{p^3/27 + q^2/4}}$$

$n = 4 \rightarrow a_1x^4 + a_2x^3 + a_3x^2 + a_4x + a_5 = 0$
Cardano divulgou um método de Ferrari de redução de qualquer equação de grau n = 4 para n = 3.

$n = 5 \rightarrow a_1x^5 + a_2x^4 + a_3x^3 + a_4x^2 + a_5x + a_6 = 0$
Em 1824 Abel provou que é impossível obter solução para $n \geq 5$ por radiciação.

b) **Avaliação do valor de $p_n(x)$**

Precisamos avaliar o valor numérico de $p_n(x)$, bem como $p'_n(x)$, em x=u (valor inicial x*), para podermos aplicar o método de Newton-Raphson às equações polinomiais $p_n(x)=0$.

Se expressarmos $p_n(x)$ na forma padrão $\Rightarrow$
$p_n(x) = a_1x^n + a_2x^{n-1} + ... + a_{n+1} \Rightarrow$

$$p_n(u) \text{ necessita de } \begin{cases} n & \text{adições} \\ & \quad \text{e} \\ 2n-1 & \text{multiplicações} \end{cases}$$

Porém expressando $p_n(x)$ na forma de Horner teremos:

$p_n(x) = ((((a_1x + a_2)x + a_3)x + a_4)x + ... + a_n)x + a_{n+1}$

Desta forma,

$$p_n(u) \text{ necessita de } \begin{cases} n & \text{adições e} \\ n & \text{multiplicações} \end{cases}$$

```
Rotina Horner
    Leia N, u
    Para i = 1 até n+1 faça
            leia a[i]
    fim para
    Para k = 1 até n faça
            p = p*u + a[k+1]
    fim para
Fim
```

Se usarmos a Divisão de Polinômios - Divisão Sintética - o cálculo de $p_n(u)$ também necessita de $\begin{cases} n & \text{adições e} \\ n & \text{multiplicações} \end{cases}$

Veja que $\quad p_n(x) \div (x - u) \Rightarrow \boxed{p_n(x) = (x-u).p_{n-1}(x) + \text{Resto}} \quad (1)$

onde o quociente da divisão, $p_{n-1}(x) = b_1x^{n-1} + b_2x^{n-2} + ... + b_n$, pode ser obtido através da igualdade da expressão a seguir, termo a termo:

$p_n(x) = (x-u).p_{n-1}(x) + R_1$ onde R_1 é o Resto e foi definido com b_{n+1}, por convenção.

$(a_1.x^n + a_2.x^{n-1} + ... + a_n.x^1 + a_{n+1})=(x-u).(b_1.x^{n-1} + b_2.x^{n-2} + b_3.x^{n-2} + ... + b_n)+ b_{n+1}$
$(a_1.x^n + a_2.x^{n-1} + ... + a_n.x^1 + a_{n+1})=((b_1).x^n +(b_2 - u.b_1).x^{n-1}+(b_3 - u.b_2).x^{n-2} + ... + (b_{n+1}-u.b_n)$

$a_1 = b_1$ $a_2 = b_2 - u.b_1$ $a_3 = b_3 - u.b_2$ . . . $a_n = b_n - u.b_{n-1}$ $a_{n+1} = b_{n+1} - u.b_n$	$b_1 = a_1$ $b_2 = a_2 + u.b_1$ $b_3 = a_3 + u.b_2$ . . . $b_n = a_n + u.b_{n-1}$ $\boxed{b_{n+1} = a_{n+1} + u.b_n = R_1}$

Observe que o resto R_1, definido como b_{n+1}, segue a mesma lei de formação de b_n, e por isso convencionou-se definir $R_1=b_{n+1}$.
Este método de divisão sintética de polinômios é conhecido por **Método de Briot-Ruffini**.

<u>Propriedade 4</u>:

"O valor numérico de $p_n(x)$ em x = u é o resto R_1 da primeira divisão de $p_n(x) \div (x\text{-}u)$."

Lembre que neste caso, $p_n(x)$ necessita também de $\begin{cases} n & \text{adições e} \\ n & \text{multiplicações} \end{cases}$

Pois $p_n(x{=}u) = (u-u).p_{n-1}(x) + R_1 = R_1 \quad \Rightarrow \quad p_n(x{=}u) = R_1$

<u>Propriedade 5</u>:

"O valor numérico de $p'_n(x)$ em x = u é o resto R_2 da segunda divisão de $p_n(x)$ por (x-u)."

Pois efetuando uma segunda divisão sintética $p_{n-1}(x) \div (x\text{-}u)$ temos que :

$\Rightarrow p_{n-1}(x) = (x - u).p_{n-2}(x) + R_2$

Substituindo $p_{n-1}(x)$ em $p_n(x) = (x-u).p_{n-1}(x) + R_1$

$\Rightarrow p_n(x) = (x - u).[(x - u).p_{n-2}(x) + R_2] + R_1$

$\Rightarrow p_n(x) = (x - u)^2 . p_{n-2}(x) + (x - u).R_2 + R_1$

Derivando $p_n(x)$ para obter $.p'_n(x)$, temos:

$\Rightarrow p'_n(x) = (x - u)^2. p'_{n-2}(x) + 2.(x - u).p_{n-2}(x) +1.R_2 + 0$

$p'_n(x{=}u) = (u - u)^2. p'_{n-2}(u) + 2.(u - u).p_{n-2}(u) +1.R_2 + 0$ e

$\Rightarrow p'_n(x{=}u) = R_2$

<u>Propriedade 6</u>:

" $p_n^{(k)}(u) = k!R_{k+1}$, onde R_{k+1} = resto da *(k+1)*-ésima divisão sucessiva de $p_n(x) \div (x\text{-}u)$."

(Importante para tratamento de raízes múltiplas, onde p'(x) tende a zero junto com p(x))

Exercício: Obtenha p'(2), p''(2), p'''(2) e p''''(2) em $p(x)=2x^4+3x-2$
(verificação com: $p'(x)=8x^3+3$; $p''(x)=24x^2$; $p'''(x)=48x$; $p''''(x)=48$).

Solução: em u = 2

$b_1 = 2$

$b_2 = 0 + 2 * 2 = 4$

$b_3 = 0 + 4 * 2 = 8$

$b_4 = 3 + 8 * 2 = 19$

$b_5 = R_1 = -2 + 19 * 2 = 36$ => p(2) = 36

$c_1 = 2$

$c_2 = 4 + 2 * 2 = 8$

$c_3 = 8 + 2 * 8 = 24$

$c_4 = R_2 = 19 + 24 * 2 = 67$ => p'(2) = 1! . R_2 = 1*67

$d_1 = 2$
$d_2 = 8 + 2 * 2 = 12$
$d_3 = R_3 = 24 + 2 * 12 = 48$ => $p''(2) = 2! * R_3 = 2.1 * 48 = 96$

$e_1 = 2$
$e_2 = R_4 = 12 + 2 * 2 = 16$ => $p'''(2) = 3! * R_4 = 3.2.1 * 16 = 96$

$f_1 = R_5 = 2$ => $p''''(2) = 4! * R_5 = 4.3.2.1 * 2 = 48$

Exercício: Crie uma rotina que calcule a derivada qualquer de um polinômio

Teorema 1: (Teorema Fundamental da Álgebra)
"Toda polinomial $p_n(x) = 0$ tem raiz."

Propriedade 7:
Se α é raiz de $p_n(x) = 0$ então, $p_n(x)$ é divisível por $(x - \alpha)$.

Propriedade 8:
Uma raiz α de $p_n(x) = 0$ possui multiplicidade M se e somente se $p_n^{(i)}(\alpha) = 0$, $\forall$ i = 0, 1, ..., M-1 e $p_n^{(M)}(\alpha) \neq 0$.
Alternativamente, podemos verificar o número M de vezes que $p_n(x)$ é divisível por $(x-\alpha)$, ou seja, se $p_n(x)$ é M vezes divisivel por $(x-\alpha)$, a raiz α é de multiplicidade M e $R_1 = R_2 = ... = R_M = 0 \; e \; R_{M+1} \neq 0$.

Corolário 1:
Uma polinomial $p_n(x) = 0$ possui 'n' raízes, reais ou complexas, distintas ou repetidas.

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

Solução:
$p(x) = x^4 - 5x^3 + 6x^2 + 4x - 8 \Rightarrow p(2) = 0$
$p'(x) = 4x^3 - 15x^2 + 12x + 4 \Rightarrow p'(2) = 0$
$p''(x) = 12x^2 - 30x + 12 \Rightarrow p''(2) = 0$
$p'''(x) = 24x - 30 = 18 \Rightarrow p'''(2) \neq 0 \Rightarrow M = 3 \;\therefore\; \alpha = 2$ é repetida 3 vezes

Exercício: Crie uma rotina para testar a multiplicidade de uma raiz $\alpha \in \Re$ de uma polinomial $p_n(x) = 0$.

d) **Localização de raízes α de $p_n(x) = 0$**

<u>Propriedade 9</u>: Em $a_1x^n + a_2x^{n-1} + ... + a_{n-1} = 0$, seja $M = máx\{|a_2|, |a_3|, ..., |a_{n+1}|\}$. Então $\forall$ α raiz $\Rightarrow |\alpha| < 1 + \frac{M}{|a_1|}$

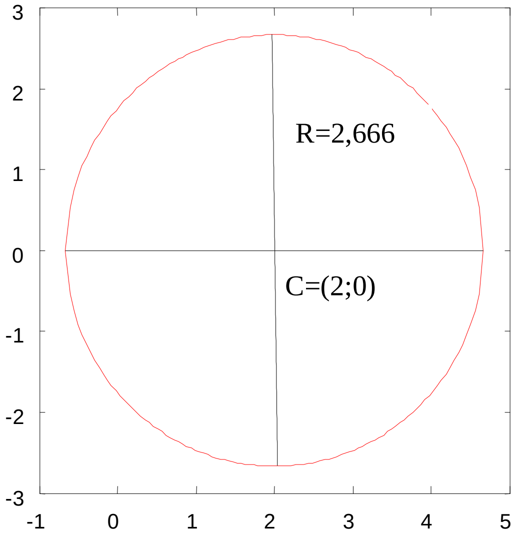
Exemplo: Localize as raízes de $3x^6 + 4x^3 - 2x^2 - 6 = 0$

Solução:
Aplicando a propriedade 9 temos que:
$M = 6 \Rightarrow |\alpha| < 1 + \frac{6}{|3|} \Rightarrow |\alpha| < 3 \;\cong\; \alpha \in (-3,3)$ se $\alpha \in \Re$

<u>Propriedade 10</u>:

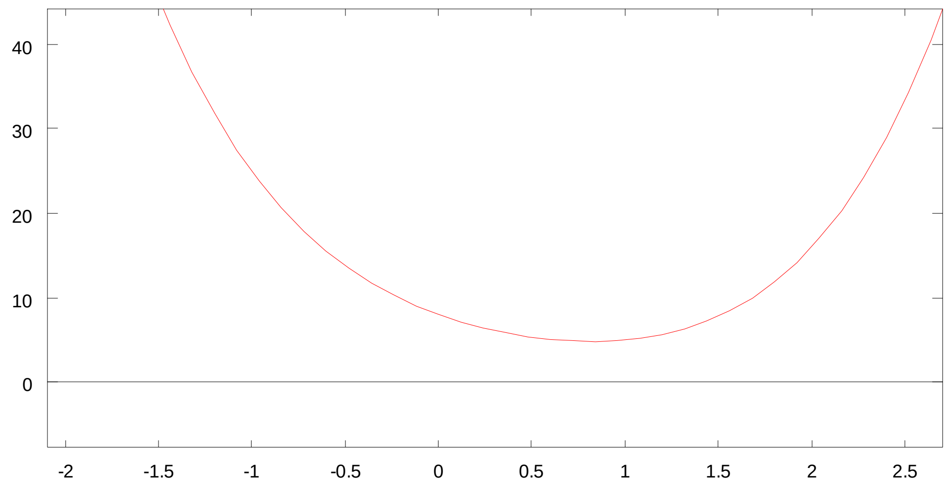
Seja u = a + bi tal que $p'(u) \neq 0$ e $R = n * \left|\frac{p_n(u)}{p'_n(u)}\right| \Rightarrow$ existe pelo menos uma raiz α no círculo de centro u = (a,b) e raio R.

Exemplo: Para $p_4(x)=x^4 - 2x^3 + 6x^2 - 8x + 8$ com u = 2+0.i.
Solução:
$\Rightarrow n=4$
$\Rightarrow p_n(2) = 16$
$\Rightarrow p'_n(2) = 24$
$\Rightarrow R = 4 * \frac{16}{24} = 2{,}666... \Rightarrow \exists$ pelo menos uma raiz α no circulo de raio R=2,666 e centro C=(2;0), no gráfico com abscissa x Real e ordenada y Imaginária:



Podemos verificar que todas as raízes deste polinômio são $1 \pm i$ e $\pm 2i$, portanto dentro do circulo vermelho do gráfico acima.

O polinômio $p_4(x)$ não tem zeros reais, conforme gráfico abaixo.



e) **Natureza das raízes**

<u>Propriedade 11</u>:

As raízes complexas de equações polinomiais de "coeficientes reais" sempre ocorrem aos pares conjugados, isto é, se $\alpha = a + bi$ for raiz, $\alpha = a - bi$ também é raiz.

Exemplo: $x^2 - x + 2 = 0$

Solução:

Por Báscara temos que: $\alpha = \dfrac{1 \pm \sqrt{1-8}}{2} \Rightarrow \alpha = \dfrac{1 \pm \sqrt{-7}}{2}$

$$\alpha_1 = 0{,}5 + \sqrt{7/4}\, i \text{ e } \alpha_2 = 0{,}5 - \sqrt{7/4}\, i$$

Conseqüência:

Toda $p_n(x) = 0$ de "coeficientes reais" com 'n' ímpar possui ao menos uma raiz real.

<u>Propriedade 12</u>:

Em $p_n(x) = 0$ se n for par e $a_1 * a_{n+1} < 0 \Rightarrow$ existe ao menos duas raízes reais, uma positiva e outra negativa.

Exemplo: $x^{12} - 5x^2 + 3x - 2 = 0$ possui ao menos duas raízes reais.

<u>Propriedade 13</u>

Em $p_n(x) = 0$, se existe k = 2, 3, ..., n tal que $a_k^2 \leq a_{k+1} * a_{k-1}$ e $a_{n+1} \neq 0 \Rightarrow$ existem raízes complexas.

<u>Propriedade 14</u>: Regra dos sinais de Descartes

Origem: Wikipédia, http://pt.wikipedia.org/wiki/Regra_dos_sinais_de_Descartes

A **regra dos sinais de Descartes**, primeiramente descrita por René Descartes no seu trabalho *La géométrie*, é um teorema que determina o número de raízes positivas e negativas de um polinômio.

Segundo a regra, se os termos de um polinômio com coeficientes reais não nulos Pn(x) são colocados em ordem decrescente de grau, então o número de raízes positivas do polinômio é igual ao número de permutações de sinal de Pn(x), ou menor por uma diferença par. O número de raízes negativas do polinômio é igual ao número de permutações de sinal de Pn(-x), ou menor por uma diferença par. Ou seja, o número de permutações é igual ao número de raízes positivas e negativas acrescido do número de raízes imaginárias (que sempre acontecem ao pares em polinômio de coeficientes reais).

Exemplo

$$x^3 + x^2 - x - 1$$

Possui uma mudança de sinal entre o segundo e o terceiro termos. Portanto possui apenas uma raíz positiva.

Para contar o número de raízes negativa, fazemos a substituição $x \longrightarrow -x$:

$$-x^3 + x^2 + x - 1$$

Este polinômio tem duas permutações de sinal, logo o polinômio original possui 2 ou 0 raízes negativas.

Para confirmar o resultado, observe a fatoração do polinômio:

$$(x+1)^2(x-1),$$

Então as raízes são **-1** (aparece 2 vezes) e **1** (aparece 1 vez).

f) **Determinação (refinamento) de raízes**

1º) Tomando $p_n(x) = a_1x^n + ... + a_{n+1}$ e um $x_0 \cong \alpha$ e aplicando o método de Newton para refinar vem: $\Delta x = -\dfrac{p_n(x_k)}{p_n'(x_k)} = -\dfrac{R_1}{R_2} \rightarrow x_{k+1} = x_k + \Delta x$ (também chamado de Método de Birge-Vieta).

onde

R_1 = resto de $p_n(x) \div (x - x_k) = p_n(x_k)$

R_2 = resto de $p_{n-1}(x) \div (x - x_k) = p_n'(x_k)$

sendo $p_{n-1}(x)$ = polinômio quociente da primeira divisão.

Exemplo: Aproxime com 3 iterações uma 1ª raiz de $p_3(x)=2x^3 - x - 2 = 0$ por Newton a partir de $x_0=1$.

Solução: $p_3(x)=2.x^3+0.x^2 -1.x - 2.x^0 = 0$

$a_1 = +2$
$a_2 = 0$
$a_3 = -1$
$a_4 = - 2$

Aplicando a divisão sintética de Briot-Rufini:

1ª iteração: a partir de $x_0=1$.

$b_1 = 2$
$b_2 = 0 + 2 . 1 = 2$
$b_3 = -1 + 2 . 1 = 1$
$b_4 = p(1)= R_1 = - 2 + 1 . 1 = -1= R_1$

$c_1 = b_1 = 2$
$c_2 = b_2 + x_0.c_1 = 2 + 2 . 1 = 4$
$c_3 = p'(1) = R_2 = b_3 + x_0.c_2 = 1 + 1 . 4 = 5 = R_2$

$x_1 = x_0 - \frac{R_1}{R_2} = 1 - (-\frac{1}{5}) = \underline{1,2}$

2ª iteração: a partir de $x_1 = 1,2$

$b_1 = 2$
$b_2 = 0 + 1,2 . 2 = 2,4$
$b_3 = -1 + 1,2 . 2,4 = 1,88$
$b_4 = R_1 = 0,256= R_1$

$c_1 = 2$
$c_2 = 2,4 + 1,2 . 2 = 4,8$
$c_3 = R_2 = 1,88 + 1,2 . 4,8 = 7,64 = R_2$

$x_2 = 1,2 - (\frac{0,256}{7,64}) = \underline{1,166492}$

3ª iteração: a partir de $x_2 = 1,166492$

$b_1 = 2$
$b_2 = 0 + 2 . 1,166492 = 2,332984$
$b_3 = -1 + 2,332984. 1,166492= 1,721408$
$b_4 = R_1 = - 2 + 1,721408. 1,166492= 0,008009= R_1$

$c_1 = b_1 = 2$
$c_2 = 2,332984+ 1,166492. 2 = 4,665969$
$c_3 = R_2 = b_3 + x_0.c_2 = 1,721408+ 1,166492. 4,665969 = 7,164224 = R_2$

$x_3 = 1,166492- (\frac{0,008009}{7,164224}) = 1,165374 \cong \alpha_1$

com Erro=$|p(x_3=1,165374)=0,00000874|<0,001$.

Observação:

Para determinar a 2ª e 3ª raízes e se evitar de obter a mesma 1ª raiz α_1 novamente, pode-se efetuar redução de grau (<u>deflações</u>), isto é:

- Tomar $p_n(x) \div (x - \alpha_1)$ obtendo $p_{n-1}(x)$, nesse caso com Resto R_1 será quase nulo (nas divisões exatas, $R_1=0$);

$p_3(x)=2.x^3+0.x^2 -1.x - 2.x^0$ dividido por (x-1,165374):

$p_3(x)=2.x^3+0.x^2 -1.x - 2.x^0 = p_2(x).(x-1,165374)$ -> $p_2(x) = b_1.x^2+ b_2.x^1 + b_3.x^0$

$b_1 = 2$
$b_2 = 0 + 2 . 1,165374= 2,330749$
$b_3 = -1 + 2,330749. 1,165374= 1,716194$
$b_4 = R_1 = - 2 + 1,716194. 1,165374= 0,00000874 = R_1$

$p_2(x)=2.x^2+2,330749.x^1 +1,716194.x^0 = 0$

- Testar a multiplicidade M de α_1, verificando se $p'_n(\alpha_1)=0$, $p''_n(\alpha_1)=0$, ... $p^M{}_n(\alpha_1)\neq 0$, ou reaplicar diretamente o método no polinômio deflacionado.

$c_1 = b_1 = 2$
$c_2 = 2,330749+ 1,165374 . 2 = 4,665969$
$c_3 = R_2 = b_3 + x_0.c_2 = 1,716194+ 1,165374 . 4,665969 = 7,148583 = R_2$
$p'_3(\alpha_1=1,165374)\neq 0$ –> Multiplicidade M=1 -> $\alpha_1=1,165374$ Não repetida.

- Determinar a 2ª e 3ª raízes em $p_2(x)=2.x^2+2,330749.x^1 +1,716194.x^0 = 0$, que pode ser via Newton-Raphson a partir de um novo x_0 ou aplicar diretamente por Bascara.

Algoritmo de Newton-Raphson para determinação de raízes com multiplicidade M:

Exemplo: Determine as 3 raízes de $P(x)=x^3-3.x^2+3.x-1=(x-1)^3=0$. Verifique que $\alpha_1=1$ tem multiplicidade M=3:

Pela divisão sintética de Briot-Ruffini:

1a. Divisão	1	1	-3	3	-1	
			1	-2	1	
2a. Divisão	1	1	-2	1	0	=> R1
			1	-1		
3a. Divisão	1	1	-1	0	=> R2	
			1			
4a. Divisão	1	1	0	=> R3		
		1	=> R4 (não é possivel dividir 1/(x-1))			

ou usando as divisões tradicionais:

$1x^3-3x^2+3x-1$	x-1			
R1=0	$1x^2-2x+1$	x-1		
	R2=0	x-1	x-1	
		R3=0	1	x-1
			R4=1	0

Pela Propriedade 6: " $p_n^{(k)}(u) = k!R_{k+1}$, onde R_{k+1} = resto da (*k+1*)-ésima divisão."

$P(1) = 0! * R_1 = 1*0 = 0$
$P'(1) = 1! * R_2 = 1*0 = 0$
$P''(1) = 2! * R_3 = 2*0 = 0$
$P'''(1) = 3! * R_4 = 6*1 = 6$ => Pela propriedade 8, a Multiplicidade de $\alpha_1=1$ é M=3.

A essência do Método de Newton, para raízes não repetidas, prevê que $\Delta x = -\dfrac{p_n(x_k)}{p_n'(x_k)} = -\dfrac{R_1}{R_2}$

mas observe que no caso de aproximações de raízes x_k múltiplas, R_1 e R_2 tendem a zero juntos, e podem gerar uma indeterminação 0/0, reduzindo a taxa de convergência (perde-se a taxa de convergencia quadrática) e assim precisaremos aplicar a regra de L'Hospital ao valor em torno de uma aproximação de x_k "ainda em processo de determinação":

$$\Delta x = -\frac{p_n(x_k)}{p_n'(x_k)} = -\frac{0!.R_1}{1!.R_2} \text{ para M} = 1 \quad \text{(Formula normal)}$$

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

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

.

.

$$\Delta x = -\frac{p_n^{M-1}(x_k)}{p_n^{M}(x_k)} = -\frac{(M-1)!.R_M}{M!.R_{M+1}} = -\frac{(M-1)!.R_M}{M.(M-1)!.R_{M+1}} = -\frac{1}{M}.\frac{R_M}{R_{M+1}} \text{ para M genérico}$$

Temos uma dificuldade adicional na determinação de raízes múltiplas, que é o fato de não conhecermos as raízes x_k previamente para definir a sua multiplicidade, mas é possível verificar que no processo iterativo de convergência, os valores sucessivos de x_k vão se aproximando de cada raiz de modo gradativo, "lento no caso de raízes múltiplas", onde perde-se a convergência quadrática, quando R_2 também tende a zero gradativamente no denominador de Δx. Normalmente não atingimos um valor com erro razoável, com R_1/R_2.

Para acelerar a convergência, é necessário definir um valor limite mínimo de R_2, R_3,...,R_M, no caso de raízes de multiplicidade M, ou seja, definir quais valores de R_2, R_3,...,R_M, são pequenos suficientes para ser considerados nulos (ou quase nulos), por exemplo, se $|R_2|$, $|R_3|$,..., $|R_M|$ são menores ou iguais a 0,1 (precisa ser melhor testado) podemos dizer que a Multiplicidade da raiz x_k "em avaliação" é M, para o processo iterativo. Assim, para uma raiz x_k de multiplicidade M genérica:

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

Um algoritmo possível para o método de Newton-Raphson, generalizado para raízes x_k de multiplicidade M pode ser o seguinte:

```
function p = FNRPolM(xa,n,ai,tol)
## Metodo Newton-Raphson para polinomios
% n  % Grau do polinomio Pn(x)
% ai  % Coeficientes do polinomio na entrada
%Encontrando uma das n raizes de Pn(x)=0, com multiplicidade M
# xa  Valor inicial das raízes
erro=1.;iter=0; r(n+1)=0;
while(iter<50 & erro>tol)
   a=ai;
   iter=iter+1
   naux=n;
   k=1;   % contador de multiciplidade M
   M=1;  % Multiplicidade inicial de xa
   while(k<=M+1)
         printf ("\n k=%d, naux=%d \n",k,naux);
         b(1)=a(1);
         for i=2:(naux+1)
            b(i)=a(i)+xa*b(i-1);
         end
         r(k)=b(naux+1);% %r(n+1)=b(1);  %Ultimo resto, se todas as 'n' divisões ocorrerem
%Teste de multiplicidade M
         if abs(r(k))<0.10
            M=k; %[k r(k) M]
         endif
%        c(1)=b(1);
%        for i=2:n
%          c(i)=b(i)+xa*c(i-1);
%        end
%        r2(k)=c(n);
         naux=naux-1;
         a=b;
         k=k+1; %N. de divisões
         r
```

```
        M
        endwhile %k
%       M=1; % Desativa a correção de multiplicidade
        Dx=-r(M)/(M*r(M+1));  % M=1 => f/f'    = 0!.r(1)/1.0!.r(2)
                              % M=2 => f'/f''  = 1!.r(2)/2.1!.r(3)
                              % M=3 => f''/f''' = 2!.r(3)/3.2!.r(4) => r(M)/(M.r(M+1))
        xn=xa+Dx
        erro=abs(Dx)
        xa=xn;
endwhile
p=xn;
erro;
endfunction
```

Exemplo de refinamento de raízes:

$p_4(x)=x^4 - 11{,}101x^3 + 11{,}1111x^2 - 1{,}0111x + 0{,}001$ tem as raízes:

$\alpha_1 = 0{,}001$;

$\alpha_2 = 0{,}1$;

$\alpha_3 = 1$ e

$\alpha_4 = 10$.

Aqui poderá ocorrer o erro da instabilidade numérica, se não aplicarmos o refinamento das raízes encontradas no processo de redução de grau.

Considere encontrada a raiz $\overline{\alpha_4} = 10{,}000005$. Deflacionando teremos:

$p_4(x) \div (x - \overline{\alpha_4}) \Rightarrow p_3(x) = x^3 - 1{,}0099x^2 + 0{,}101144x + 0{,}3405$ que não possui raízes próximas de $\alpha_1 = 0{,}001$, pois $p_3(0{,}001) = 0{,}3406$.

A solução para este problema seria um refinamento dos resultados parciais, a partir da segunda raiz, ou seja, reaplicar o método de Newton-Raphson ao polinômio original de grau n original, $p_4(x)=0$, usando as raízes parciais obtidas nos polinômios deflacionados como valor inicial de 'raiz' do polinômio original de grau n, $p_4(x)=0$.

Exemplo 1 : Determinar as 3 raízes de $P_3(x)=x^3-3.x^2+3.x-1=(x-1)^3=0$.

Testar **function p = FNRPolM(xa,n,ai,tol)**

Raizes:

1.0000000000000000
1.0000000000000000
1.0000000000000000

Testar roots([1, -3, 3, -1]) do octave -> Não converge com a mesma exatidão.

1.000004484536201 + 0.000000000000000i
0.999997757731900 + 0.000003883723679i
0.999997757731900 - 0.000003883723679i

Exemplo 2 : Determinar as 4 raízes de $P_4(x)= x^4+3x^3+x^2+3x-1=0$.

Testar **function p = FNRPolM(xa,n,ai,tol)**

Raizes:

0.282200133829172 + 0.000000000000000i
-3.032346594830436 + 0.000000000000000i
-0.124926769499369 + 1.073772830301651i
-0.124926769499369 - 1.073772830301651i

Testar roots([1, 3, 1, 3, -1]) do octave -> Converge com a mesma exatidão:

0.282200133829173 + 0.000000000000000i
-3.032346594830435 + 0.000000000000000i
-0.124926769499369 + 1.073772830301651i
-0.124926769499369 - 1.073772830301651i

Exemplo 3 : Determinar as 6 raízes de $P_6(x)$ = $x^6+0.9261.x^5-4.54057649667.x^4-2.95044074436.x^3+7.30497592295.x^2+2.39702970297.x-4.16412047140=0$

Testar **function p = FNRPolM(xa,n,ai,tol):**

Raizes:

1.12340000869803

tol=0.00000001

1.12340000869803
1.12340000869803
-1.43210002128482
-1.43210002128482
-1.43210002128482

Testar roots([1, 0.9261, -4.54057649667, -2.95044074436, 7.30497592295, 2.39702970297, -4.16412047140]) do octave -> Não converge com a mesma exatidão:

1.12516234299630 + 0.00305122247113i
1.12516234299630 - 0.00305122247113i
1.11987535143748 + 0.00000000000000i
-1.43398911089592 + 0.00326535282788i
-1.43398911089592 - 0.00326535282788i
-1.42832181563823 + 0.00000000000000i

2^o) Método de Müller

Trata-se de uma extensão do Método da Secante, e que busca minimizar alguns problemas presentes nos métodos de Newton, Secante, Falsa-posição, tais como:

- Convergência lenta quando a função e sua derivada são próximos de zero;
- Não encontram raízes complexas a menos que seja dado um valor inicial complexo.

Em geral, os métodos citados são pobres para encontrar raízes complexas em polinômios, pois mesmo em se tratando de polinômios com coeficientes reais, ele normalmente tem raízes complexas.

Para melhor entender o método de Müller vamos analisar a figura 1 abaixo, para a resolução da equação,

$$f(x) = 0 \qquad (1)$$

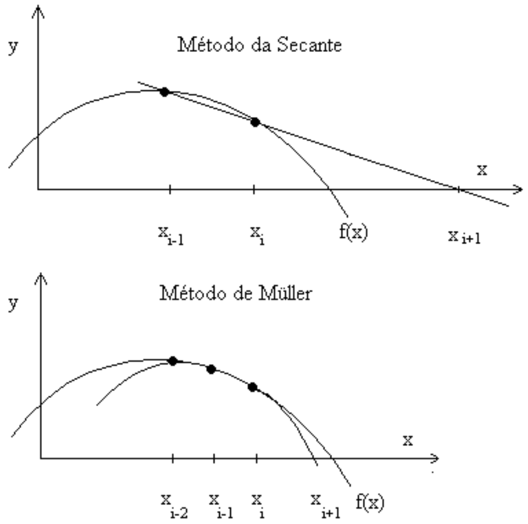


Figura 1 - Comparativo entre Método da Secante e Müller.

No método da Secante avalia-se o zero da função linear que passa por dois pontos da curva de f(x)=0. Já no método de Müller avalia-se o zero da função quadrática que passa por três pontos da curva de f(x)=0 a ser resolvida.

Supomos que três aproximações iniciais x_{i--2}, x_{i-1} e x_i são dadas para a solução de f(x)=0. A aproximação seguinte x_{i+1} é obtida do zero do polinômio quadrático P(x) que passa pelas três aproximações iniciais de f(x).

$$P(x) = a\,(x- x_i)^2 + b(x - x_i) + c \qquad (2)$$

Portanto,

$$\begin{cases} P(x_{i-2}) = a(x_{i-2} - x_i)^2 + b(x_{i-2} - x_i) + c = f(x_{i-2}) \\ P(x_{i-1}) = a(x_{i-1} - x_i)^2 + b(x_{i-1} - x_i) + c = f(x_{i-1}) \\ P(x_i) = a(x_i - x_i)^2 + b(x_i - x_i) + c = f(x_i) \end{cases} \qquad (3)$$

Das eqs. (3) obtemos os coeficientes a, b e c e encontramos a aproximação seguinte x_{i+1} resolvendo $P(x_{i+1}) = 0$ (usando a fórmula de Báskara).

$$x_{i+1} = x_i - (x_i - x_{i-1})\left[\frac{2c}{b \pm \sqrt{b^2 - 4ac}}\right] \qquad (4)$$

onde

$a = q\, f(x_i)-q\,(1+q)\, f(x_{i-1}) + q^2 f(x_{i-2})$
$b = (2q+1)\, f(x_i) - (1+q)^2 f(x_{i-1}) + q^2 f(x_{i-2})$
$c = (1+q)\, f(x_i)$

$$q = \frac{x_i - x_{i-1}}{x_{i-1} - x_{i-2}}$$

A fórmula escolhida para a eq.(4) produz menos erros de arredondamento que a forma tradicional. Note que na eq. (4) temos duas possibilidades para obter x_{i+1}, dependendo do sinal escolhido no radicando.

No método de Müller o sinal do radicando é escolhido de modo que o valor absoluto do denominador, ou módulo para números complexos, seja o maior possível. Esta escolha, que gera um denominador maior em magnitude, evita possibilidades de denominador nulo, e gera um valor de x_{i+1} mais próximo de x_i, dando mais estabilidade ao método numérico.

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

$$x_{i+1} = x_i - (x_i - x_{i-1})\left[\frac{2c}{\mathrm{Max}\left|b \pm \sqrt{b^2 - 4ac}\right|}\right] \qquad (5)$$

No caso onde as parcelas do denominador forem números reais, a expressão para calcular xi+1 pode ser dada pela eq. (6).

$$x_{i+1} = x_i - (x_i - x_{i-1})\left[\frac{2c}{b \pm \mathrm{sign}(b)\sqrt{b^2 - 4ac}}\right] \qquad (6)$$

Então aplica-se iterativamente a fórmula dada na eq. (5) até que uma condição de parada seja satisfeita.

Note que, se o radicando gerar números complexos, podemos então obter raízes complexas de f(x)=0. Além disso, este método não está limitado somente a equações com coeficientes reais, pode-se encontrar inclusive raízes complexas de equações transcendentais. Também podemos ter equações com coeficientes complexos e que podem gerar raízes complexas não conjugadas.

<u>Exercícios:</u>

1). Determine um zero do polinômio $Pn(x) = 16x^4-40x^3+5x^2+20x+6$, partindo das seguintes condições iniciais:

i). $x_0 = 0.5$, $x_1 = 1.0$ e $x_2 = 1.5$

ii). $x_0 = 0.5$, $x_1 = -0.5$ e $x_2 = 0.0$

2). Compare a eficiência do método de Müller frente ao Método da Secante, partindo de uma mesma condição inicial x_0, para equações a sua escolha.

Sugestão para valores iniciais:

Secante: x_0 e $x_1 = 1{,}10\ x_0$

Müller: x_0, $x_1 = 1{,}05\ x_0$ e $x_2 = 1{,}10\ x_0$