

# TMEC001 - CÁLCULO NUMÉRICO

## CAPÍTULO 02 – SOLUÇÕES NUMÉRICAS DE EQUAÇÕES DE UMA VARIÁVEL

*Prof. Felipe R. Loyola*  
*Disciplina: Cálculo Numérico*  
*2º Semestre de 2019*

# 2.3 Raízes de Polinômios

## 2.3 – Raízes de Polinômios

- As raízes de um polinômio  $f_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$  seguem as seguintes regras:
  1. Para uma equação de grau  $n$ , existem  $n$  raízes reais ou complexas deve-se observar que essas raízes não serão necessariamente distintas
  2. Se  $n$  for ímpar, existe pelo menos uma raiz real
  3. Se existirem raízes complexas, elas existem em pares conjugados (isto é,  $\lambda + \mu i$  e  $\lambda - \mu i$ ) onde  $i = \sqrt{-1}$

## 2.3 – Raízes de Polinômios

---

- Na prática da engenharia o cálculo e derivação de polinômios são amplamente utilizados para ajuste de curvas. Além disso, uma aplicação muito importante está na caracterização de **sistemas dinâmicos**, sobretudo em sistemas lineares. Pode-se citar como exemplo dispositivos mecânicos, estruturas e circuitos elétricos.

## 2.3.1 – Cálculo e Derivação de Polinômios

- Embora seja a forma mais comum:

$$f_n(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n \quad (24)$$

Tal equação fornece **métodos insatisfatórios** para determinar o valor de um polinômio em um valor de  $x$  particular. Por exemplo, calcular um polinômio de terceiro grau como:

$$f_3(x) = a_3x^3 + a_2x^2 + a_1x + a_0 \quad (25)$$

Envolve seis multiplicações e três adições. Em geral, para um polinômio de grau  $n$ , essa abordagem exige  $n(n + 1)/2$  multiplicações e  $n$  adições.

## 2.3.1 – Cálculo e Derivação de Polinômios

- Em contraste, uma forma estruturada,

$$f_3(x) = ((a_3x + a_2)x + a_1)x + a_0 \quad (26)$$

envolve três multiplicações e três adições. Para um polinômio de grau  $n$ , essa abordagem exige  **$n$  multiplicações** e  **$n$  adições**. Como a forma estruturada minimiza o número de operações, tende a minimizar os erros de arredondamento. Observe-se que, dependendo da preferência, a ordem da estrutura pode ser invertida:

$$f_3(x) = a_0 + x(a_1 + x(a_2 + xa_3)) \quad (27)$$

## 2.3.1 – Cálculo e Derivação de Polinômios

- Um pseudocódigo sucinto para implementar a forma estruturada pode ser escrito simplesmente como:

DO FOR  $j = n, 0, -1$

$$p = p * x + a(j)$$

END DO

onde  $p$  guarda o valor do polinômio (definido por seus coeficientes, os  $a$ 's) calculado em  $x$ .

## 2.3.1 – Cálculo e Derivação de Polinômios

- Há casos, como no método de Newton-Raphson, em que se pode querer calcular tanto a função quanto sua derivada. Esse cálculo também pode ser naturalmente incluído adicionando-se uma única linha no pseudocódigo anterior:

```
DO FOR j = n, 0, -1
```

```
    df = df * x + p
```

```
    p = p * x + a(j)
```

```
END DO
```

onde df guarda o valor da primeira derivada do polinômio.

## 2.3.1 – Cálculo e Derivação de Polinômios

---

### \* Deflação polinomial

Suponha que uma única raiz de um polinômio de grau  $n$  tenha sido determinado. Se repetir o procedimento para a localização de raízes, poder-se-á encontrar a mesma raiz. Portanto, seria bom remover a raiz encontrada antes de continuar. Esse procedimento de remoção é chamado de deflação polinomial.

## 2.3.1 – Cálculo e Derivação de Polinômios

Um esquema simples é fornecido pelo seguinte pseudocódigo, que divide um polinômio de grau  $n$  por um fator do tipo monômio  $x-t$ :

$$r = a(n)$$

$$a(n) = 0$$

DO FOR  $i = n-1, 0, -1$

$$s = a(i)$$

$$a(i) = r$$

$$r = s + r*t$$

END DO

Se o monômio corresponder a uma raiz do polinômio, o resto  $r$  será nulo e os coeficientes do quociente estarão armazenados em  $a$  no final do laço.

## 2.3.1 – Cálculo e Derivação de Polinômios

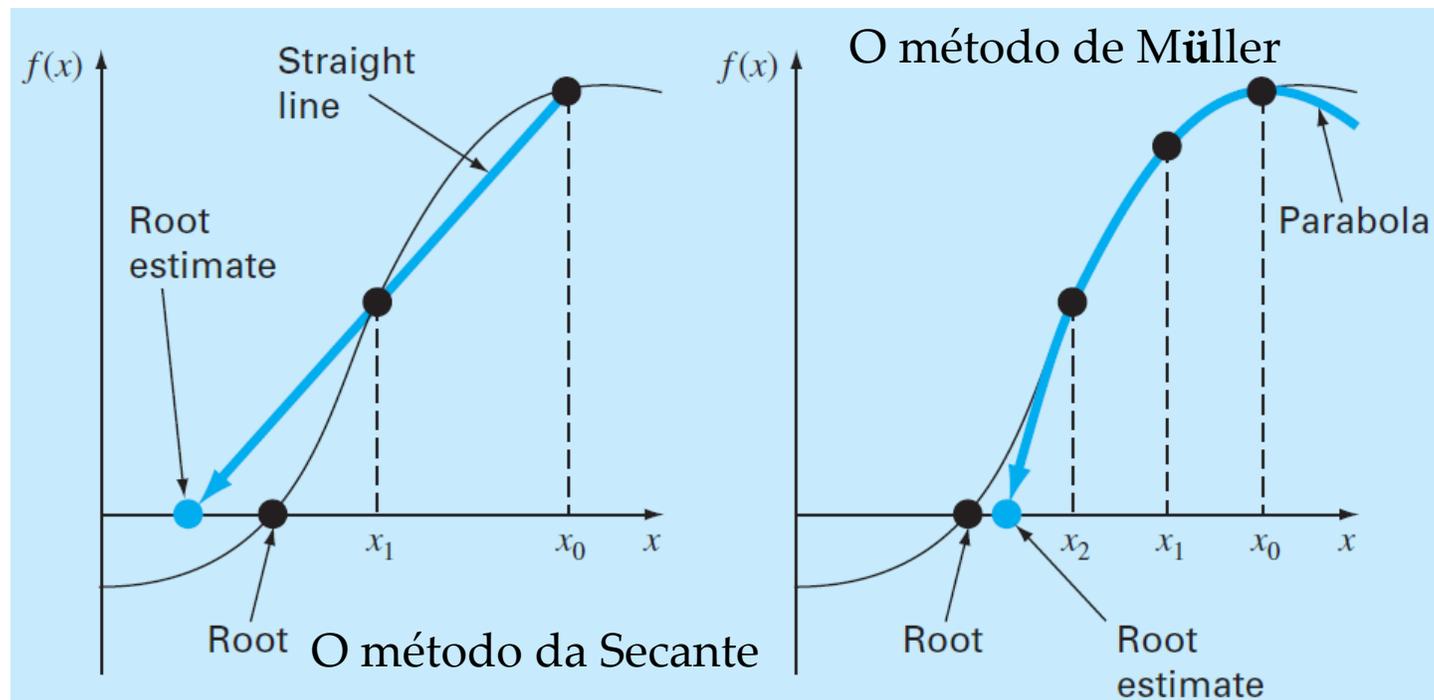
---

**Exemplo:** Divida o polinômio de segunda-ordem a seguir pelo fator  $(x - 4)$ :

$$f(x) = (x - 4)(x + 6) = x^2 + 2x - 24$$

## 2.3.2 – O Método de Müller

O método da secante obtém uma estimativa da raiz prolongando uma reta por dois valores da função até o eixo  $x$ . O método de Müller adota uma abordagem parecida, mas prolonga uma parábola através de três pontos:



## 2.3.2 – O Método de Müller

O método consiste na determinação dos coeficientes da parábola que passa pelos três pontos. Esses coeficientes podem então ser substituídos na fórmula quadrática para se obter o ponto no qual a parábola intercepta o eixo  $x$ , isto é, a estimativa da raiz. O método se torna mais fácil escrevendo-se a equação da parábola em uma forma conveniente:

$$f_2(x) = a(x - x_2)^2 + b(x - x_2) + c \quad (28)$$

## 2.3.2 – O Método de Müller

Observa-se que essa parábola passa por três pontos  $(x_0, f(x_0))$ ,  $(x_1, f(x_1))$  e  $(x_2, f(x_2))$ . Os coeficientes da equação (28) pode ser calculados substituindo cada um dos três pontos para obter:

$$f(x_0) = a(x_0 - x_2)^2 + b(x_0 - x_2) + c \quad (29)$$

$$f(x_1) = a(x_1 - x_2)^2 + b(x_1 - x_2) + c \quad (30)$$

$$f(x_2) = a(x_2 - x_2)^2 + b(x_2 - x_2) + c \quad (31)$$

## 2.3.2 – O Método de Müller

Solucionando-se tal sistema, tem-se:

$$a = \frac{\delta_1 - \delta_0}{h_1 + h_0} \quad (32)$$

$$b = ah_1 + \delta_1 \quad (33)$$

$$c = f(x_2) \quad (34)$$

sendo

$$h_0 = x_1 - x_0 \quad (35)$$

$$h_1 = x_2 - x_1 \quad (36)$$

$$\delta_0 = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \quad (37)$$

$$\delta_1 = \frac{f(x_2) - f(x_1)}{x_2 - x_1} \quad (38)$$

## 2.3.2 – O Método de Müller

Para encontrar a raiz, aplica-se a fórmula quadrática (28). No entanto, por causa do possível erro de arredondamento, em vez de usar a forma convencional, será usada a formulação alternativa para obter:

$$x_3 - x_2 = \frac{-2c}{b \pm \sqrt{b^2 - 4ac}} \quad (39)$$

ou isolando a incógnita  $x_3$ ,

$$x_3 = x_2 + \frac{-2c}{b \pm \sqrt{b^2 - 4ac}} \quad (40)$$

observa-se que o uso da fórmula quadrática significa que podem ser localizadas tanto raízes reais quanto complexas. Essa é uma importante vantagem do método.

## 2.3.2 – O Método de Müller

Além disso, a equação (39) fornece uma forma natural de determinar o erro aproximado. Como o lado esquerdo representa a diferença entre a estimativa atual ( $x_3$ ) e anterior ( $x_2$ ) da raiz, o erro pode ser calculado por:

$$\varepsilon_a = \left| \frac{x_3 - x_2}{x_3} \right| 100\% \quad (41)$$

Um problema com a equação (39) é que ela fornece duas raízes, correspondendo ao termo  $\pm$  no denominador. No método de Müller, o sinal é escolhido para coincidir com o sinal de  $b$ . Essa escolha resultará no denominador maior e, portanto, dará a estimativa da raiz que está mais próxima de  $x_2$ .

## 2.3.2 – O Método de Müller

Uma vez que  $x_3$  esteja determinada, o processo é repetido. Isso traz à tona a questão de qual ponto descartar. Duas estratégias gerais são tipicamente usadas:

1. Se estiverem sendo localizadas apenas raízes reais, escolhem-se os dois pontos originais que estejam mais próximos da nova estimativa da raiz  $x_3$ .
2. Se estiverem sendo calculadas tanto raízes reais quanto complexas, é usada uma abordagem sequencial. Isto é, da mesma forma que no método da secante,  $x_1, x_2$  e  $x_3$  tomam o lugar de  $x_0, x_1$  e  $x_2$ .

## 2.3.2 – O Método de Müller

Pseudocode for Müller's method.

```
SUB Muller(xr, h, eps, maxit)
  x2 = xr
  x1 = xr + h*xr
  x0 = xr - h*xr
  DO
    iter = iter + 1
    h0 = x1 - x0
    h1 = x2 - x1
    d0 = (f(x1) - f(x0)) / h0
    d1 = (f(x2) - f(x1)) / h1
    a = (d1 - d0) / (h1 + h0)
    b = a*h1 + d1
    c = f(x2)
    rad = SQRT(b*b - 4*a*c)
```

```
  If |b+rad| > |b-rad| THEN
    den = b + rad
  ELSE
    den = b - rad
  END IF
  dxr = -2*c / den
  xr = x2 + dxr
  PRINT iter, xr
  IF (|dxr| < eps*xr OR iter >= maxit) EXIT
  x0 = x1
  x1 = x2
  x2 = xr
END DO
END Muller
```

## 2.3.2 – O Método de Müller

---

**Exemplo:** Utilize o método de Müller com chutes iniciais de  $x_0$ ,  $x_1$  e  $x_2$  de 4,5, 5,5 e 5, respectivamente, para determinar as raízes da seguinte equação.

$$f(x) = x^3 - 13x - 12$$

## 2.3.3 – O Método de Bairstow

- O método de Bairstow é uma abordagem iterativa vagamente relacionada a ambos os métodos, de Müller e de Newton-Raphson. Antes de uma descrição matemática da técnica, deve-se lembrar da forma fatorada do polinômio,

$$f_s(x) = (x + 1)(x - 4)(x - 5)(x - 3)(x - 2) \quad (42)$$

se o polinômio for dividido por um fator que não corresponda a uma raiz, o quociente será um polinômio de quarto grau. Entretanto, nesse caso, resultará um resto não nulo.

## 2.3.3 – O Método de Bairstow

• Com base no exposto, pode-se elaborar um algoritmo para determinar uma raiz de um polinômio:

- (1) Escolha um valor para a raiz  $x = t$
- (2) Divida o polinômio pelo fator  $x - t$
- (3) Determine se existe um resto. Se não existir, a escolha foi perfeita e a raiz é igual a  $t$ . Se existir um resto, a escolha pode ser sistematicamente ajustada e o procedimento ser repetido até que o resto desapareça e a raiz seja localizada.

Depois de conseguir isso, todo procedimento pode ser repetido para o quociente, para determinar uma outra raiz.

## 2.3.3 – O Método de Bairstow

- O método de Bairstow é, de modo geral, baseado nessa abordagem. Conseqüentemente, ele depende do processo matemático de dividir um polinômio por um fator. Recordando-se da deflação de polinômios, tem-se que a divisão sintética envolve a divisão de um polinômio por um fator  $x - t$ . Por exemplo, o polinômio geral:

$$f_n(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n \quad (43)$$

pode ser dividido pelo fator  $x - t$ . Para fornecer um segundo polinômio que tem um grau a menos,

## 2.3.3 – O Método de Bairstow

$$f_{n-1}(x) = b_1 + b_2x + b_3x^2 + \cdots + b_nx^{n-1} \quad (44)$$

com um resto  $R = b_0$ , onde os coeficientes podem ser calculados pela relação de recorrência

$$b_n = a_n$$

$$b_i = a_i + b_{i+1}t, \text{ para } i = n - 1 \text{ a } 0$$

observa-se que, se  $t$  fosse uma raiz do polinômio original, o resto  $b_0$  seria igual a zero.

## 2.3.3 – O Método de Bairstow

Para permitir o cálculo de raízes complexas, o método de Bairstow divide o polinômio por um fator quadrático  $x^2 - rx - s$ . Isso é feito para a equação (43) e o resultado é um novo polinômio

$$f_{n-2}(x) = b_2 + b_3x + b_4x^2 + \cdots + b_nx^{n-2}$$

com um resto

$$R = b_1(x - r) + b_0 \tag{45}$$

como no caso da divisão sintética normal, uma relação de recorrência simples pode ser usada para fazer a divisão pelo fator quadrático:

## 2.3.3 – O Método de Bairstow

$$b_n = a_n \quad (46)$$

$$b_{n-1} = a_{n-1} + r b_n \quad (47)$$

$$b_i = a_i + r b_{i+1} + s b_{i+2} \text{ para } i = n - 2 \text{ a } 0 \quad (48)$$

O fator quadrático é introduzido para permitir a determinação de raízes complexas. Isso se relaciona ao fato de que, se os coeficientes do polinômio original forem reais, as raízes complexas ocorrem em pares conjugados.

Uma inspeção da equação (45) leva a concluir que, para o resto ser nulo,  $b_0$  e  $b_1$  precisam ser nulos. Como é pouco provável que as aproximações iniciais dos valores de  $r$  e  $s$  levem a esse resultado. Deve-se determinar uma maneira sistemática de modificar as aproximações de modo que  $b_0$  e  $b_1$  se aproximem de zero. Para implementar o método de Bairstow usa uma estratégia parecida com a abordagem podem ser expandidos usando-se uma série de Taylor como em:

## 2.3.3 – O Método de Bairstow

$$\begin{aligned} b_1(r + \Delta r, s + \Delta s) &= b_1 + \frac{\partial b_1}{\partial r} \Delta r + \frac{\partial b_1}{\partial s} \Delta s \\ b_0(r + \Delta r, s + \Delta s) &= b_0 + \frac{\partial b_0}{\partial r} \Delta r + \frac{\partial b_0}{\partial s} \Delta s \end{aligned} \quad (49)$$

As variações,  $\Delta r$  e  $\Delta s$ , necessárias para melhorar nossas aproximações podem ser estimadas igualando-se a equação (49) a zero para obter:

$$\frac{\partial b_1}{\partial r} \Delta r + \frac{\partial b_1}{\partial s} \Delta s = -b_1 \quad (50)$$

$$\frac{\partial b_0}{\partial r} \Delta r + \frac{\partial b_0}{\partial s} \Delta s = -b_0 \quad (51)$$

## 2.3.3 – O Método de Bairstow

Se as derivadas parciais dos  $b$ 's puderem ser determinadas, elas são um sistema de duas equações que podem ser resolvidas simultaneamente para as duas incógnitas,  $\Delta r$  e  $\Delta s$ . Bairstow mostrou que as derivadas parciais podem ser obtidas por uma divisão sintética dos  $b$ 's de um modo análogo à maneira como os próprios  $b$ 's foram determinados:

$$c_n = b_n \quad (52)$$

$$c_{n-1} = b_{n-1} + r c_n \quad (53)$$

$$c_i = b_i + r c_{i+1} + s c_{i+2} \text{ para } i = n - 2 \text{ a } 0 \quad (54)$$

onde  $\partial b_0 / \partial r = c_1$ ,  $\partial b_0 / \partial s = \partial b_1 / \partial r = c_2$  e  $\partial b_1 / \partial s = c_3$ . Portanto, as derivadas parciais são obtidas por uma divisão sintética dos  $b$ 's. A seguir, as derivadas parciais podem ser substituídas nas equações (50) e (51) junto com os  $b$ 's para fornecer:

$$c_2 \Delta r + c_3 \Delta s = -b_1$$

$$c_1 \Delta r + c_2 \Delta s = -b_0$$

## 2.3.3 – O Método de Bairstow

Essas equações podem ser resolvidas para determinar  $\Delta r$  e  $\Delta s$ , os quais, por sua vez, podem ser usados para melhorar as aproximações iniciais de  $r$  e  $s$ .

Em cada passo, pode-se fazer uma estimativa de um erro em  $r$  e um em  $s$ , como em:

$$|\varepsilon_{a,r}| = \left| \frac{\Delta r}{r} \right| 100\% \quad (55)$$

$$|\varepsilon_{a,s}| = \left| \frac{\Delta s}{s} \right| 100\% \quad (56)$$

Quando ambas as estimativas de erro caírem abaixo de um critério de parada pré-especificado  $\varepsilon_s$ , os valores das raízes podem ser determinados por:

$$x = \frac{r \pm \sqrt{r^2 + 4s}}{2} \quad (57)$$

## 2.3.3 – O Método de Bairstow

Nesse ponto, há três possibilidades:

- (1) O quociente é um polinômio de grau igual ou maior a três. Nesse caso, o método de Bairstow seria aplicado ao quociente para calcular novos valores de  $r$  e  $s$ . Os valores anteriores de  $r$  e  $s$  podem servir como aproximações iniciais nessa aplicação.
- (2) O quociente é um polinômio quadrático. Nesse caso, as duas raízes restantes poderiam ser calculadas diretamente com a equação (50).
- (3) O quociente é um polinômio de grau um. Nesse caso, a única raiz restante pode ser calculada simplesmente por:

$$x = -\frac{s}{r} \quad (58)$$

## 2.3.3 – O Método de Bairstow

### (a) Bairstow Algorithm

```
SUB Bairstow (a,nn,es,rr,ss,maxit,re,im,ier)
DIMENSION b(nn), c(nn)
r = rr; s = ss; n = nn
ier = 0; ea1 = 1; ea2 = 1
DO
  IF n < 3 OR iter ≥ maxit EXIT
  iter = 0
DO
  iter = iter + 1
  b(n) = a(n)
  b(n - 1) = a(n - 1) + r * b(n)
  c(n) = b(n)
  c(n - 1) = b(n - 1) + r * c(n)
DO i = n - 2, 0, -1
  b(i) = a(i) + r * b(i + 1) + s * b(i + 2)
  c(i) = b(i) + r * c(i + 1) + s * c(i + 2)
END DO
det = c(2) * c(2) - c(3) * c(1)
IF det ≠ 0 THEN
  dr = (-b(1) * c(2) + b(0) * c(3))/det
  ds = (-b(0) * c(2) + b(1) * c(1))/det
  r = r + dr
  s = s + ds
  IF r ≠ 0 THEN ea1 = ABS(dr/r) * 100
  IF s ≠ 0 THEN ea2 = ABS(ds/s) * 100
ELSE
  r = r + 1
  s = s + 1
  iter = 0
END IF
IF ea1 ≤ es AND ea2 ≤ es OR iter ≥ maxit EXIT
END DO
CALL Quadroot(r,s,r1,i1,r2,i2)
re(n) = r1
im(n) = i1
re(n - 1) = r2
im(n - 1) = i2
END DO
END Bairstow
```

```
r = r + dr
s = s + ds
IF r ≠ 0 THEN ea1 = ABS(dr/r) * 100
IF s ≠ 0 THEN ea2 = ABS(ds/s) * 100
ELSE
  r = r + 1
  s = s + 1
  iter = 0
END IF
IF ea1 ≤ es AND ea2 ≤ es OR iter ≥ maxit EXIT
END DO
CALL Quadroot(r,s,r1,i1,r2,i2)
re(n) = r1
im(n) = i1
re(n - 1) = r2
im(n - 1) = i2
n = n - 2
DO i = 0, n
  a(i) = b(i + 2)
END DO
END DO
END Bairstow
```

```
IF iter < maxit THEN
  IF n = 2 THEN
    r = -a(1)/a(2)
    s = -a(0)/a(2)
    CALL Quadroot(r,s,r1,i1,r2,i2)
    re(n) = r1
    im(n) = i1
    re(n - 1) = r2
    im(n - 1) = i2
  ELSE
    re(n) = -a(0)/a(1)
    im(n) = 0
  END IF
  ier = 1
END IF
END Bairstow
```

### (b) Roots of Quadratic Algorithm

```
SUB Quadroot(r,s,r1,i1,r2,i2)
disc = r ^ 2 + 4 * s
IF disc > 0 THEN
  r1 = (r + SQRT(disc))/2
  r2 = (r - SQRT(disc))/2
  i1 = 0
  i2 = 0
ELSE
  r1 = r/2
  r2 = r1
  i1 = SQRT(ABS(disc))/2
  i2 = -i1
END IF
END QuadRoot
```

## 2.3.3 – O Método de Bairstow

**Exemplo:** Utilize o método de Bairstow para determinar as raízes do polinômio:

$$f_5(x) = x^5 - 3,5x^4 + 2,75x^3 + 2,125x^2 - 3,875x + 1,25$$

Utilize como chutes iniciais  $r = s = -1$  e itere até um nível que  $\varepsilon_s = 1\%$