

MÉTODOS COMPUTACIONAIS

DR. MARCOS NAPOLEÃO RABELO

DR. WANDERLEI M. PEREIRA JUNIOR

Sistemas não lineares: Newton - Raphson

Grupo de Pesquisa e Estudos em Engenharia (GPÉE)



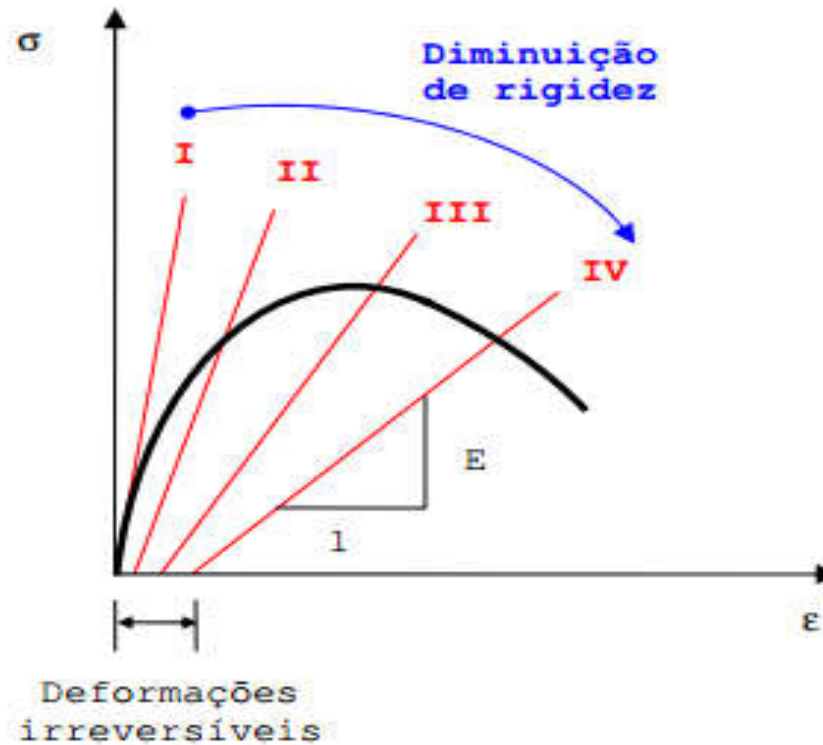
MODELOS NÃO LINEARES

Ao contrário do tópico anterior imaginemos agora que as equações que formam o nosso sistema não têm características lineares, ou seja, as variáveis que formam estas equações não são de grau 1.

Tais equações tem grandes aplicações no campo das ciências exatas e tecnológicas. Por exemplo a modelagem do comportamento do concreto é um exemplo característico de uma resposta não linear.

$$\begin{cases} f_1(x_1, x_2, \dots, x_{n-1}, x_n) = 0 \\ f_2(x_1, x_2, \dots, x_{n-1}, x_n) = 0 \\ \vdots \\ f_n(x_1, x_2, \dots, x_{n-1}, x_n) = 0 \end{cases} \quad (1)$$

Figura 1 – Comportamento não linear do concreto [1].



RELEMEBRANDO O MÉTODO DE NEWTON-RAPHSON PARA UMA EQUAÇÃO

O método de Newton-Raphson (ou Método de Newton) é um modelo de método computacional baseado em informações da derivada da função à qual deseja-se determinar a solução.

Primeiro precisamos discutir a importância de empregar informações da derivada, logo falaremos a sua interpretação geométrica.

A primeira derivada pode ser interpretada como uma medida de coeficiente de variação angular de uma reta tangente que passa por um determinado ponto P de uma função f . Para que se tenha validade a seguinte definição é necessário que f seja contínua em P .

Para que f seja contínua em P devem ser válidas as seguintes afirmações:

- Existe $f(a)$;
- Existe $\lim_{x \rightarrow a} f(x)$;
- $\lim_{x \rightarrow a} f(x) = f(a)$.

Imaginemos que uma **função $f(x)$** pode ser **escrita como uma série de Taylor** no ponto x_1 , ou seja, possui o seguinte formato da equação (2):

$$f(x) = \sum_{n=0}^{\infty} a_n \cdot (x - a)^n \quad (2)$$

$$a_n = f^{(n)}(a) / n! \quad (3)$$

Portanto, se $f(x)$ no ponto x_1 pode ser escrito no formato de uma **série polinomial de Taylor conforme equação (4)**. Neste caso a série foi expandida até o termo de ordem 2:

$$f(x) \approx f(x_1) + f'(x_1) \frac{(x - x_1)^1}{1!} + f''(a) \frac{(x - x_1)^2}{2!} \quad (4)$$

Admitindo que o termo de f'' é muito pequeno e que a $f(x)$ possui dentro de um intervalo $[x_1, x_2]$ uma raiz, ou seja, $f(x) = 0$. Logo podemos reescrever a equação (4):

$$0 \approx f(x_1) + f'(x_1) \cdot (x - x_1) \quad (5)$$

$$\frac{-f(x_1)}{f'(x_1)} \approx x - x_1 \quad (6)$$

$$x \approx x_1 - \frac{f(x_1)}{f'(x_1)} \quad (7)$$

A equação (7) pode ser reescrita como um processo recursivo, chamado aqui de Newton-Raphson.

$$x^{t+1} = x^t - \frac{f(x^t)}{f'(x^t)} \quad (8)$$

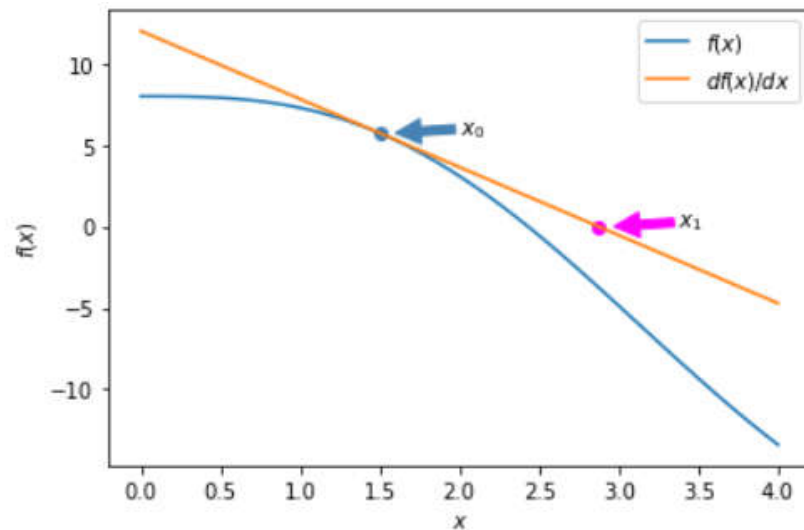
O erro de cada iteração para esse método é dado pela avaliação do novo ponto médio conforme eq's (9) e (10) [2]:

$$erro = |x^{t+1} - x^t| \quad (9)$$

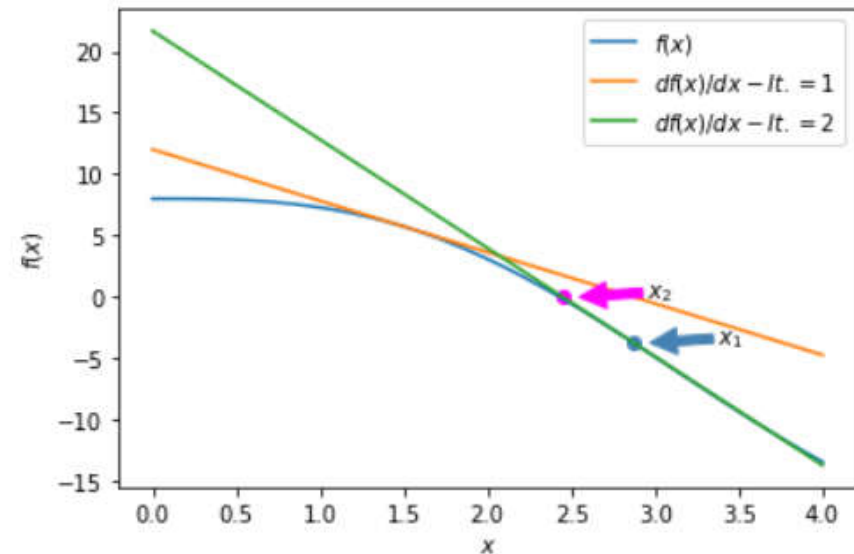
$$erro = 0 \quad \text{Para a condição que } f(x^{t+1}) = 0 \quad (10)$$

O movimento gráfico do método de Newton-Raphson (NR) pode ser visto na Figura 2.

Figura 2 – Movimento gráfico do método de NR para função $f(x) = 8 - 4,50 \cdot (x - \text{sen}x)$ no intervalo 0 e 4 [3].



Iteração $t = 1$

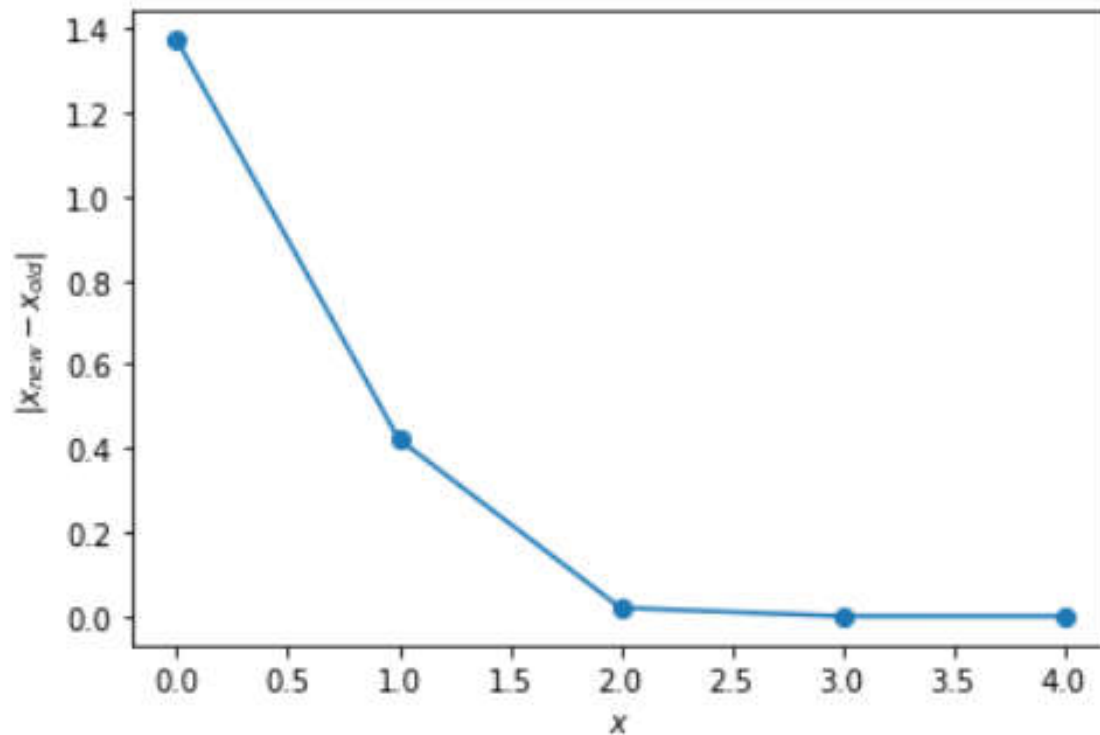


Iteração $t = 2$


```
1      ERRO = 100, TOL = 1E-2, X = X_INI
2      while ERRO > TOL:
3          Avalie  $f(x)$  e  $f'(x)$ 
4           $X = \text{eq. 8}$ 
5          ERRO = eq's 9 e 10
6      RAIZ = X
```

	$f(x)$	$\text{diff}(f(x))$	x	erro
0	5.738727e+00	-4.181683	2.872349	1.372349e+00
1	-3.728559e+00	-8.837875	2.450465	4.218840e-01
2	-1.587621e-01	-7.967374	2.430538	1.992653e-02
3	-5.740125e-04	-7.909534	2.430466	7.257222e-05
4	-7.734007e-09	-7.909321	2.430466	9.778347e-10

Iterações



Convergência

O MÉTODO DE NEWTON-RAPHSON PARA UM SISTEMA DE EQUAÇÕES

Utilizando o mesmo conceito visto anteriormente agora vamos estabelecer uma maneira de determinar as derivadas do sistema de equações. Esta maneira será feita utilizando o conceito de Jacobiano do sistema de equações, dado conforme equação (11).

$$J = \begin{bmatrix} \partial f_1 / x_1 & \dots & \partial f_1 / x_n \\ \vdots & \ddots & \vdots \\ \partial f_n / x_1 & \dots & \partial f_n / x_n \end{bmatrix} \quad (11)$$

Agora reescreveremos a equação (8) para a forma matricial, onde o vetor \mathbf{s} armazena a direção de busca do novo conjunto solução \mathbf{x}^{t+1} :

$$\mathbf{x}^{t+1} = \mathbf{x}^t + \mathbf{s} \quad (12)$$

$$\mathbf{s} = -\mathbf{J} \cdot \mathbf{f}(\mathbf{x}^t) \quad (13)$$

O processo iterativo só finaliza quando o critério de parada é satisfeito. Este critério é disposto na equação (14):

$$\frac{\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|_{\infty}}{\|\mathbf{x}^{(k+1)}\|_{\infty}} \leq tol, tol > 0 \quad (14)$$

Exemplo 1.1 [4,5]: Determinar o conjunto solução dos sistemas apresentados a seguir. Considere uma tolerância de 10^{-3} como critério de parada das iterações.

$$\begin{cases} x_1^2 + x_1 \cdot x_2 = 10 \\ x_2 + 3 \cdot x_1 \cdot x_2^2 = 57 \end{cases} \quad \mathbf{x}^t = [1,50 \quad 3,50] \quad \text{a)}$$

$$\begin{cases} x_1^3 + 2 \cdot x_2 + x_3 = 4 \\ 2 \cdot x_1^2 + x_2^2 - 4 \cdot x_3 = -1 \\ 3 \cdot x_1^2 + 4 \cdot x_2 + x_3 = 0 \end{cases} \quad \mathbf{x}^t = [1 \quad 1 \quad 2] \quad \text{b)}$$

Solução do exemplo a)

Determinação da Jacobiana da função: Vamos usar a biblioteca [SymPy](#).

$$\begin{bmatrix} 2 \cdot x_1 + x_2 & x_1 \\ 3 \cdot x_2^2 & 1 + 6 \cdot x_1 \cdot x_2 \end{bmatrix}$$

Iteração $t = 1$ aplicaremos o valor de $\mathbf{x}^1 = [1,50 \quad 3,50]$

$$J(\mathbf{x}^1) = \begin{bmatrix} 2 \cdot 1,50 + 3,50 & 1,50 \\ 3 \cdot 3,50^2 & 1 + 6 \cdot 1,50 \cdot 3,50 \end{bmatrix} = \begin{bmatrix} 6,50 & 1,50 \\ 36,75 & 32,50 \end{bmatrix}$$

$$J(\mathbf{x}^1)^{-1} = \begin{bmatrix} 0.20816653 & -0.00960769 \\ -0.23538831 & 0.04163331 \end{bmatrix}$$

$$f(\mathbf{x}^1) = \begin{bmatrix} 1,50^2 + 1,50 \cdot 3,50 - 10 \\ 3,50 + 3 \cdot 1,50 \cdot 3,50^2 - 57 \end{bmatrix} = \begin{bmatrix} -2,50 \\ 1,625 \end{bmatrix}$$

$$\mathbf{s} = -J(\mathbf{x}^1)^{-1} \cdot f(\mathbf{x}^1) = \begin{bmatrix} 0,5360 \\ -0,6561 \end{bmatrix}$$

$$\mathbf{x}^2 = \begin{bmatrix} 1,50 \\ 3,50 \end{bmatrix} + \begin{bmatrix} 0,5360 \\ -0,6561 \end{bmatrix} = \begin{bmatrix} 2,0360 \\ 2,8439 \end{bmatrix}$$

REFERÊNCIAS

- [1] Faglioni AF [UNESP. Análise não-linear física de vigas de concreto armado utilizando o elemento finito prismático regular linear associado ao de barra. Mestrado em Engenharia Civil. Universidade Estadual Paulista Júlio de Mesquita Filho (UNESP), 2006.
- [2] Burden RL, Faires JD. Numerical analysis. 9. ed., International ed. Belmont, Calif.: Brooks/Cole; 2011.
- [3] Gilat A, Suramanian V. Métodos numéricos para engenheiros e cientistas: uma introdução com aplicações usando o MATLAB. 2008.
- [4] Chapra SC. Métodos Numéricos Aplicados com MATLAB® para Engenheiros e Cientistas. 3ª edição. AMGH; 2013.

- [5] Vitorino A, Ruggiero MAG. Algebra Linear e Aplicações 20--.
https://www.ime.unicamp.br/~marcia/AlgebraLinear/sistemas_triangulares.html
(accessed August 23, 2021).