Matemática - RC/UFG

Laboratório de Simulação Matemática

Parte 8[2]

Prof. Thiago Alves de Queiroz

2/2017

[2][Cap. 10 e 11] BURDEN, R. L.; FAIRES, J. D. Numerical Analysis (9th ed). Cengage Learning, 2010.

Sistema de Equações Não Lineares

- Resolver um sistema de equações não lineares é um problema que tem sido evitado, sendo preferível resolver um sistema linear.
- Um sistema de equações não lineares tem a forma:

$$\begin{aligned} f_1(x_1, x_2, \ldots, x_n) &= 0, \\ f_2(x_1, x_2, \ldots, x_n) &= 0, \\ \vdots \qquad & \qquad \vdots \\ f_n(x_1, x_2, \ldots, x_n) &= 0, \end{aligned}$$

em que cada função f_i representa um mapeamento do vetor $x = (x_1, x_2, \ldots, x_n)$ do $\mathbb{R}^n$ em um número real no $\mathbb{R}$.

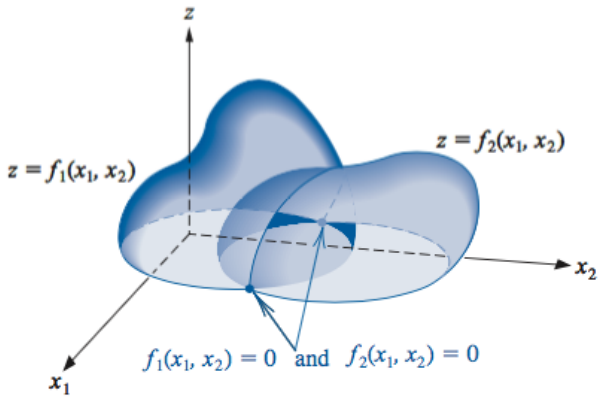

Figura: Representação geométrica de sistema linear com 2 funções.

Sistema de Equações Não Lineares

- Um sistema de n equações não lineares em n variáveis pode ser representado por uma função F que mapeia do $\mathbb{R}^n$ para o $\mathbb{R}^n$ como:

$$F(x_1, x_2, \ldots, x_n) = (f_1(x_1, x_2, \ldots, x_n), f_2(x_1, x_2, \ldots, x_n), \ldots, f_n(x_1, x_2, \ldots, x_n)) \tag{1}$$

- Ao usar a notação de vetor para representar $x_1, x_2, \ldots, x_n$, o sistema pode ser representado na forma $F(x) = 0$.
- As funções $f_1, f_2, \ldots, f_n$ são chamadas de *funções coordenadas* de F.
- **Exemplo.** Represente o sistema linear abaixo na forma $F(x) = 0$.

$$\begin{aligned} 3x_1 - \cos(x_2 x_3) - \frac{1}{2} &= 0, \\ x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1.06 &= 0, \\ e^{-x_1 x_2} + 20x_3 + \frac{10\pi - 3}{3} &= 0 \end{aligned}$$

Exemplo

- **Resposta.** Define-se as três funções coordenadas f_1, f_2 e f_3 do $\mathbb{R}^3$ para $\mathbb{R}$ como:

$$
\begin{aligned}
f_1(x_1, x_2, x_3) &= 3x_1 - \cos(x_2 x_3) - \frac{1}{2}, \\
f_2(x_1, x_2, x_3) &= x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1.06, \\
f_3(x_1, x_2, x_3) &= e^{-x_1 x_2} + 20x_3 + \frac{10\pi - 3}{3},
\end{aligned}
$$

- Assim, define-se F do $\mathbb{R}^3$ para o $\mathbb{R}^3$ como:

$$
\begin{aligned}
\mathbf{F}(\mathbf{x}) &= \mathbf{F}(x_1, x_2, x_3) \\
&= (f_1(x_1, x_2, x_3), f_2(x_1, x_2, x_3), f_3(x_1, x_2, x_3))^t \\
&= \left(3x_1 - \cos(x_2 x_3) - \frac{1}{2}, x_1^2 - 81(x_2 + 0.1)^2 \right. \\
&\quad \left. + \sin x_3 + 1.06, e^{-x_1 x_2} + 20x_3 + \frac{10\pi - 3}{3}\right)^t.
\end{aligned}
$$

Sistema de Equações Não Lineares

- **Definição.** Seja a função F definida em $D \subset \mathbb{R}^n$ para o $\mathbb{R}^n$ da forma $F(x) = (f_1(x), f_2(x), \ldots, f_n(x))$, sendo f_i o mapeamento de $\mathbb{R}^n$ em $\mathbb{R}$ para cada i. Define-se

$$\lim_{x \to x_0} F(x) = L = (L_1, L_2, \ldots, L_n), \tag{2}$$

  se e somente se $\lim_{x \to x_0} f_i(x) = L_i$, para cada $i = 1, 2, \ldots, n$.
- A função F é contínua em $x_0 \in D$ dado que $\lim_{x \to x_0} F(x)$ existe e $\lim_{x \to x_0} F(x) = F(x_0)$.
- A função F é contínua em um conjunto D se F é contínua em cada $x \in D$, sendo expresso por $F \in C(D)$.
- **Definição.** Uma função G definida em $D \subset \mathbb{R}^n$ para o $\mathbb{R}^n$ tem um *ponto fixo* em $p \in D$ se $G(p) = p$.

Sistema de Equações Não Lineares

- **Teorema.** Seja $D = \{(x_1, x_2, \ldots, x_n) | a_i \leq x_i \leq b_i,$ para cada $i = 1, 2, \ldots, n\}$ para alguma coleção de constantes $a_1, a_2, \ldots, a_n$ e $b_1, b_2, \ldots, b_n$. Suponha que G seja uma função contínua em $D \subset \mathbb{R}^n$ para $\mathbb{R}^n$ com a propriedade que $G(x) \in D$ para qualquer $x \in D$. Então, G tem um ponto fixo em D.
- **Exemplo.** Escreva o sistema de equações não lineares do exemplo anterior na forma de ponto fixo $G(x) = x$. Aplique uma iteração de ponto fixo partindo do ponto inicial $x^{(0)} = (0,1; 0,1; -0,1)$ e calcule o erro na norma l_∞.
- **Resposta.** Inicialmente, resolve-se a i-ésima equação para x_i, isto é:

$$x_1 = \frac{1}{3}\cos(x_2 x_3) + \frac{1}{6},$$

$$x_2 = \frac{1}{9}\sqrt{x_1^2 + \sin x_3 + 1.06} - 0.1,$$

$$x_3 = -\frac{1}{20}e^{-x_1 x_2} - \frac{10\pi - 3}{60}.$$

Exemplo

- Para aproximar o ponto fixo p, usa-se o ponto inicial dado $x^{(0)}$. Assim, a sequência de vetores são gerados por:

$$x_1^{(k)} = \frac{1}{3}\cos x_2^{(k-1)} x_3^{(k-1)} + \frac{1}{6},$$

$$x_2^{(k)} = \frac{1}{9}\sqrt{\left(x_1^{(k-1)}\right)^2 + \sin x_3^{(k-1)} + 1.06} - 0.1,$$

$$x_3^{(k)} = -\frac{1}{20}e^{-x_1^{(k-1)} x_2^{(k-1)}} - \frac{10\pi - 3}{60}$$

- A convergência na norma l_∞ se dá calculando o erro absoluto $\|x^{(k)} - x^{(k-1)}\|_\infty$.
- Assim, a solução para $k = 1$ é $x^{(1)} = (0,49998333; 0,00944115; -0.52310127)$, com erro absoluto igual a $0,423$.
- Uma forma de acelerar a convergência da iteração de ponto fixo, faz-se igual ao método de Gauss-Seidel.

Método de Newton

- A construção de um algoritmo que conduz a um método de ponto fixo no caso n-dimensional envolve uma matriz $A(x)$, de ordem $n \times n$, em que cada entrada $a_{ij}(x)$ é uma função que leva do $\mathbb{R}^n$ para o $\mathbb{R}$.
- Com isso, tem-se que encontrar uma matriz não singular $A(x)$ que resolve o sistema $F(x) = 0$ tal que:

$$G(x) = x - A(x)^{-1}F(x) \tag{3}$$

- Uma escolha apropriada para $A(x)$ é a matriz Jacobiana $J(x)$, dada por:

$$J(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}) & \frac{\partial f_1}{\partial x_2}(\mathbf{x}) & \cdots & \frac{\partial f_1}{\partial x_n}(\mathbf{x}) \\ \frac{\partial f_2}{\partial x_1}(\mathbf{x}) & \frac{\partial f_2}{\partial x_2}(\mathbf{x}) & \cdots & \frac{\partial f_2}{\partial x_n}(\mathbf{x}) \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1}(\mathbf{x}) & \frac{\partial f_n}{\partial x_2}(\mathbf{x}) & \cdots & \frac{\partial f_n}{\partial x_n}(\mathbf{x}) \end{bmatrix}.$$

Método de Newton

- Segue que a função G é definida por:

$$G(x) = x - J(x)^{-1}F(x). \tag{4}$$

- O procedimento iterativo para achar a solução do sistema $F(x)$ envolve, partir de $x^{(0)}$, gerar, para $k \geq 1$, a sequência:

$$x^{(k)} = G(x^{(k-1)}) - J(x^{(k-1)})^{-1}F(x^{(k-1)}). \tag{5}$$

- Esse procedimento é chamado de *método de Newton para sistemas não lineares*, que geralmente converge em ordem quadrática.
- Uma fraqueza no método de Newton surge a partir do cálculo da inversa da matriz $J(x)$ a cada iteração.
- Um forma de contornar essa situação é encontrar um vetor y, tal que $J(x^{(k-1)})y = -F(x^{(k-1)})$. Em seguida, a aproximação para $x^{(k)}$ é obtida adicionado y em $x^{(k-1)}$.

Exemplo

- **Exemplo**. Escreva para o sistema não linear abaixo a função $F(x)$ e a matriz Jacobiana $J(x)$.

$$3x_1 - \cos(x_2x_3) - \frac{1}{2} = 0,$$
$$x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1.06 = 0,$$
$$e^{-x_1x_2} + 20x_3 + \frac{10\pi - 3}{3} = 0$$

- **Resposta**. Define-se a função $F(x_1, x_2, x_3) = (f_1, f_2, f_3)$:

$$f_1(x_1, x_2, x_3) = 3x_1 - \cos(x_2x_3) - \frac{1}{2},$$
$$f_2(x_1, x_2, x_3) = x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1.06,$$

$$f_3(x_1, x_2, x_3) = e^{-x_1x_2} + 20x_3 + \frac{10\pi - 3}{3}.$$

Exemplo

- A matriz Jacobiana $J(x)$ é dada por:

$$J(x_1, x_2, x_3) = \begin{bmatrix} 3 & x_3 \sin x_2 x_3 & x_2 \sin x_2 x_3 \\ 2x_1 & -162(x_2 + 0.1) & \cos x_3 \\ -x_2 e^{-x_1 x_2} & -x_1 e^{-x_1 x_2} & 20 \end{bmatrix}$$

- Assim, no k-ésimo passo resolve-se $J(x^{(k-1)})y^{(k-1)} = -F(x^{(k-1)})$:

$$J\left(\mathbf{x}^{(k-1)}\right) = \begin{bmatrix} 3 & x_3^{(k-1)} \sin x_2^{(k-1)} x_3^{(k-1)} & x_2^{(k-1)} \sin x_2^{(k-1)} x_3^{(k-1)} \\ 2x_1^{(k-1)} & -162\left(x_2^{(k-1)} + 0.1\right) & \cos x_3^{(k-1)} \\ -x_2^{(k-1)} e^{-x_1^{(k-1)} x_2^{(k-1)}} & -x_1^{(k-1)} e^{-x_1^{(k-1)} x_2^{(k-1)}} & 20 \end{bmatrix},$$

$$\mathbf{y}^{(k-1)} = \begin{bmatrix} y_1^{(k-1)} \\ y_2^{(k-1)} \\ y_3^{(k-1)} \end{bmatrix},$$

ıd

$$\mathbf{F}\left(\mathbf{x}^{(k-1)}\right) = \begin{bmatrix} 3x_1^{(k-1)} - \cos x_2^{(k-1)} x_3^{(k-1)} - \frac{1}{2} \\ \left(x_1^{(k-1)}\right)^2 - 81\left(x_2^{(k-1)} + 0.1\right)^2 + \sin x_3^{(k-1)} + 1.06 \end{bmatrix}.$$

Exemplo

- Segue que o cálculo de $x^{(k)}$ é então:

$$\begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ x_3^{(k)} \end{bmatrix} = \begin{bmatrix} x_1^{(k-1)} \\ x_2^{(k-1)} \\ x_3^{(k-1)} \end{bmatrix} + \begin{bmatrix} y_1^{(k-1)} \\ y_2^{(k-1)} \\ y_3^{(k-1)} \end{bmatrix},$$

- Ao considerar $x^{(0)} = (0,1; 0,1; -0,1)$, obtém-se: $F(x^0) = (-0,199995; -2,269833417; 8,462025346)$ e

$$J(\mathbf{x}^{(0)}) = \begin{bmatrix} 3 & 9.999833334 \times 10^{-4} & 9.999833334 \times 10^{-4} \\ 0.2 & -32.4 & 0.9950041653 \\ -0.09900498337 & -0.09900498337 & 20 \end{bmatrix}$$

- A resolução do sistema linear $J(x^{(0)})y^{(0)} = -F(x^0)$ fornece:

$$\mathbf{y}^{(0)} = \begin{bmatrix} 0.3998696728 \\ -0.08053315147 \\ -0.4215204718 \end{bmatrix} \quad \text{and} \quad \mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \mathbf{y}^{(0)} = \begin{bmatrix} 0.4998696782 \\ 0.01946684853 \\ -0.5215204718 \end{bmatrix}.$$

Método de Newton para Sistemas Não Lineares

INPUT number n of equations and unknowns; initial approximation $\mathbf{x} = (x_1, \ldots, x_n)^t$, tolerance TOL; maximum number of iterations N.
OUTPUT approximate solution $\mathbf{x} = (x_1, \ldots, x_n)^t$ or a message that the number of iterations was exceeded.

Step 1 Set $k = 1$.

Step 2 While $(k \leq N)$ do Steps 3–7.

Step 3 Calculate $\mathbf{F}(\mathbf{x})$ and $J(\mathbf{x})$, where $J(\mathbf{x})_{i,j} = (\partial f_i(\mathbf{x})/\partial x_j)$ for $1 \leq i, j \leq n$.

Step 4 Solve the $n \times n$ linear system $J(\mathbf{x})\mathbf{y} = -\mathbf{F}(\mathbf{x})$.

Step 5 Set $\mathbf{x} = \mathbf{x} + \mathbf{y}$.

Step 6 If $||\mathbf{y}|| < TOL$ then OUTPUT $(\mathbf{x})$;
(*The procedure was successful.*)
STOP.

Step 7 Set $k = k + 1$.

Step 8 OUTPUT ('Maximum number of iterations exceeded');
(*The procedure was unsuccessful.*)
STOP.

Método Quase-Newton

- Uma fraqueza do método de Newton é determinar a matriz Jacobiana e resolver o sistema linear de ordem $n \times n$ que envolve essa matriz.
- Uma forma de aliviar no cálculo da matriz Jacobiana é usar a aproximação por diferenças finitas para o cálculo das derivadas parciais:

$$\frac{\partial f_j(x^{(i)})}{\partial x_k} \approx \frac{f_j(x^{(i)} + e_k h) - f_j(x^{(i)})}{h}, \tag{6}$$

  em que h é um valor bem pequeno e e_k é um vetor com entradas nulas exceto na k-ésima posição que recebe o valor 1.
- Outra alternativa ao método de Newton é uma generalização ao método da Secante para sistemas não lineares, chamado de *método de Broyden*.
- O método de Broyden é chamado de *quase-Newton*, de forma que a matriz Jacobina é substituída por uma matriz aproximada mais fácil de ser calculada.

Método Quase-Newton

- A matriz Jacobiana do método de Newton é, então, substituída, para obter $x^{(i+1)}$, por:

$$A_i = A_{i-1} + \frac{y_i - A_{i-1}s_i}{\|s_i\|_2^2}s_i, \tag{7}$$

em que $y_i = F(x^{(i)}) - F(x^{(i-1)})$ e $s_i = x^{(i)} - x^{(i-1)}$.

- Com isso, segue que $x^{(i+1)} = x^{(i)} - A_i^{-1}F(x^{(i)})$.
- A resolução do sistema linear ainda continua por meio de:

$$A_i s_{i+1} = -F(x^i). \tag{8}$$

- Lembrar que na iteração $k = 0$, tem-se $A_0 = J(x^{(0)})$.;
- Outra forma de calcular a inversa de A_i é usando a *fórmula de Sherman-Morrison*:

$$A_i^{-1} = A_{i-1}^{-1} + \frac{(s_i - A_{i-1}^{-1}y_i)s_i A_{i-1}^{-1}}{s_i A_{i-1}^{-1}y_i}. \tag{9}$$

Método de Broyden

INPUT number n of equations and unknowns; initial approximation $\mathbf{x} = (x_1, \ldots, x_n)^t$; tolerance TOL; maximum number of iterations N.

OUTPUT approximate solution $\mathbf{x} = (x_1, \ldots, x_n)^t$ or a message that the number of iterations was exceeded.

Step 1 Set $A_0 = J(\mathbf{x})$ where $J(\mathbf{x})_{i,j} = \frac{\partial f_i}{\partial x_j}(\mathbf{x})$ for $1 \leq i, j \leq n$;

$\mathbf{v} = \mathbf{F}(\mathbf{x})$. (*Note:* $\mathbf{v} = \mathbf{F}(\mathbf{x}^{(0)})$.)

Step 2 Set $A = A_0^{-1}$. (*Use Gaussian elimination.*)

Step 3 Set $\mathbf{s} = -A\mathbf{v}$; (*Note:* $\mathbf{s} = \mathbf{s}_1$.)

$\mathbf{x} = \mathbf{x} + \mathbf{s}$; (*Note:* $\mathbf{x} = \mathbf{x}^{(1)}$.)

$k = 2$.

Step 4 While $(k \leq N)$ do Steps 5–13.

Step 5 Set $\mathbf{w} = \mathbf{v}$; (*Save* $\mathbf{v}$.)

$\mathbf{v} = \mathbf{F}(\mathbf{x})$; (*Note:* $\mathbf{v} = \mathbf{F}(\mathbf{x}^{(k)})$.)

$\mathbf{y} = \mathbf{v} - \mathbf{w}$. (*Note:* $\mathbf{y} = \mathbf{y}_k$.)

Step 6 Set $\mathbf{z} = -A\mathbf{y}$. (*Note:* $\mathbf{z} = -A_{k-1}^{-1}\mathbf{y}_k$.)

Método de Broyden

Step 7 Set $p = -\mathbf{s}^t\mathbf{z}$. $\left(\textit{Note: } p = \mathbf{s}_k^t A_{k-1}^{-1}\mathbf{y}_k.\right)$

Step 8 Set $\mathbf{u}^t = \mathbf{s}^t A$.

Step 9 Set $A = A + \frac{1}{p}(\mathbf{s} + \mathbf{z})\mathbf{u}^t$. $\left(\textit{Note: } A = A_k^{-1}.\right)$

Step 10 Set $\mathbf{s} = -A\mathbf{v}$. $\left(\textit{Note: } \mathbf{s} = -A_k^{-1}\mathbf{F}(\mathbf{x}^{(k)}).\right)$

Step 11 Set $\mathbf{x} = \mathbf{x} + \mathbf{s}$. $\left(\textit{Note: } \mathbf{x} = \mathbf{x}^{(k+1)}.\right)$

Step 12 If $||\mathbf{s}|| < TOL$ then OUTPUT $(\mathbf{x})$;
$\left(\textit{The procedure was successful.}\right)$
STOP.

Step 13 Set $k = k + 1$.

Step 14 OUTPUT ('Maximum number of iterations exceeded');
(*The procedure was unsuccessful.*)
STOP.