

Modelagem Computacional

Aula 8²

Prof. Thiago Alves de Queiroz

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:

$$f_1(x_1, x_2, \dots, x_n) = 0,$$

$$f_2(x_1, x_2, \dots, x_n) = 0,$$

$$\vdots \quad \quad \quad \vdots$$

$$f_n(x_1, x_2, \dots, x_n) = 0,$$

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

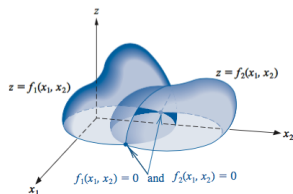


Figura: Representação geométrica de um sistema com 2 funções.

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:

$$F(x_1, x_2, \dots, x_n) = (f_1(x_1, x_2, \dots, x_n), f_2(x_1, x_2, \dots, x_n), \dots, f_n(x_1, x_2, \dots, x_n)) \quad (1)$$

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

$$3x_1 - \cos(x_2x_3) - \frac{1}{2} = 0,$$

$$x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1.06 = 0,$$

$$e^{-x_1x_2} + 20x_3 + \frac{10\pi - 3}{3} = 0$$

Exemplo

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

$$f_1(x_1, x_2, x_3) = 3x_1 - \cos(x_2x_3) - \frac{1}{2},$$

$$f_2(x_1, x_2, x_3) = x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1.06,$$

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

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

$$\begin{aligned} \mathbf{F}(\mathbf{x}) &= \mathbf{F}(x_1, x_2, x_3) \\ &= (f_1(x_1, x_2, x_3), f_2(x_1, x_2, x_3), f_3(x_1, x_2, x_3))^t \\ &= \left(3x_1 - \cos(x_2x_3) - \frac{1}{2}, x_1^2 - 81(x_2 + 0.1)^2 \right. \\ &\quad \left. + \sin x_3 + 1.06, e^{-x_1x_2} + 20x_3 + \frac{10\pi - 3}{3} \right)^t. \end{aligned}$$

Sistema de Equações Não Lineares

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

$$\lim_{x \rightarrow x_0} F(x) = L = (L_1, L_2, \dots, L_n), \quad (2)$$

se e somente se $\lim_{x \rightarrow x_0} f_i(x) = L_i$, para cada $i = 1, 2, \dots, n$.

- A função F é contínua em $x_0 \in D$ dado que $\lim_{x \rightarrow x_0} F(x)$ existe e $\lim_{x \rightarrow x_0} F(x) = F(x_0)$.
- 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

$D = \{(x_1, x_2, \dots, x_n) \mid a_i \leq x_i \leq b_i, \text{ para cada } i = 1, 2, \dots, n\}$ para alguma coleção de constantes a_1, a_2, \dots, a_n e b_1, b_2, \dots, 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 $\|x^{(k)} - x^{(k-1)}\|_\infty$.
- 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:

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

- 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:

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

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

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

- Esse procedimento é chamado de *método de Newton para sistemas não lineares*, que geralmente converge em ordem quadrática.
- Uma dificuldade 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 $J(x^{(k-1)})y^{(k-1)} = -F_X^{(k-1)}$:

$$J(x^{(k-1)}) = \begin{bmatrix} 3 & x_3^{(k-1)} \sin x_2^{(k-1)} x_3^{(k-1)} & x_2^{(k-1)} \sin x_2^{(k-1)} x_3^{(k-1)} \\ 2x_1^{(k-1)} & -162(x_2^{(k-1)} + 0.1) & \cos x_3^{(k-1)} \\ -x_2^{(k-1)} e^{-x_1^{(k-1)} x_2^{(k-1)}} & -x_1^{(k-1)} e^{-x_1^{(k-1)} x_2^{(k-1)}} & 20 \end{bmatrix},$$
$$y^{(k-1)} = \begin{bmatrix} y_1^{(k-1)} \\ y_2^{(k-1)} \\ y_3^{(k-1)} \end{bmatrix},$$

id

$$F(x^{(k-1)}) = \begin{bmatrix} 3x_1^{(k-1)} - \cos x_2^{(k-1)} x_3^{(k-1)} - \frac{1}{2} \\ (x_1^{(k-1)})^2 - 81(x_2^{(k-1)} + 0.1)^2 + \sin x_3^{(k-1)} + 1.06 \\ e^{-x_1^{(k-1)} x_2^{(k-1)}} + 20x_3^{(k-1)} + \frac{10\pi-3}{3} \end{bmatrix}.$$

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

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

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

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

Método de Newton para Sistemas Não Lineares

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

OUTPUT approximate solution $\mathbf{x} = (x_1, \dots, x_n)^t$ 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 dificuldade do método de Newton é determinar a matriz Jacobiana e resolver o sistema 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:

$$\frac{\partial f_j(\mathbf{x}^{(i)})}{\partial x_k} \approx \frac{f_j(\mathbf{x}^{(i)} + \mathbf{e}_k h) - f_j(\mathbf{x}^{(i)})}{h}, \quad (6)$$

em que h é um valor bem pequeno e \mathbf{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 Jacobiana é substituída por uma matriz aproximada.

Método Quase-Newton

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

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

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:

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

- 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*:

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

Método de Broyden

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

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

Step 1 Set $A_0 = J(\mathbf{x})$ where $J(\mathbf{x})_{ij} = \frac{\partial f_i}{\partial x_j}(\mathbf{x})$ 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 $p = -s^t z$. (Note: $p = s_k^t A_{k-1}^{-1} y_k$.)

Step 8 Set $u^t = s^t A$.

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

Step 10 Set $s = -Av$. (Note: $s = -A_k^{-1} F(x^{(k)})$.)

Step 11 Set $x = x + s$. (Note: $x = x^{(k+1)}$.)

Step 12 If $\|s\| < TOL$ then OUTPUT (x);
(The procedure was successful.)
STOP.

Step 13 Set $k = k + 1$.

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

Problemas de Valor de Contorno

- Problemas de valor de contorno são definidos por equações diferenciais com condições impostas em diferentes pontos, não apenas no ponto inicial.
- Para equações diferenciais de primeira ordem, apenas uma condição é imposta, de forma que não há distinção com um problema de valor inicial.

- **Teorema.** Suponha que a função f no problema de valor de contorno de segunda ordem:

$y'' = f(x, y, y')$, para $a \leq x \leq b$, com $y(a) = \alpha$ e $y(b) = \beta$,
seja contínua no conjunto:

$D = \{(x, y, y') \mid \text{para } a \leq x \leq b, \text{ com } -\infty < y < \infty \text{ e } -\infty < y' < \infty\}$,
e que as derivadas parciais f_y e f_x sejam contínuas em D . Se:

- ▶ $f_y(x, y, y') > 0$ para todo $(x, y, y') \in D$ e
- ▶ existe uma constante M com $|f_{y'}(x, y, y')| \leq M$, para todo $(x, y, y') \in D$,

então o problema de valor de contorno tem solução única.

Exemplo

- **Exemplo.** Verifique que o problema de valor de contorno abaixo tem solução única.

$$y'' - e^{-xy} + \sin(y') = 0, \text{ para } 1 \leq x \leq 2, \text{ com } y(1) = y(2) = 0.$$

- **Resposta.** Tem-se que $f(x, y, y') = e^{-xy} - \sin(y')$ para todo $x \in [1, 2]$.
- Pelo Teorema anterior, $f_y(x, y, y') = xe^{-xy} > 0$ e $|f_{y'}(x, y, y')| = |-\cos(y')| \leq 1$.
- Logo, o problema tem solução única, uma vez que as duas condições foram atendidas, a segunda com $M = 1$.
- A equação diferencial $y'' = f(x, y, y')$ é linear quando as funções $p(x)$, $q(x)$ e $r(x)$ existem satisfazendo:

$$f(x, y, y') = p(x)y' + q(x)y + r(x). \quad (10)$$

Problema de Valor de Contorno

- **Corolário.** Seja o seguinte problema de valor de contorno linear: $y'' = p(x)y' + q(x)y + r(x)$, para $a \leq x \leq b$, com $y(a) = \alpha$ e $y(b) = \beta$, que satisfaz:
 - ▶ $p(x)$, $q(x)$ e $r(x)$ são contínuas em $[a, b]$ e
 - ▶ $q(x) > 0$ em $[a, b]$.

Então, o problema de valor de contorno tem solução única.

- Para fazer a aproximação, geram-se dois problemas de valor inicial a partir do problema de valor de contorno, isto é:

$$y'' = p(x)y' + q(x)y + r(x), \text{ para } a \leq x \leq b, \text{ com } y(a) = \alpha \text{ e } y'(a) = 0, \quad (11)$$

$$y'' = p(x)y' + q(x)y, \text{ para } a \leq x \leq b, \text{ com } y(a) = 0 \text{ e } y'(a) = 1, \quad (12)$$

Problema de Valor de Contorno

- Seja $y_1(x)$ a solução do primeiro problema de valor inicial na eq. (11) e $y_2(x)$ a solução do segundo problema de valor inicial na eq. (12).
- Então, define-se a solução do problema de valor de contorno como:

$$y(x) = y_1(x) + \frac{\beta - y_1(b)}{y_2(b)} y_2(x). \quad (13)$$

- Os passos anteriores em substituir o problema de valor de contorno em dois problemas de valor inicial consiste no *método do Tiro Linear*.
- Para a aproximação das soluções $y_1(x)$ e $y_2(x)$ dos problemas de valor inicial, usam-se métodos numéricos conhecidos, como o Runge-Kutta de quarta ordem.

Método do Tiro Linear

INPUT endpoints a, b ; boundary conditions α, β ; number of subintervals N .

OUTPUT approximations $w_{1,i}$ to $y(x_i)$; $w_{2,i}$ to $y'(x_i)$ for each $i = 0, 1, \dots, N$.

Step 1 Set $h = (b - a)/N$;

$$u_{1,0} = \alpha;$$

$$u_{2,0} = 0;$$

$$v_{1,0} = 0;$$

$$v_{2,0} = 1.$$

Step 2 For $i = 0, \dots, N - 1$ do Steps 3 and 4.

(The Runge-Kutta method for systems is used in Steps 3 and 4.)

Step 3 Set $x = a + ih$.

Step 4 Set $k_{1,1} = hu_{2,i}$;

$$k_{1,2} = h [p(x)u_{2,i} + q(x)u_{1,i} + r(x)];$$

$$k_{2,1} = h [u_{2,i} + \frac{1}{2}k_{1,2}];$$

$$k_{2,2} = h [p(x + h/2) (u_{2,i} + \frac{1}{2}k_{1,2}) \\ + q(x + h/2) (u_{1,i} + \frac{1}{2}k_{1,1}) + r(x + h/2)];$$

$$k_{3,1} = h [u_{2,i} + \frac{1}{2}k_{2,2}];$$

$$k_{3,2} = h [p(x + h/2) (u_{2,i} + \frac{1}{2}k_{2,2}) \\ + q(x + h/2)(u_{1,i} + \frac{1}{2}k_{2,1}) + r(x + h/2)];$$

Método do Tiro Linear

$$k_{4,1} = h [u_{2,i} + k_{3,2}];$$

$$k_{4,2} = h [p(x+h)(u_{2,i} + k_{3,2}) + q(x+h)(u_{1,i} + k_{3,1}) + r(x+h)];$$

$$u_{1,i+1} = u_{1,i} + \frac{1}{6} [k_{1,1} + 2k_{2,1} + 2k_{3,1} + k_{4,1}];$$

$$u_{2,i+1} = u_{2,i} + \frac{1}{6} [k_{1,2} + 2k_{2,2} + 2k_{3,2} + k_{4,2}];$$

$$k'_{1,1} = hv_{2,i};$$

$$k'_{1,2} = h [p(x)v_{2,i} + q(x)v_{1,i}];$$

$$k'_{2,1} = h [v_{2,i} + \frac{1}{2}k'_{1,2}];$$

$$k'_{2,2} = h [p(x+h/2)(v_{2,i} + \frac{1}{2}k'_{1,2}) + q(x+h/2)(v_{1,i} + \frac{1}{2}k'_{1,1})];$$

$$k'_{3,1} = h [v_{2,i} + \frac{1}{2}k'_{2,2}];$$

$$k'_{3,2} = h [p(x+h/2)(v_{2,i} + \frac{1}{2}k'_{2,2}) + q(x+h/2)(v_{1,i} + \frac{1}{2}k'_{2,1})];$$

$$k'_{4,1} = h [v_{2,i} + k'_{3,2}];$$

$$k'_{4,2} = h [p(x+h)(v_{2,i} + k'_{3,2}) + q(x+h)(v_{1,i} + k'_{3,1})];$$

Método do Tiro Linear

$$v_{1,i+1} = v_{1,i} + \frac{1}{6} [k'_{1,1} + 2k'_{2,1} + 2k'_{3,1} + k'_{4,1}];$$

$$v_{2,i+1} = v_{2,i} + \frac{1}{6} [k'_{1,2} + 2k'_{2,2} + 2k'_{3,2} + k'_{4,2}].$$

Step 5 Set $w_{1,0} = \alpha$;

$$w_{2,0} = \frac{\beta - u_{1,N}}{v_{1,N}};$$

OUTPUT ($a, w_{1,0}, w_{2,0}$).

Step 6 For $i = 1, \dots, N$

$$\text{set } W1 = u_{1,i} + w_{2,0}v_{1,i};$$

$$W2 = u_{2,i} + w_{2,0}v_{2,i};$$

$$x = a + ih;$$

OUTPUT ($x, W1, W2$). (*Output is $x_i, w_{1,i}, w_{2,i}$.*)

Step 7 STOP. (*The process is complete.*)

Problema de Valor de Contorno Não Linear

- O método do Tiro Linear para problemas de valor de contorno de segunda ordem não lineares é similar ao método para o caso linear.
- A diferença é que o problema de valor de contorno agora é dividido em uma sequência de problemas de valor inicial envolvendo um parâmetro t , isto é:

$$y'' = f(x, y, y'), \text{ para } a \leq x \leq b, \text{ com } y(a) = \alpha \text{ e } y'(a) = t. \quad (14)$$

- A escolha do parâmetro $t = t_k$ é feita de forma a garantir que:

$$\lim_{k \rightarrow \infty} y(b, t_k) = y(b) = \beta, \quad (15)$$

- $y(x, t_k)$ denota a solução do problema de valor inicial na eq. (14) com $t = t_k$ e $y(x)$ é a solução do problema do valor de contorno não linear.

Problema de Valor de Contorno Não Linear

- O método inicia com um parâmetro t_0 que determina a elevação necessária para que $y(b, t_0)$ aproxime-se de β . Caso contrário, escolhe-se novos parâmetros t_1, t_2, \dots
- Para determinar os parâmetros t_k , para $k = 1, 2, \dots$, usa-se o método de Newton sabendo da aproximação inicial t_0 , isto é:

$$t_k = t_{k-1} - \frac{y(b, t_{k-1}) - \beta}{\frac{dy}{dt}(b, t_{k-1})}, \quad (16)$$

que necessita conhecer $\frac{dy}{dt}(b, t_{k-1})$.

- Por esse motivo, rescreve-se o problema de valor inicial na eq. (14) como dependente de x e t :

$$y''(x, t) = f(x, y(x, t), y'(x, t)), \text{ para } a \leq x \leq b, \text{ com } y(a, t) = \alpha \text{ e } y'(a, t) = t. \quad (17)$$

Problema de Valor de Contorno Não Linear

- Outro problema de valor inicial surge, isto é:

$$z''(x, t) = \frac{\partial f}{\partial y}(x, y, y')z(x, t) + \frac{\partial f}{\partial y'}(x, y, y')z'(x, t), \quad (18)$$

para $a \leq x \leq b$, com $z(a, t) = 0$ e $z'(a, t) = 1$.

- Assim, o método de Newton requer que os dois problemas de valor inicial nas eqs. (17) e (18) sejam resolvidos para cada iteração, ou seja:

$$t_k = t_{k-1} - \frac{y(b, t_{k-1}) - \beta}{z(b, t_{k-1})}. \quad (19)$$

- Nenhum desses problemas de valor inicial são resolvidos exatamente, sendo preciso empregar algum método numérico, como o Runge-Kutta de quarta ordem.

Método do Tiro Não Linear

INPUT endpoints a, b ; boundary conditions α, β ; number of subintervals $N \geq 2$; tolerance TOL ; maximum number of iterations M .

OUTPUT approximations $w_{1,i}$ to $y(x_i)$; $w_{2,i}$ to $y'(x_i)$ for each $i = 0, 1, \dots, N$ or a message that the maximum number of iterations was exceeded.

Step 1 Set $h = (b - a)/N$;

$$k = 1;$$

$$TK = (\beta - \alpha)/(b - a). \quad (\text{Note: } TK \text{ could also be input.})$$

Step 2 While $(k \leq M)$ do Steps 3–10.

Step 3 Set $w_{1,0} = \alpha$;

$$w_{2,0} = TK;$$

$$u_1 = 0;$$

$$u_2 = 1.$$

Step 4 For $i = 1, \dots, N$ do Steps 5 and 6.

(The Runge-Kutta method for systems is used in Steps 5 and 6.)

Step 5 Set $x = a + (i - 1)h$.

Step 6 Set $k_{1,1} = hw_{2,i-1}$;

$$k_{1,2} = hf(x, w_{1,i-1}, w_{2,i-1});$$

$$k_{2,1} = h(w_{2,i-1} + \frac{1}{2}k_{1,2});$$

$$k_{2,2} = hf(x + h/2, w_{1,i-1} + \frac{1}{2}k_{1,1}, w_{2,i-1} + \frac{1}{2}k_{1,2});$$

$$k_{3,1} = h(w_{2,i-1} + \frac{1}{2}k_{2,2});$$

Método do Tiro Não Linear

$$k_{3,2} = hf(x + h/2, w_{1,i-1} + \frac{1}{2}k_{2,1}, w_{2,i-1} + \frac{1}{2}k_{2,2});$$

$$k_{4,1} = h(w_{2,i-1} + k_{3,2});$$

$$k_{4,2} = hf(x + h, w_{1,i-1} + k_{3,1}, w_{2,i-1} + k_{3,2});$$

$$w_{1,i} = w_{1,i-1} + (k_{1,1} + 2k_{2,1} + 2k_{3,1} + k_{4,1})/6;$$

$$w_{2,i} = w_{2,i-1} + (k_{1,2} + 2k_{2,2} + 2k_{3,2} + k_{4,2})/6;$$

$$k'_{1,1} = hu_2;$$

$$k'_{1,2} = h[f_y(x, w_{1,i-1}, w_{2,i-1})u_1 \\ + f_{y'}(x, w_{1,i-1}, w_{2,i-1})u_2];$$

$$k'_{2,1} = h[u_2 + \frac{1}{2}k'_{1,2}];$$

$$k'_{2,2} = h[f_y(x + h/2, w_{1,i-1}, w_{2,i-1})(u_1 + \frac{1}{2}k'_{1,1}) \\ + f_{y'}(x + h/2, w_{1,i-1}, w_{2,i-1})(u_2 + \frac{1}{2}k'_{1,2})];$$

$$k'_{3,1} = h(u_2 + \frac{1}{2}k'_{2,2});$$

$$k'_{3,2} = h[f_y(x + h/2, w_{1,i-1}, w_{2,i-1})(u_1 + \frac{1}{2}k'_{2,1}) \\ + f_{y'}(x + h/2, w_{1,i-1}, w_{2,i-1})(u_2 + \frac{1}{2}k'_{2,2})];$$

Método do Tiro Não Linear

$$k'_{4,1} = h(u_2 + k'_{3,2});$$

$$k'_{4,2} = h [f_y(x + h, w_{1,i-1}, w_{2,i-1}) (u_1 + k'_{3,1}) \\ + f_y(x + h, w_{1,i-1}, w_{2,i-1}) (u_2 + k'_{3,2})];$$

$$u_1 = u_1 + \frac{1}{6}[k'_{1,1} + 2k'_{2,1} + 2k'_{3,1} + k'_{4,1}];$$

$$u_2 = u_2 + \frac{1}{6}[k'_{1,2} + 2k'_{2,2} + 2k'_{3,2} + k'_{4,2}].$$

Step 7 If $|w_{1,N} - \beta| \leq TOL$ then do Steps 8 and 9.

Step 8 For $i = 0, 1, \dots, N$
set $x = a + ih$;
OUTPUT $(x, w_{1,i}, w_{2,i})$.

Step 9 (*The procedure is complete.*)
STOP.

Step 10 Set $TK = TK - \frac{w_{1,N} - \beta}{u_1}$;

(*Newton's method is used to compute TK.*)

$$k = k + 1.$$

Step 11 OUTPUT ('Maximum number of iterations exceeded');
(*The procedure was unsuccessful.*)
STOP.

Exemplo

- **Exemplo.** Aplique o método do tiro não linear para o problema de valor de contorno:

$$y'' = \frac{1}{8}(32 + 2x^3 - yy'), \text{ para } 1 \leq x \leq 3, \text{ com } y(1) = 17 \text{ e } y'(3) = \frac{43}{3}.$$

Considere $N = 20$, $M = 10$, $TOL = 10^{-5}$ e compare os resultados com a solução exata $y(x) = x^2 + \frac{16}{x}$.

- **Resposta.** Ao codificar o método do tiro não linear, aproxima-se, em cada iteração, os problemas de valor inicial:

$$y''(x, t) = \frac{1}{8}(32 + 2x^3 - yy'), \text{ para } 1 \leq x \leq 3, \text{ com } y(1) = 17 \text{ e } y'(1) = t_k, \quad (20)$$

$$z''(x, t) = \frac{\partial f}{\partial y}z + \frac{\partial f}{\partial y'}z' = -\frac{1}{8}(y'z + yz'), \text{ para } 1 \leq x \leq 3, \text{ com } z(1) = 0 \text{ e } z'(1) = 1. \quad (21)$$

- O critério de parada do algoritmo requer: $|w_{1,N}(t_k) - y(3)| \leq 10^{-5}$ chegando em $t_k = t_4 = -14,000203$.

Método de Diferenças Finitas

- Métodos que envolvem as diferenças finitas para resolver problemas de valor de contorno substituem cada uma das derivadas da equação diferencial por uma aproximação por quociente-diferença.
- O *método de Diferenças Finitas* para problemas de valor de contorno lineares de segunda ordem requer que as derivadas y' e y'' sejam aproximadas por quociente-diferenças.
- Inicialmente, o intervalo $[a, b]$ é dividido em $N + 1$ intervalos iguais de forma que os pontos sejam dados por $x_i = a + ih$, para $i = 0, 1, \dots, N + 1$ com $h = \frac{b-a}{N+1}$.
- Nos pontos x_i do interior, isto é, para $i = 1, 2, \dots, N$, a equação diferencial é aproximada por:

$$y''(x_i) = p(x_i)y'(x_i) + q(x_i)y(x_i) + r(x_i). \quad (22)$$

Método de Diferenças Finitas

- Ao expandir o termo y em x_i por um polinômio de Taylor de terceira ordem e avaliando em x_{i+1} e x_{i-1} , chega-se em:

$$y''(x_i) = \frac{1}{h} [y(x_{i+1}) - 2y(x_i) + y(x_{i-1})] - \frac{h^2}{12} y^{(4)}(\xi_i), \quad (23)$$

para algum $\xi_i \in (x_{i-1}, x_{i+1})$.

- Essa fórmula para $y''(x_i)$ é chamada de *fórmula de diferença centrada*.
- De maneira similar para obter a fórmula de diferença centrada para $y'(x_i)$, tem-se:

$$y'(x_i) = \frac{1}{2h} [y(x_{i+1}) - y(x_{i-1})] - \frac{h^2}{6} y'''(\eta_i), \quad (24)$$

para algum $\eta_i \in (x_{i-1}, x_{i+1})$.

- Assim, faz-se a substituição das fórmulas de $y''(x_i)$ e $y'(x_i)$ na eq. (22).

Método de Diferenças Finitas

- Segue que o método de Diferenças Finitas tem erro de truncamento da ordem $O(h^2)$.
- Aplicando as condições de contorno $y(a) = \alpha$ e $y(b) = \beta$ no seguinte sistema de equações lineares para $i = 1, 2, \dots, N$:

$$w_0 = \alpha, \quad w_{N+1} = \beta$$

$$\left(\frac{-w_{i+1} + 2w_i - w_{i-1}}{h^2} \right) + p(x_i) \left(\frac{w_{i+1} - w_{i-1}}{2h} \right) + q(x_i)w_i = -r(x_i).$$

- O sistema de equações resultantes pode ser expresso em uma matriz tridiagonal de ordem $N \times N$ na forma $Aw = b$, que possui solução única quando $q(x) > 0$ e $h < \frac{2}{L}$, com $L = \max_{a \leq x \leq b} |p'(x)|$.

Método de Diferenças Finitas

$$A = \begin{bmatrix} 2 + h^2q(x_1) & -1 + \frac{h}{2}p(x_1) & 0 & \dots & 0 \\ -1 - \frac{h}{2}p(x_2) & 2 + h^2q(x_2) & -1 + \frac{h}{2}p(x_2) & \dots & 0 \\ 0 & \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & -1 + \frac{h}{2}p(x_{N-1}) \\ 0 & \dots & 0 & -1 - \frac{h}{2}p(x_N) & 2 + h^2q(x_N) \end{bmatrix},$$

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{N-1} \\ w_N \end{bmatrix}, \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} -h^2r(x_1) + \left(1 + \frac{h}{2}p(x_1)\right)w_0 \\ -h^2r(x_2) \\ \vdots \\ -h^2r(x_{N-1}) \\ -h^2r(x_N) + \left(1 - \frac{h}{2}p(x_N)\right)w_{N+1} \end{bmatrix}.$$

Método de Diferenças Finitas

INPUT endpoints a, b ; boundary conditions α, β ; integer $N \geq 2$.

OUTPUT approximations w_i to $y(x_i)$ for each $i = 0, 1, \dots, N + 1$.

Step 1 Set $h = (b - a)/(N + 1)$;
 $x = a + h$;
 $a_1 = 2 + h^2q(x)$;
 $b_1 = -1 + (h/2)p(x)$;
 $d_1 = -h^2r(x) + (1 + (h/2)p(x))\alpha$.

Step 2 For $i = 2, \dots, N - 1$
set $x = a + ih$;
 $a_i = 2 + h^2q(x)$;
 $b_i = -1 + (h/2)p(x)$;
 $c_i = -1 - (h/2)p(x)$;
 $d_i = -h^2r(x)$.

Step 3 Set $x = b - h$;
 $a_N = 2 + h^2q(x)$;
 $c_N = -1 - (h/2)p(x)$;
 $d_N = -h^2r(x) + (1 - (h/2)p(x))\beta$.

Método de Diferenças Finitas

Step 4 Set $l_1 = a_1$; (Steps 4–8 solve a tridiagonal linear system using Algorithm 6.7.)

$$u_1 = b_1/a_1;$$

$$z_1 = d_1/l_1.$$

Step 5 For $i = 2, \dots, N - 1$ set $l_i = a_i - c_i u_{i-1}$;

$$u_i = b_i/l_i;$$

$$z_i = (d_i - c_i z_{i-1})/l_i.$$

Step 6 Set $l_N = a_N - c_N u_{N-1}$;

$$z_N = (d_N - c_N z_{N-1})/l_N.$$

Step 7 Set $w_0 = \alpha$;

$$w_{N+1} = \beta.$$

$$w_N = z_N.$$

Step 8 For $i = N - 1, \dots, 1$ set $w_i = z_i - u_i w_{i+1}$.

Step 9 For $i = 0, \dots, N + 1$ set $x = a + ih$;

OUTPUT (x, w_i) .

Step 10 STOP. (The procedure is complete.)