Aproximação por Spline

Prof. Doherty Andrade

www.metodosnumericos.com.br

1 Introdução

A natureza oscilatória dos polinômios de alto grau pode induzir a grandes erros no processo de aproximação por interpolação polinomial de uma função no intervalo $[a, b]$. Este fenômeno, conhecido como Runge, basicamente nos diz que, em se tratando de interpolação polinomial, o erro aumenta com o grau do polinômio interpolador e é maior nas extremidades a e b do intervalo do que na região central desse intervalo. Isso retringe muito o uso dessa técnica.

Um enfoque alternativo que traz vantagens é usar aproximação polinomial por partes de Hermite, para isso os valores de f e f' devem ser conhecidos nos pontos $x_0 < x_1 < x_2 < \ldots < x_n$. Assim, um polinômio cúbico de Hermite pode ser usado em cada subintervalo $[x_0, x_1], [x_1, x_2], [x_2, x_3], \ldots, [x_{n-1}, x_n]$ para obter uma função que tenha derivada contínua em $[x_0, x_n]$, mas nem sempre as informaçoes sobre f' são conhecidas.

Uma outra alternativa para a aproximação de funções arbitrárias em intervalos fechados é o uso de spline: aproximação polinomial por partes. A raiz da palavra spline é a mesma que a da palavra *splint* em inglês, que é uma régua fina e flexível de madeira ou metal que era usada para desenhar uma curva contínua e suave forçando-a a passar por pontos específicos.

Definição 1.1 *Uma função $S(x)$ é denominada spline de grau m com nós nos pontos $a = x_0 < x_1 < \cdots x_{n-1} < x_n = b$ se satisfaz:*
(a) em cada subintervalo $[x_{i-1}, x_i]$, $i = 1, 2, \ldots, n$, $S(x)$ é um polinômio $s_i(x)$ de grau m.
(b) $S(x)$ é contínua e tem derivadas contínuas até ordem $m - 1$ em $[a, b]$.

Definição 1.2 *Dizemos que uma spline $S(x)$ é interpoladora de f nos pontos $a = x_0 < x_1 < \cdots x_{n-1} < x_n = b$ se $S(x_i) = f(x_i), i = 0, 1, 2, \ldots, n$.*

Uma spline de grau um é denominada de spline linear. Neste caso, a função possui gráfico dado por uma sequência de segmentos de retas unindo os pontos determinados pelos nós. Veja o gráfico.

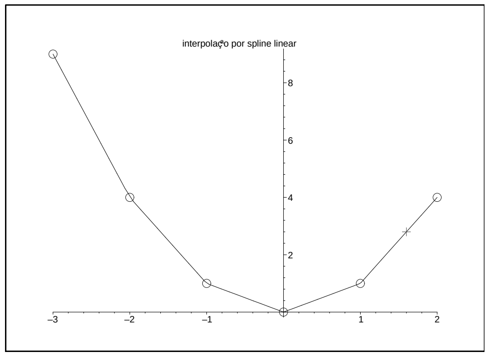

Figura 1: Spline linear

A função spline linear interpolante de $f(x)$, facilmente deduzida da equação da reta, pode ser escrita em cada subintervalo $[x_{i-1}, x_i]$, $i = 1, 2, \ldots, n$ como

$$S_i(x) = f(x_{i-1})\frac{x_i - x}{x_i - x_{i-1}} + f(x_i)\frac{x - x_{i-1}}{x_i - x_{i-1}}, x \in [x_{i-1}, x_i]. \tag{1}$$

• **Exemplo 1.3** *Considere a função f tabelada por*

x_i	*1*	*2*	*5*	*7*
y_i	*1.0*	*2.0*	*3.0*	*2.5*

Usando (1) temos os seguintes segmentos de reta que compõem a spline linear:

$$S_1(x) = f(x_0)\frac{x_1 - x}{x_1 - x_0} + f(x_1)\frac{x - x_0}{x_1 - x_0} = x, x \in [1, 2]; \tag{2}$$

$$S_2(x) = f(x_1)\frac{x_2 - x}{x_2 - x_1} + f(x_2)\frac{x - x_1}{x_2 - x_1} = \frac{1}{3}(x + 4), x \in [2, 5]; \tag{3}$$

$$S_3(x) = f(x_2)\frac{x_3 - x}{x_3 - x_2} + f(x_3)\frac{x - x_2}{x_3 - x_2} = \frac{1}{2}(-0.5x + 8.5), x \in [5, 7]. \tag{4}$$

A spline linear tem a desvantagem de ter a primeira derivada descontínua.

O tipo mais simples de função polinomial por partes diferenciável em todo um intervalo $[a = x_0, b = x_n]$ é aquele obtido ajustando um polinômio quadrático entre cada par sucessivo de nós de modo que suas derivadas coincidam em pontos extremos dos intervalos. A spline quadrática tem apenas a derivada até ordem 1 contínua e portanto a curvatura pode variar muito nos nós. Por esta razão, as splines cúbicas são as mais utilizadas.

1.1 Spline cúbica

Considere uma função $f(x)$ tabelada nos pontos $a = x_0 < x_1 < \ldots < x_n = b$. Uma spline cúbica interpolante $S_3(x)$ é uma função polinomial por partes contínua, onde cada parte $s_k(x)$ é um polinômio de grau 3 no intervalo $[x_{k-1}, x_k]$, $k = 1, 2, \ldots, n$.

Como $s_k(x)$ é um polinômio de grau 3, podemos supor que tenha a forma

$$s_k(x) = a_k(x - x_k)^3 + b_k(x - x_k)^2 + c_k(x - x_k) + d_k, k = 1, 2, \ldots, n.$$

Como $S_3(x)$ tem a primeira e a segunda derivadas contínuas, a spline cúbica interpolante deve satisfazer:

1. $S_3(x) = s_k(x)$, para todo $x \in [x_{k-1}, x_k]$ e $k = 1, 2, \ldots, n$.

2. $S_3(x_i) = f(x_i)$, $i = 0, 1, 2, \ldots, n$ (condições de interpolação).

3. $s_k(x_k) = s_{k+1}(x_k)$, $k = 1, 2, \ldots, n-1$.

4. $s'_k(x_k) = s'_{k+1}(x_k)$, $k = 1, 2, \ldots, n-1$.

5. $s''_k(x_k) = s''_{k+1}(x_k)$, $k = 1, 2, \ldots, n-1$.

6. Condições de bordo, de acordo com o interesse, geralmente são:

   (a) $S''_3(x_0) = g_0 = 0$ e $S''_3(x_n) = g_n = 0$, que dá origem à chamada spline **natural**, obrigando que a spline seja aproximadamente linear nos extremos do intervalo.

   (b) $g_0 = g_1$ e $g_n = g_{n-1}$, obrigando que a spline seja aproximadamente parábola nos extremos.

   (c) Impor outros valores para as inclinações em cada extremo: $S'_3(x_0) = A$ e $S'_3(x_n) = B$.

Vamos deduzir a fórmula da spline cúbica natural, isto é, utilizando as condições de 1 a 6a.
Para simplificar a notação vamos denotar h_k por

$$h_k = x_k - x_{k-1}, k = 1, 2, \ldots, n.$$

Da condição 2, temos

$$s_k(x_k) = d_k = f(x_k), k = 1, 2, \ldots, n. \tag{1}$$

e

$$s_1(x_0) = -a_1 h_1^3 + b_1 h_1^2 - c_1 h_1 + d_1 = f(x_0). \tag{2}$$

Da condição 3, temos para $k = 1, 2, \ldots, n-1$ temos

$$-a_{k+1} h_{k+1}^3 + b_{k+1} h_{k+1}^2 - c_{k+1} h_{k+1} + d_{k+1} = f(x_k). \tag{3}$$

As condições 4 e 5 utilizam as derivadas de $s_k(x)$:

$$s'_k(x) = 3a_k(x - x_k)^2 + 2b_k(x - x_k) + c_k \tag{4}$$

$$s''_k(x) = 6a_k(x - x_k)^2 + 2b_k. \tag{5}$$

Como $s''_k(x_k) = 2b_k$, segue que

$$b_k = \frac{s''_k(x_k)}{2}. \tag{6}$$

Do mesmo modo, como $s''_k(x_{k-1}) = -6a_k h_k + 2b_k$, podemos escrever a_k:

$$\begin{aligned} a_k &= \frac{2b_k - s''_k(x_{k-1})}{6h_k} \text{ e substituindo } b_k \\ a_k &= \frac{s''_k(x_k) - s''_k(x_{k-1})}{6h_k}. \end{aligned}$$

Impondo agora a condição 5: $s''_k(x_{k-1}) = s''_{k-1}(x_{k-1})$ podemos reescrever a_k:

$$a_k = \frac{s''_k(x_k) - s''_{k-1}(x_{k-1})}{6h_k}. \tag{7}$$

No caso $k=1$, impomos uma variável arbitrária $s''_0(x_0)$ e $k=n$, impomos uma variável arbitrária $s''_n(x_n)$.

Assim, já expressamos $d_k = f(x_k)$ dado em (1), a_k dado em (7) e b_k dado em (6). Podemos usar (2) e (3) para determinar c_k.

Da equação (2) podemos obter c_1:

$$c_1 = \frac{-a_1 h_1^3 + b_1 h_1^2 + d_1 - f(x_0)}{h_1}$$

e em geral, de (3) obtemos $c_k, k = 2, 3, \ldots, n-1$.

$$\begin{aligned}
c_k &= \frac{-a_k h_k^3 + b_k h_k^2 + d_k - f(x_{k-1})}{h_k} \\
&= \frac{f(x_k) - f(x_{k-1})}{h_k} - \left(a_k h_k^2 - b_k h_k \right) \\
&= \frac{f(x_k) - f(x_{k-1})}{h_k} - \left\{ \frac{s''_k(x_k) - s''_k(x_{k-1})}{6} h_k - \frac{s''_k(x_k)}{2} h_k \right\}.
\end{aligned}$$

Ou seja,

$$\begin{aligned}
c_k &= \frac{f(x_k) - f(x_{k-1})}{h_k} - \left\{ \frac{s''_k(x_k)h_k - s''_k(x_{k-1})h_k - 3s''_k(x_k)h_k}{6} \right\} \\
c_k &= \frac{f(x_k) - f(x_{k-1})}{h_k} + \frac{2s''_k(x_k)h_k + s''_{k-1}(x_{k-1})h_k}{6}.
\end{aligned}$$

Resumindo:

$$c_k = \frac{f(x_k) - f(x_{k-1})}{h_k} + \frac{2s''_k(x_k)h_k + s''_{k-1}(x_{k-1})h_k}{6}. \tag{8}$$

Vamos denotar $f(x_k) = y_k$ e $s''(x_k) = g_k$ e assim temos:

De (7),

$$a_k = \frac{g_k - g_{k-1}}{6h_k}. \tag{9}$$

De (6),

$$b_k = \frac{g_k}{2}. \tag{10}$$

De (8),

$$c_k = \frac{y_k - y_{k-1}}{h_k} + \frac{2h_k g_k + g_{k-1}h_k}{6}. \tag{11}$$

De (1),

$$d_k = f(x_k) = y_k. \tag{12}$$

Assim, o polinômio cúbico fica:

$$s_k(x) = \underbrace{\frac{g_k - g_{k-1}}{6h_k}}_{a_k}(x - x_k)^3 + \underbrace{\frac{g_k}{2}}_{b_k}(x - x_k)^2 + \underbrace{\left[\frac{y_k - y_{k-1}}{h_k} + \frac{2h_k g_k + g_{k-1}h_k}{6}\right]}_{c_k}(x - x_k) + \underbrace{y_k}_{d_k},$$

$k = 1, 2, \ldots, n$, com coeficientes dependendo apenas de g_k.

Impondo agora a condição 4 que ainda não foi utilizada: $s'_k(x_k) = s'_{k+1}(x_k)$, temos

$$s'_k(x_k) = c_k = 3a_{k+1}h_{k+1}^2 - 2b_{k+1}h_{k+1} + c_{k+1},$$

donde

$$c_{k+1} = c_k - 3a_{k+1}h_{k+1}^2 + 2b_{k+1}h_{k+1}$$

e usando (9),(10) e (11) obtemos

$$\frac{y_{k+1} - y_k}{h_{k+1}} + \frac{2h_{k+1}g_{k+1} + g_k h_{k+1}}{6} = \frac{y_k - y_{k-1}}{h_k} + \frac{2h_k g_k + g_{k-1}h_k}{6} - 3\left(\frac{g_{k+1} - g_k}{6}\right)h_{k+1} + 2\left(\frac{g_{k+1}}{2}\right)h_{k+1}.$$

Agrupando os termos semelhantes:

$$\frac{1}{6}\left[h_k g_{k-1} + (2h_k + 3h_{k+1} - h_{k-1})g_k + (6h_{k+1} - 3h_{k+1} - 2h_{k+1})g_{k+1}\right] = \frac{y_{k+1} - y_k}{h_{k+1}} - \frac{y_k - y_{k-1}}{h_k}.$$

Ou seja,

$$h_k g_{k-1} + 2(h_k + h_{k+1})g_k + h_{k+1}g_{k+1} = 6\left(\frac{y_{k+1} - y_k}{h_{k+1}} - \frac{y_k - y_{k-1}}{h_k}\right), k = 1, 2, \ldots, n-1. \tag{13}$$

O sistema (13) é um sistema tridiagonal de equações lineares $Ax = b$ com $n-1$ equações e $n+1$ incógnitas $g_0, g_1, \ldots, g_n$. Portanto, indeterminado, onde

$$A = \begin{bmatrix} h_1 & 2(h_1 + h_2) & h_2 & 0 & \cdots & \cdots & 0 \\ 0 & h_2 & 2(h_2 + h_3) & h_3 & \cdots & \cdots & 0 \\ 0 & 0 & h_3 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & h_{n-1} & 2(h_{n-1} - h_n) & h_n \end{bmatrix}_{(n-1)\times(n+1)}$$

e

$$b = 6\begin{bmatrix} \dfrac{y_2 - y_1}{h_2} - \dfrac{y_1 - y_0}{h_1} \\ \frac{y_3 - y_2}{h_3} - \dfrac{y_2 - y_1}{h_2} \\ \vdots \\ \dfrac{y_n - y_{n-1}}{h_n} - \dfrac{y_{n-1} - y_{n-2}}{h_{n-1}} \end{bmatrix}_{(n-1)\times 1}$$

Temos que impor mais duas condições, são as condições de spline natural:

$$S_3''(x_0) = g_0 = 0 = S_3''(x_n) = g_n = 0.$$

Nosso sistema de equações lineares $Ax = b$ passa agora a $A'x' = b'$ tendo $n-1$ equações e $n-1$ incógnitas $g_1, g_2, \ldots, g_{n-1}$ que compõem o vetor $x' = (g_1, \ldots, g_{n-1})$ e

$$A' = \begin{bmatrix} 2(h_1 + h_2) & h_2 & 0 & 0 & \cdots & 0 \\ h_2 & 2(h_2 + h_3) & h_3 & 0 & \cdots & 0 \\ 0 & h_3 & 2(h_3 + h_4) & h_4 & \cdots & \cdots \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & \cdots & & \cdots & h_{n-1} & 2(h_{n-1} + h_n) \end{bmatrix}_{(n-1)\times(n-1)}$$

e

$$b' = 6 \begin{bmatrix} \dfrac{y_2 - y_1}{h_2} - \dfrac{y_1 - y_0}{h_1} \\ \frac{y_3-y_2}{h_3} - \dfrac{y_2 - y_1}{h_2} \\ \vdots \\ \dfrac{y_n - y_{n-1}}{h_n} - \dfrac{y_{n-1} - y_{n-2}}{h_{n-1}} \end{bmatrix}_{(n-1)\times 1}.$$

Como A' é estritamente diagonalmente dominante (também é simétrica e positiva definida), essa matriz tem inversa e portanto o sistema tem solução única. Com isso concluímos a demonstração do seguinte teorema.

Teorema 1.4 *Se f for definida em $a = x_0 < x_1 < \cdots < x_n = b$, então f tem uma única spline cúbica interpolante natural S que satisfaz às condições de contorno $S''(a) = S''(b) = 0$.*

Seja $S(x)$ a spline cúbica natural, i. e., $S(x)$ satisfaz $S''(x_0) = S''(x_n) = 0$. Pode-se mostrar que se $R(x)$ é outra spline cúbica que é duas vezes continuamente diferenciável então,

$$\int_a^b S(x)dx \leq \int_a^b R(x)dx,$$

assim a spline cúbica natural tem curvatura mínima.

É comum usar a notação

$$e_i = 6\left(\frac{y_{i+1} - y_i}{h_i}\right).$$

Assim, o sistema $A'x' = b'$ pode ser escrito como

$$\begin{bmatrix} 2(h_1+h_2) & h_2 & 0 & 0 & \cdots & 0 \\ h_2 & 2(h_2+h_3) & h_3 & 0 & \cdots & 0 \\ 0 & h_3 & 2(h_3+h_4) & h_4 & \cdots & \cdots \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & \cdots & & \cdots & h_{n-1} & 2(h_{n-1}+h_n) \end{bmatrix} \cdot \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_{n-1} \end{bmatrix} = \begin{bmatrix} e_1 - e_0 \\ e_2 - e_1 \\ \vdots \\ e_{n-1} - e_{n-2} \end{bmatrix}.$$

Usando a condição 6b para fronteira fixa e mais uma hipótese sobre diferenciabilidade sobre a f, obtemos também a existência e unicidade da spline cúbica. Veja o teorema a seguir.

Teorema 1.5 *Se $f : [a,b] \to \mathbb{R}$ e $a = x_0 < x_1 < \ldots < x_n = b$, e diferenciável em $x = a$ e em $x = b$, então existe uma única spline cúbica interpolante S para f nos nós $x_0 < x_1 < \ldots < x_n$ que satisfaz $S'(a) = f'(a)$ e $S'(b) = f'(b)$.*

A demonstração desse resultado segue as mesmas ideias da demonstração do teorema anterior.

Como o sistema de equações lineares $A'x' = b'$ tem a matriz A' tridiagonal e estritamente diagonalmente dominante (também é simétrica e positiva definida), esse pode ser resolvido diretamente por meio da decomposição LU, que neste caso toma a seguinte forma:

$$\begin{cases} u_1 = 2(h_1 + h_2) \\ l_j = \frac{h_j}{u_{j-1}}, j = 2, \ldots, n-1; \\ u_j = 2(h_{j-1} + h_j) - \frac{h_j^2}{u_{j+1}}, j = 2, \ldots, n-1 \end{cases}$$

Uma vez feita a decomposição LU, o sistema é resolvido em dois passos. Primeiramente calculando o vetor y de $Ly = b'$ por substituição direta e finalmente calculado o vetor x' composto de g_k de $Ux' = y$:

$$\begin{cases} y_1 = e_1 - e_0; \\ y_j = (e_j - e_{j-1}) - l_j y_{j-1}, j = 2, \ldots, n-1. \end{cases}$$

e

$$\begin{cases} g_{n-1} = \frac{y_{n-1}}{u_{n-1}}; \\ g_j = \frac{y_j - h_j g_{j+1}}{u_j}, j = n-2, \ldots, 1. \end{cases}$$

A interpolação por spline cúbica pode ser facilmente implementada computacionalmente. Uma sugestão é dividir este trabalho em duas subrotinas, a primeira lê os $n+1$ valores tabelados e retorna um array com os h's e outro com o g's. A outra rotina usa esses dois arrays para calcular os valores g_k e em seguida calcular os coeficientes a_k, b_k, c_k e d_k dos polinômios cúbicos.

• **Exemplo 1.6**

Consideremos a função $f(x) = x \sin(x)$ e tomemos os pontos $(x_i, f(x_i))$, onde os pontos x_i são $0, 2/5\,\pi, 4/5\,\pi, 6/5\,\pi, 8/5\,\pi, 2\,\pi$. Vamos traçar a spline cúbica natural interpolante para f nesses pontos.

A função spline interpolante $S(x)$ é dada por

$$\begin{aligned} S_0(x) &= 0.92947x + 0.013672x^3, \\ S_1(x) &= -0.05426 + 0.99424x + 0.05154\,(x - 1.25664)^2 - 0.52845\,(x - 1.25664)^3, \\ S_2(x) &= 4.94484 - 1.37970x - 1.94067\,(x - 2.51328)^2 + 0.55694\,(x - 2.51328)^3, \\ S_3(x) &= 11.42614 - 3.61867\,x + 0.15895\,(x - 3.76991)^2 + 0.87266\,(x - 3.76991)^3, \\ S_4(x) &= -9.37967 + 0.91497\,x + 3.448797\,(x - 5.026548246)^2 - 0.914822\,(x - 5.02655)^3. \end{aligned}$$

Veja a figura (2).

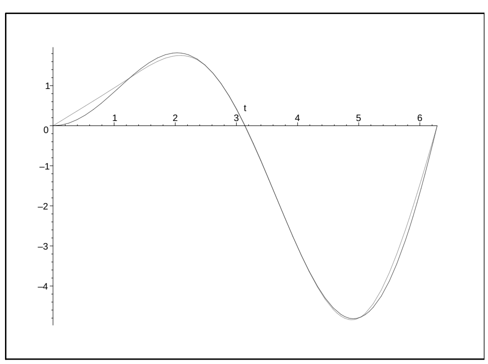

Figura 2: Spline cúbica

2 Alguns softwares

Atualmente dispomos de vários softwares que determinam as splines interpolantes lineares, quadráticas ou cúbicas. Dentre eles podemos citar Maple, MatLab, Mathematica, Maxima e Octave.

O Maple oferece rotinas prontas para serem utilizadas facilmente.

Veja como usar rotinas do Maple para determinar splines.

```
> restart:
> spline([0,1,2,3],[0,1,4,3],x,linear);
> spline([0,1,2,3],[0,1,4,3],x,cubic);
```

Outra opção é usar o pacote *CurveFitting*:

```
> restart:
> with(CurveFitting):
> Spline([0,1,2,3], [0,1,4,3], v, degree=1);
> Spline([0,1,2,3], [0,1,4,3], v, degree=3);
```

O MatLab também dispõe de um completo toolbox para determinar splines. O exemplo em MatLab abaixo gera o gráfico do seno juntamente com a spline.

```
x = 0:10; y = sin(x); xx = 0:.25:10; yy = spline(x,y,xx);
plot(x,y,'o',xx,yy)
```

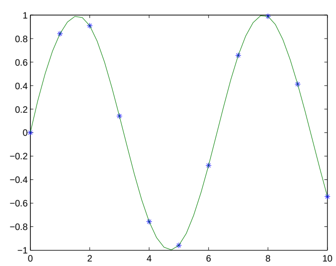

Figura 3: Spline cúbica por MatLab

Uma observação importante é que na decomposição LU usamos o caso especial da decomposição para uma matriz tridiagonal. Se A é a matriz tridiagonal a seguir

$$A = \begin{bmatrix} b_1 & c_1 & 0 & 0 & \cdots & & 0 \\ a_2 & b_2 & c_2 & 0 & \cdots & & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & c_{n-1} \\ 0 & 0 & \cdots & 0 & & a_{n-1} & b_n \end{bmatrix},$$

obtemos nesse caso que $A = LU$ onde

$$L = \begin{bmatrix} 1 & c_1 & 0 & 0 & \cdots & 0 \\ l_2 & 1 & 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \ddots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & & 1 & 0 \\ 0 & 0 & \cdots & 0 & l_n & 1 \end{bmatrix} \text{ e } U = \begin{bmatrix} u_1 & c_1 & 0 & 0 & \cdots & & 0 \\ 0 & u_2 & c_2 & 0 & \cdots & & 0 \\ \cdots & \cdots & \ddots & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & 0 & u_{n-1} & c_{n-1} \\ 0 & 0 & \cdots & 0 & 0 & 0 & u_n \end{bmatrix}$$

onde

$$
\begin{aligned}
u_1 &= b_1 \\
l_j &= \frac{a_j}{u_{j-1}} \\
u_j &= b_j - l_j c_{j-1}, j = 2, 3, \ldots, n
\end{aligned}
$$

Para resolver $Ax = d$, calcula-se a solução de $Ly = d$ dada por

$$
\begin{aligned}
y_1 &= d_1 \\
y_i &= d_i - l_i y_{i-1}, i = 2, 3, \ldots, n.
\end{aligned}
$$

Calcula-se então a solução x de $Ux = y$ dada por

$$
\begin{aligned}
x_n &= \frac{y_n}{u_n} \\
x_k &= y_k - c_k \frac{x_{k+1}}{u_k}, k = n-1, \ldots, 1.
\end{aligned}
$$

Referências

[1] S. D. Conte, Elementary Numerical Analysis. MacGraw-Hill, 1965.

[2] R. A. DeCarlo. *Linear Systems*. Prentice Hall, Inc. Englewood Cliffs, NJ, 1989.

[3] G. Golub & C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 1985.

[4] K. Atkinson. *An Introduction to Numerical Analysis*. John Willey& Sons, New York, 1983.