Interpolation & Polynomial Approximation

Cubic Spline Interpolation II

Numerical Analysis (9th Edition)

R L Burden & J D Faires

Beamer Presentation Slides
prepared by
John Carroll
Dublin City University

Outline

1. Unique natural cubic spline interpolant

Outline

1. Unique natural cubic spline interpolant

2. Natural cubic spline approximating $f(x) = e^x$

Outline

1. Unique natural cubic spline interpolant
2. Natural cubic spline approximating $f(x) = e^x$
3. Natural cubic spline approximating $\int_0^3 e^x \, dx$

Outline

1. Unique natural cubic spline interpolant
2. Natural cubic spline approximating $f(x) = e^x$
3. Natural cubic spline approximating $\int_0^3 e^x \, dx$

Existence of a unique natural spline interpolant

Theorem

Existence of a unique natural spline interpolant

Theorem

If f is defined at $a = x_0 < x_1 < \cdots < x_n = b$, then f has a unique natural spline interpolant S on the nodes $x_0, x_1, \ldots, x_n$; that is, a spline interpolant that satisfies the natural boundary conditions

$$S''(a) = 0 \qquad \text{and} \qquad S''(b) = 0$$

Existence of a unique natural spline interpolant

Proof (1/4)

- Using the notation

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$$

the boundary conditions in this case imply that $c_n = \frac{1}{2} S''_n(x_n)2 = 0$ and that $0 = S''(x_0) = 2c_0 + 6d_0(x_0 - x_0)$ so $c_0 = 0$.

Existence of a unique natural spline interpolant

Proof (1/4)

- Using the notation

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$$

the boundary conditions in this case imply that $c_n = \frac{1}{2}S_n''(x_n)2 = 0$ and that $0 = S''(x_0) = 2c_0 + 6d_0(x_0 - x_0)$ so $c_0 = 0$.

- The two equations $c_0 = 0$ and $c_n = 0$ together with the equations

$$h_{j-1}c_{j-1} + 2(h_{j-1} + h_j)c_j + h_j c_{j+1} = \frac{3}{h_j}(a_{j+1} - a_j) - \frac{3}{h_{j-1}}(a_j - a_{j-1})$$

produce a linear system described by the vector equation $A\mathbf{x} = \mathbf{b}$:

Existence of a unique natural spline interpolant

Proof (2/4)

A is the $(n+1) \times (n+1)$ matrix:

$$A = \begin{bmatrix} 1 & 0 & 0 & \cdots & \cdots & 0 \\ h_0 & 2(h_0+h_1) & h_1 & \ddots & & \vdots \\ 0 & h_1 & 2(h_1+h_2) & h_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ 0 & \cdots & \cdots & 0 & 0 & 1 \end{bmatrix}$$

Existence of a unique natural spline interpolant

Proof (3/4)

b and **x** are the vectors

$$\mathbf{b} = \begin{bmatrix} 0 \\ \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ \vdots \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ 0 \end{bmatrix} \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix}$$

Existence of a unique natural spline interpolant

Proof (4/4)

- The matrix A is strictly diagonally dominant, that is, in each row the magnitude of the diagonal entry exceeds the sum of the magnitudes of all the other entries in the row.

Existence of a unique natural spline interpolant

Proof (4/4)

- The matrix A is strictly diagonally dominant, that is, in each row the magnitude of the diagonal entry exceeds the sum of the magnitudes of all the other entries in the row.
- A linear system with a matrix of this form can be shown ▸ Theorem to have a unique solution for $c_0, c_1, \ldots, c_n$.

Existence of a unique natural spline interpolant

Proof (4/4)

- The matrix A is strictly diagonally dominant, that is, in each row the magnitude of the diagonal entry exceeds the sum of the magnitudes of all the other entries in the row.
- A linear system with a matrix of this form can be shown ▸ Theorem to have a unique solution for $c_0, c_1, \ldots, c_n$.

The solution to the cubic spline problem with the boundary conditions $S''(x_0) = S''(x_n) = 0$ can be obtained by applying the Natural Cubic Spline Algorithm.

Natural Cubic Spline Algorithm

To construct the cubic spline interpolant S for the function f, defined at the numbers $x_0 < x_1 < \cdots < x_n$, satisfying $S''(x_0) = S''(x_n) = 0$ (*Note:* $S(x) = S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$ *for* $x_j \leq x \leq x_{j+1}$.):

Natural Cubic Spline Algorithm

To construct the cubic spline interpolant S for the function f, defined at the numbers $x_0 < x_1 < \cdots < x_n$, satisfying $S''(x_0) = S''(x_n) = 0$ (*Note:* $S(x) = S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$ *for* $x_j \leq x \leq x_{j+1}$.):

INPUT $n; x_0, x_1, \ldots, x_n; a_0 = f(x_0), a_1 = f(x_1), \ldots, a_n = f(x_n)$

Natural Cubic Spline Algorithm

To construct the cubic spline interpolant S for the function f, defined at the numbers $x_0 < x_1 < \cdots < x_n$, satisfying $S''(x_0) = S''(x_n) = 0$ (*Note:* $S(x) = S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$ *for* $x_j \leq x \leq x_{j+1}$.):

INPUT	$n; x_0, x_1, \ldots, x_n; a_0 = f(x_0), a_1 = f(x_1), \ldots, a_n = f(x_n)$
OUTPUT	a_j, b_j, c_j, d_j for $j = 0, 1, \ldots, n-1$

Natural Cubic Spline Algorithm

To construct the cubic spline interpolant S for the function f, defined at the numbers $x_0 < x_1 < \cdots < x_n$, satisfying $S''(x_0) = S''(x_n) = 0$ (*Note:* $S(x) = S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$ *for* $x_j \le x \le x_{j+1}$.):

INPUT	$n; x_0, x_1, \ldots, x_n; a_0 = f(x_0), a_1 = f(x_1), \ldots, a_n = f(x_n)$
OUTPUT	a_j, b_j, c_j, d_j for $j = 0, 1, \ldots, n-1$
Step 1	For $i = 0, 1, \ldots, n-1$ set $h_i = x_{i+1} - x_i$

Natural Cubic Spline Algorithm

To construct the cubic spline interpolant S for the function f, defined at the numbers $x_0 < x_1 < \cdots < x_n$, satisfying $S''(x_0) = S''(x_n) = 0$ (*Note:* $S(x) = S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$ *for* $x_j \leq x \leq x_{j+1}$.):

INPUT $n; x_0, x_1, \ldots, x_n; a_0 = f(x_0), a_1 = f(x_1), \ldots, a_n = f(x_n)$

OUTPUT a_j, b_j, c_j, d_j for $j = 0, 1, \ldots, n-1$

Step 1 For $i = 0, 1, \ldots, n-1$ set $h_i = x_{i+1} - x_i$

Step 2 For $i = 1, 2, \ldots, n-1$ set

$$\alpha_i = \frac{3}{h_i}(a_{i+1} - a_i) - \frac{3}{h_{i-1}}(a_i - a_{i-1})$$

Natural Cubic Spline Algorithm

To construct the cubic spline interpolant S for the function f, defined at the numbers $x_0 < x_1 < \cdots < x_n$, satisfying $S''(x_0) = S''(x_n) = 0$ (*Note:* $S(x) = S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$ *for* $x_j \leq x \leq x_{j+1}$.):

INPUT $n; x_0, x_1, \ldots, x_n; a_0 = f(x_0), a_1 = f(x_1), \ldots, a_n = f(x_n)$

OUTPUT a_j, b_j, c_j, d_j for $j = 0, 1, \ldots, n-1$

Step 1 For $i = 0, 1, \ldots, n-1$ set $h_i = x_{i+1} - x_i$

Step 2 For $i = 1, 2, \ldots, n-1$ set

$$\alpha_i = \frac{3}{h_i}(a_{i+1} - a_i) - \frac{3}{h_{i-1}}(a_i - a_{i-1})$$

(Note: In what follows, Steps 3, 4, 5 and part of Step 6 solve a tridiagonal linear system using a Crout Factorization algorithm.)

Natural Cubic Spline Algorithm (Cont'd)

Step 3 Set $l_0 = 1$

$$\mu_0 = 0$$

$$z_0 = 0$$

Natural Cubic Spline Algorithm (Cont'd)

Step 3 Set $l_0 = 1$

$$\mu_0 = 0$$

$$z_0 = 0$$

Step 4 For $i = 1, 2, \ldots, n-1$

set $l_i = 2(x_{i+1} - x_{i-1}) - h_{i-1}\mu_{i-1}$

$$\mu_i = h_i / l_i$$

$$z_i = (\alpha_i - h_{i-1}z_{i-1}) / l_i$$

Natural Cubic Spline Algorithm (Cont'd)

Step 3 Set $l_0 = 1$

$\mu_0 = 0$

$z_0 = 0$

Step 4 For $i = 1, 2, \ldots, n-1$

set $l_i = 2(x_{i+1} - x_{i-1}) - h_{i-1}\mu_{i-1}$

$\mu_i = h_i / l_i$

$z_i = (\alpha_i - h_{i-1} z_{i-1}) / l_i$

Step 5 Set $l_n = 1$

$z_n = 0$

$c_n = 0$

Natural Cubic Spline Algorithm (Cont'd)

Step 3 Set $l_0 = 1$
$\mu_0 = 0$
$z_0 = 0$

Step 4 For $i = 1, 2, \ldots, n-1$
set $l_i = 2(x_{i+1} - x_{i-1}) - h_{i-1}\mu_{i-1}$
$\mu_i = h_i / l_i$
$z_i = (\alpha_i - h_{i-1} z_{i-1}) / l_i$

Step 5 Set $l_n = 1$
$z_n = 0$
$c_n = 0$

Step 6 For $j = n-1, n-2, \ldots, 0$
set $c_j = z_j - \mu_j c_{j+1}$
$b_j = (a_{j+1} - a_j)/h_j - h_j(c_{j+1} + 2c_j)/3$
$d_j = (c_{j+1} - c_j)/(3h_j)$

Natural Cubic Spline Algorithm (Cont'd)

Step 3 Set $l_0 = 1$
$\mu_0 = 0$
$z_0 = 0$

Step 4 For $i = 1, 2, \ldots, n-1$
set $l_i = 2(x_{i+1} - x_{i-1}) - h_{i-1}\mu_{i-1}$
$\mu_i = h_i / l_i$
$z_i = (\alpha_i - h_{i-1} z_{i-1}) / l_i$

Step 5 Set $l_n = 1$
$z_n = 0$
$c_n = 0$

Step 6 For $j = n-1, n-2, \ldots, 0$
set $c_j = z_j - \mu_j c_{j+1}$
$b_j = (a_{j+1} - a_j)/h_j - h_j(c_{j+1} + 2c_j)/3$
$d_j = (c_{j+1} - c_j)/(3h_j)$

Step 7 OUTPUT $(a_j, b_j, c_j, d_j$ for $j = 0, 1, \ldots, n-1)$ & STOP

Outline

1. Unique natural cubic spline interpolant
2. Natural cubic spline approximating $f(x) = e^x$
3. Natural cubic spline approximating $\int_0^3 e^x \, dx$

Natural Spline Interpolant

Example: $f(x) = e^x$

Use the data points $(0, 1)$, $(1, e)$, $(2, e^2)$, and $(3, e^3)$ to form a natural spline $S(x)$ that approximates $f(x) = e^x$.

Natural Spline Interpolant

Example: $f(x) = e^x$

Use the data points $(0, 1)$, $(1, e)$, $(2, e^2)$, and $(3, e^3)$ to form a natural spline $S(x)$ that approximates $f(x) = e^x$.

Solution (1/7)

With $n = 3$, $h_0 = h_1 = h_2 = 1$ and the notation

$$S(x) = S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$$

for $x_j \leq x \leq x_{j+1}$,

Natural Spline Interpolant

Example: $f(x) = e^x$

Use the data points $(0, 1)$, $(1, e)$, $(2, e^2)$, and $(3, e^3)$ to form a natural spline $S(x)$ that approximates $f(x) = e^x$.

Solution (1/7)

With $n = 3$, $h_0 = h_1 = h_2 = 1$ and the notation

$$S(x) = S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$$

for $x_j \leq x \leq x_{j+1}$, we have

- $a_0 = 1, \qquad a_1 = e$
- $a_2 = e^2, \qquad a_3 = e^3$

Natural Spline Interpolant

Solution (2/7)

So the matrix A and the vectors **b** and **x** given in the Natural Spline Theorem See A, b & x have the forms

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 0 \\ 3(e^2 - 2e + 1) \\ 3(e^3 - 2e^2 + e) \\ 0 \end{bmatrix} \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

Natural Spline Interpolant

Solution (2/7)

So the matrix A and the vectors $\mathbf{b}$ and $\mathbf{x}$ given in the Natural Spline Theorem See A, b & x have the forms

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 0 \\ 3(e^2 - 2e + 1) \\ 3(e^3 - 2e^2 + e) \\ 0 \end{bmatrix} \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

The vector-matrix equation $A\mathbf{x} = \mathbf{b}$ is equivalent to the system:

$$\begin{aligned} c_0 &= 0 \\ c_0 + 4c_1 + c_2 &= 3(e^2 - 2e + 1) \\ c_1 + 4c_2 + c_3 &= 3(e^3 - 2e^2 + e) \\ c_3 &= 0 \end{aligned}$$

Natural Spline Interpolant

Solution (3/7)

This system has the solution $c_0 = c_3 = 0$ and, to 5 decimal places,

$$\begin{aligned} c_1 &= \frac{1}{5}(-e^3 + 6e^2 - 9e + 4) \approx 0.75685 \\ c_2 &= \frac{1}{5}(4e^3 - 9e^2 + 6e - 1) \approx 5.83007 \end{aligned}$$

Natural Spline Interpolant

Solution (4/7)

Solving for the remaining constants gives

Natural Spline Interpolant

Solution (4/7)

Solving for the remaining constants gives

$$\begin{aligned} b_0 &= \frac{1}{h_0}(a_1 - a_0) - \frac{h_0}{3}(c_1 + 2c_0) \\ &= (e - 1) - \frac{1}{15}(-e^3 + 6e^2 - 9e + 4) \approx 1.46600 \\ b_1 &= \frac{1}{h_1}(a_2 - a_1) - \frac{h_1}{3}(c_2 + 2c_1) \\ &= (e^2 - e) - \frac{1}{15}(2e^3 + 3e^2 - 12e + 7) \approx 2.22285 \\ b_2 &= \frac{1}{h_2}(a_3 - a_2) - \frac{h_2}{3}(c_3 + 2c_2) \\ &= (e^3 - e^2) - \frac{1}{15}(8e^3 - 18e^2 + 12e - 2) \approx 8.80977 \end{aligned}$$

Natural Spline Interpolant

Solution (5/7)

$$d_0 = \frac{1}{3h_0}(c_1 - c_0) = \frac{1}{15}(-e^3 + 6e^2 - 9e + 4) \approx 0.25228$$

$$d_1 = \frac{1}{3h_1}(c_2 - c_1) = \frac{1}{3}(e^3 - 3e^2 + 3e - 1) \approx 1.69107$$

and

$$d_2 = \frac{1}{3h_2}(c_3 - c_1) = \frac{1}{15}(-4e^3 + 9e^2 - 6e + 1) \approx -1.94336$$

Natural Spline Interpolant

Solution (6/7)

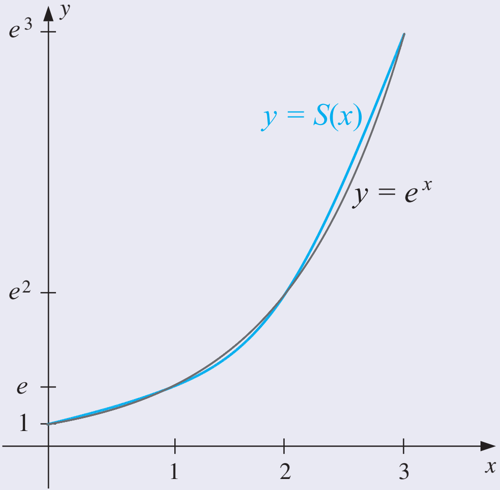
The natural cubic spine is described piecewise by

$$S(x) = \begin{cases} 1 + 1.46600x + 0.25228x^3 & \text{for } x \in [0,1] \\ 2.71828 + 2.22285(x-1) + 0.75685(x-1)^2 + 1.69107(x-1)^3 & \text{for } x \in [1,2] \\ 7.38906 + 8.80977(x-2) + 5.83007(x-2)^2 - 1.94336(x-2)^3 & \text{for } x \in [2,3] \end{cases}$$

The spline and its agreement with $f(x) = e^x$ are as shown in the following diagram.

Natural Spline Interpolant

Solution (7/7): Natural spline and its agreement with $f(x) = e^x$



Outline

1. Unique natural cubic spline interpolant
2. Natural cubic spline approximating $f(x) = e^x$
3. Natural cubic spline approximating $\int_0^3 e^x \, dx$

Natural Spline Interpolant

Example: The Integral of a Spline

Approximate the integral of $f(x) = e^x$ on $[0, 3]$, which has the value

$$\int_0^3 e^x \, dx = e^3 - 1 \approx 20.08553692 - 1 = 19.08553692,$$

by piecewise integrating the spline that approximates f on this integral.

Natural Spline Interpolant

Example: The Integral of a Spline

Approximate the integral of $f(x) = e^x$ on $[0, 3]$, which has the value

$$\int_0^3 e^x \, dx = e^3 - 1 \approx 20.08553692 - 1 = 19.08553692,$$

by piecewise integrating the spline that approximates f on this integral.

Note: From the previous example, the natural cubic spine $S(x)$ that approximates $f(x) = e^x$ on $[0, 3]$ is described piecewise by

$$S(x) = \begin{cases} 1 + 1.46600x + 0.25228x^3 & \text{for } x \in [0,1] \\ 2.71828 + 2.22285(x-1) + 0.75685(x-1)^2 + 1.69107(x-1)^3 & \text{for } x \in [1,2] \\ 7.38906 + 8.80977(x-2) + 5.83007(x-2)^2 - 1.94336(x-2)^3 & \text{for } x \in [2,3] \end{cases}$$

Natural Spline Interpolant

Solution (1/4)

We can therefore write

$$\begin{aligned}\int_0^3 S(x) = \int_0^1 & \left[1 + 1.46600x + 0.25228x^3\right] \; dx \\ + \int_1^2 & \left[2.71828 + 2.22285(x-1) + 0.75685(x-1)^2 \right. \\ & \left. +1.69107(x-1)^3\right] \; dx \\ + \int_2^3 & \left[7.38906 + 8.80977(x-2) + 5.83007(x-2)^2 \right. \\ & \left. -1.94336(x-2)^3\right] \; dx\end{aligned}$$

Natural Spline Interpolant

Solution (2/4)

Integrating and collecting values from like powers gives

Natural Spline Interpolant

Solution (2/4)

Integrating and collecting values from like powers gives

$$\int_0^3 S(x) = \left[x + 1.46600\frac{x^2}{2} + 0.25228\frac{x^4}{4}\right]_0^1$$

$$+ \left[2.71828(x-1) + 2.22285\frac{(x-1)^2}{2} +0.75685\frac{(x-1)^3}{3} + 1.69107\frac{(x-1)^4}{4}\right]_1^2$$

$$+ \left[7.38906(x-2) + 8.80977\frac{(x-2)^2}{2} +5.83007\frac{(x-2)^3}{3} - 1.94336\frac{(x-2)^4}{4}\right]_2^3$$

Natural Spline Interpolant

Solution (3/4)

Therefore:

$$\begin{aligned}\int_0^3 S(x) = \; & (1 + 2.71828 + 7.38906) \\ & + \frac{1}{2}(1.46600 + 2.22285 + 8.80977) \\ & + \frac{1}{3}(0.75685 + 5.83007) \\ & + \frac{1}{4}(0.25228 + 1.69107 - 1.94336) \\ = \; & 19.55229\end{aligned}$$

Natural Spline Interpolant

Solution (4/4)

Because the nodes are equally spaced in this example the integral approximation is simply

$$\begin{aligned}\int_0^3 S(x)\,dx &= (a_0 + a_1 + a_2) + \frac{1}{2}(b_0 + b_1 + b_2) \\ &\quad + \frac{1}{3}(c_0 + c_1 + c_2) + \frac{1}{4}(d_0 + d_1 + d_2)\end{aligned}$$

Questions?

Reference Material

Cubic Spline Interpolant

Definition

Given a function f defined on $[a, b]$ and a set of nodes $a = x_0 < x_1 < \cdots < x_n = b$, a cubic spline interpolant S for f is a function that satisfies the following conditions:

(a) $S(x)$ is a cubic polynomial, denoted $S_j(x)$, on the subinterval $[x_j, x_{j+1}]$ for each $j = 0, 1, \ldots, n-1$;

(b) $S_j(x_j) = f(x_j)$ and $S_j(x_{j+1}) = f(x_{j+1})$ for each $j = 0, 1, \ldots, n-1$;

(c) $S_{j+1}(x_{j+1}) = S_j(x_{j+1})$ for each $j = 0, 1, \ldots, n-2$; (*Implied by* (b).)

(d) $S'_{j+1}(x_{j+1}) = S'_j(x_{j+1})$ for each $j = 0, 1, \ldots, n-2$;

(e) $S''_{j+1}(x_{j+1}) = S''_j(x_{j+1})$ for each $j = 0, 1, \ldots, n-2$;

(f) One of the following sets of boundary conditions is satisfied:

   (i) $S''(x_0) = S''(x_n) = 0$ (natural *(or free)* boundary);

   (ii) $S'(x_0) = f'(x_0)$ and $S'(x_n) = f'(x_n)$ (clamped boundary).

Strictly Diagonally Dominant Matrices

Theorem

- A strictly diagonally dominant matrix A is nonsingular.
- Moreover, in this case, Gaussian elimination can be performed on any linear system of the form $A\mathbf{x} = \mathbf{b}$ to obtain its unique solution without row or column interchanges, and the computations will be stable with respect to the growth of round-off errors.

Return to Natural Spline Uniqueness Proof

Natural Spline Interpolant: Linear System $A\mathbf{x} = \mathbf{b}$

$$A = \begin{bmatrix} 1 & 0 & 0 & \cdots & \cdots & 0 \\ h_0 & 2(h_0 + h_1) & h_1 & \ddots & & \vdots \\ 0 & h_1 & 2(h_1 + h_2) & h_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & h_{n-2} & 2(h_{n-2} + h_{n-1}) & h_{n-1} \\ 0 & \cdots & \cdots & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} 0 \\ \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ \vdots \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ 0 \end{bmatrix} \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix}$$

Return to Natural Spline Example ($f(x) = e^x$)