IV - Interpolation and extrapolation

1. Introduction

(*) Problem: Given the values of a function

$f(x)$ at $x_1, x_2, \dots, x_N$ $(x_1 < x_2 < \dots < x_N)$

we wish to determine an analytic expression for $f(x)$ so that it can be computed at an arbitrary point

How to determine f(x

- Examples: Experimental data, stock markets, etc

- Interpolation: $x_1 \leq x \leq x_n$
- Extrapolation: $x < x_1$ or $x > x_n$

(*) Simplest possibility for fitting a function:

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

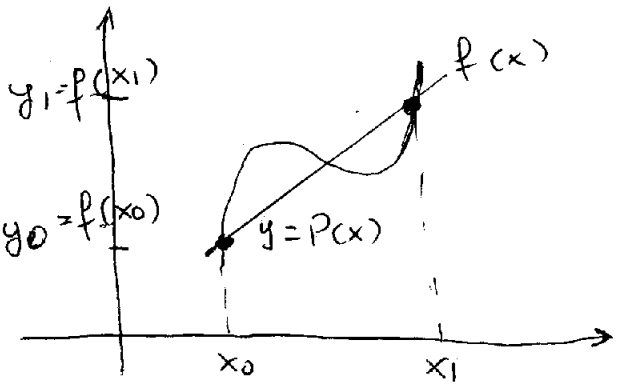
(algebraic polynomial)

=> We need a polynomial which approximates the points given in a satisfactory way

2 - Lagrange polynomials

2.1 - Key idea: To find approximating polynomials determined by specifying certain points through which they must pass

(*) Example: we wish to determine a polynomial of degree one that passes through the point (x_0, y_0) , (x_1, y_1)



This is the same as interpolating $f(x)$ ($f(x_0) = y_0$ and $f(x_1) = y_1$) by means of a 1st-degree polynomial agreeing with f at such points.

- Step 1: we define the functions

$$L_0(x) = \frac{x - x_1}{x_0 - x_1} \quad , \quad L_1(x) = \frac{x - x_0}{x_1 - x_0}$$

- Step 2: We take

$$P(x) = L_0(x) f(x_0) + L_1(x) f(x_1)$$

(*) Please note: $L_0(x_0) = 1$, $L_1(x_0) = 0$ so $P(x_0) = f(x_0)$

$L_0(x_1) = 0$, $L_1(x_1) = 1$ $P(x_1) = f(x_1)$

$\Rightarrow P(x)$ is the unique linear function passing through (x_0, y_0), (x_1, y_1)

2.2 - Generalization

- Task: we wish to construct a polynomial of degree at most n that passes through the $n+1$ points

$(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n))$

- We need a function $L_{n,k}(x)$ $(k = 0, 1, \ldots, n)$ so that

(a) $L_{n,k}(x_i) = 0$, $i \neq k$

(b) $L_{n,k}(x_k) = 1$

Hence

$$L_{n,k}(x) = \frac{(x-x_0)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_n)}{(x_k-x_0)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_n)} = \prod_{\substack{i=0 \\ i\neq k}}^{n} \frac{(x-x_i)}{(x_k-x_i)}$$

and

$$P(x) = f(x_0)L_{n,0}(x) + \cdots + f(x_n)L_{n,n}(x) = \sum_{k=0}^{n} f(x_k)L_{n,k}(x)$$

(n-th Lagrange interpolating polynomial)

(*) Error

• Lagrange error formula: $(n+1)$ st derivative of f

$$f(x) = P_n(x) + \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x-x_0)\cdots(x-x_n)$$

- $f(x)$: function to be interpolated
- $P_n(x)$: n-th interpolating polynomial
- $\xi(x)$: a number $\xi(x)$ (generally unknown)

(*) Please note: in most cases one can not apply such a formula, since:

- we do not know $f(x)$ and its derivatives
- we don't know the number ξ

(*) Drawbacks

- Error formula is difficult to apply
- One can not use P_{n-1} to compute P_n if one applies such a method directly

3 - Neville's method ("Neville's iterated interpolation)

• Key idea: The interpolating polynomials are generated recursively

$$P_{i(i+1)\cdots(i+m)} = \frac{(x-x_{i+m})P_{i(i+1)\cdots(i+m-1)} + (x_i - x)P_{(i+1)(i+2)\cdots(i+m)}}{x_i - x_{i+m}}$$

- $P_{i(i+1)\cdots(i+m-1)}$: Polynomial passing through $(x_i, y_i)\cdots(x_{i+m-1}, y_{i+m-1})$
- $P_{(i+1)(i+2)\cdots(i+m)}$: Polynomial passing through $(x_{i+1}, y_{i+1})\cdots(x_{i+m}, y_{i+m})$
- $P_{i(i+1)\cdots(i+m)}$: Lagrange polynomial passing through $(x_i, y_i)\cdots(x_{i+m}, y_{i+m})$

• Step 1 : Take the polynomials $P_0, P_1, \ldots, P_n$ (constants) passing by $(x_0, y_0) \ldots (x_n, y_n)$:

$P_0 = y_0$ passes by (x_0, y_0)

$P_1 = y_1$ passes by (x_1, y_1)

$P_2 = y_2$ passes by (x_2, y_2)

$P_n = y_n$ passes by (x_n, y_n)

• Use P_0 and P_1 to construct P_{01} passing by (x_0, y_0) and (x_1, y_1)

Step 2 : • Use P_1 and P_2 to construct P_{12} passing by (x_1, y_1) AND (x_2, y_2)

• Use P_2 and P_3 to construct P_{23} passing by (x_2, y_2) AND (x_3, y_3)

• Summary (5 points)

x_0 : $y_0 = P_0$ P_{01} P_{012} P_{0123} P_{01234}

x_1 : $y_1 = P_1$ P_{12} P_{123} P_{1234}

x_2 : $y_2 = P_2$ P_{23} P_{234}

x_3 : $y_3 = P_3$ P_{34}

x_4 : $y_4 = P_4$

(*) We are constructing an array of polynomials

$Q_{i,j} = P_{i-j}, P_{i-j+1}, \ldots P_{i-1}, P_i$.

1st column	2nd column	3rd column	4th column	...
$P_0 = Q_{00}$	$P_{01} = Q_{1,1}$	$P_{0,1,2} = Q_{2,2}$	$P_{0123} = Q_{33}$	
$P_1 = Q_{10}$	$P_{12} = Q_{2,1}$	$P_{1,2,3} = Q_{3,2}$	$P_{1234} = Q_{43}$	
$P_4 = Q_{40}$				

Algorithm

(to evaluate the polynomial $P(x)$ on the numbers $x_0, \dots, x_n$ at the number x for the function f)

Input: $x, x_0, x_1, \dots, x_n$

$f(x_0), f(x_1), \dots, f(x_n)$ as $Q_{0,0}, Q_{10}, \dots, Q_{n0}$ (first column of Q)

Output: The Table Q with $P(x) = Q_{nn}$

Step 1 For $i = 1, 2, \dots, n$

For $j = 1, 2, \dots, i$

Set $Q_{ij} = \dfrac{(x - x_{i-j})Q_{i,j-1} - (x - x_i)Q_{i-1,j-1}}{x_i - x_{i-j}}$

Step 2 Output (Q)

STOP

3 - Rational function interpolation (or Padé interpolation)

- Key idea : approximate a function $f(x)$ by the quotient of two polynomials (i.e., a "rational function") passing through the $m+1$ points $(x_i, y_i) \dots (x_{i+m}, y_{i+m})$

$$R(x) \equiv \frac{P_\mu(x)}{Q_\nu(x)} = \frac{p_0 + p_1 x + \dots + p_\mu x^\mu}{q_0 + q_1 x + \dots + q_\nu x^\nu}$$

(Polynomial $Q_\nu(x) \equiv 1$)

- Advantages :
  - i) In general, it has a smaller overall error than a polynomial approximation
  - ii) Can be used for approximating functions

with poles (i.e., zeros in the denominator)

iii) Do not oscillate like polynomials

3. Padé approximation: the rational function itself has a power series expansion which agrees with the first N+1 terms of the power series expansion of $f(x)$. (without loss of generality: $q_0 = 1$)

Explicitly: $f^{(k)}(0) = R^{(k)}(0)$, for each $k = 0, 1, \dots, N$ (*)

(particular case: $\mu = N$, $\nu = 0$ $\Rightarrow$ nth Taylor polynomial $\equiv R(x)$)

Let us consider $f(x) - R(x) = f(x) - \frac{P_\mu(x)}{Q_\nu(x)}$

with $f(x) = \sum_{i=0}^{\infty} a_i x^i$ (known; series expansion of x); $P_\mu(x) = \sum_{i=0}^{\mu} p_i x^i$ (p_i: unknown); $Q_\nu(x) = \sum_{i=0}^{\nu} q_i x^i$ (q_i: unkno

(*) How to determine p_i, q_i?

$$f(x) - R(x) = \frac{f(x)Q_\nu(x) - P_\mu(x)}{Q_\nu(x)} = \frac{\left(\sum_{i=0}^{\infty} a_i x^i\right)\left(\sum_{k=0}^{\nu} q_k x^k\right) - \sum_{k=0}^{\mu} p_k x^k}{Q_\nu(x)}$$

(*) $\Rightarrow$ The numerator has no terms of degrees less than N.

$$(q_0 + a_1 x + \cdots)(1 + q_1 x + \cdots + q_\nu x^\nu) - (p_0 + p_1 x + \cdots + p_\mu x^\mu) = 0, \quad k = 0, \dots, N$$

$$a_0 - p_0 = 0$$

$$(a_1 + a_0 q_1)x - p_1 x = 0 \Rightarrow a_1 + a_0 q_1 - p_1 = 0$$

$$a_1 q_1 + a_0 q_2 + a_2 \underbrace{q_0}_{=1} - p_2 = 0$$

One obtains a system of N+1 linear equations

General form:

$$\sum_{i=0}^{k} a_i q_{k-i} = p_k \qquad k = 0, 1, \ldots, N \qquad (*)$$

N+1 unknowns: $q_1, \ldots, q_\nu, p_0, p_1, \ldots, p_\mu$

(*) Please note: $p_k = 0 \quad , k > \mu$

$q_{k-i} = 0 \quad , k-i > \nu$

Task: solve this system

Example: find the Padé approximation of degree five to the function e^{-x}, considering the quotient of a polynomial $P(x)$ of third degree and a polynomial $Q(x)$ of degree 2.

(Note that $e^{-x} = \sum_{i=1}^{\infty} \frac{(-1)^i}{i!} x^i$)

$P(x) = p_0 + p_1 x + p_2 x^2 + p_3 x^3 \quad (\mu = 3)$

$Q(x) = 1 + q_1 x + q_2 x^2 \quad (\nu = 2)$

Padé approx

Coefficients of x^k, $k = 0, 1, \ldots, 5$ must vanish for $\underbrace{\tilde{f}(x)}_{\text{power series expansion of } f(x)} - R(x)$ with $R(x) = \frac{P(x)}{Q(x)}$

$$\underbrace{\left(1 - x + \frac{x^2}{2} - \frac{x^3}{6} + \cdots\right)}\left(1 + q_1 x + q_2 x^2\right) - \left(p_0 + p_1 x + p_2 x^2 + p_3 x^3\right)$$

Power series expansion of $f(x)$

$\longrightarrow a_0 = 1 \; ; \; a_1 = -1 \; ; \; a_2 = 1/2! \; ; \; a_n = \frac{(-1)^n}{n!}$

Eq. (*) must be satisfied for $k = 0, 1, \ldots, 5$

Coefficients of x^0: $a_0 q_0 = p_0 \Rightarrow 1 = p_0$

Coefficients of x^1: $a_0 q_1 + a_1 q_0 = p_1 \quad q_1 - 1 = p_1$

- Coefficients of x^2: $a_0 q_2 + a_1 q_1 + a_2 q_0 = p_2$

$$q_2 - q_1 + \frac{1}{2} = p_2$$

- Coefficients of x^3: $a_0 \underbrace{q_3}_{0} + \underbrace{a_1}_{-1} q_2 + \underbrace{a_2}_{1/2} q_1 + \underbrace{a_3}_{=-\frac{1}{6}} \underbrace{q_0}_{=1} = p_3$

$$-q_2 + \frac{1}{2} q_1 - \frac{1}{6} = p_3$$

- Coefficients of x^4: $\underbrace{a_0 q_4 + a_1 q_3}_{0} + a_2 q_2 + a_3 q_1 + a_4 q_0 - \underbrace{p_4}_{=0} =$

$$\frac{1}{2} q_2 - \frac{1}{6} q_1 + \frac{1}{24} = 0$$

- Coefficients of $x^5 = -\frac{1}{120} + \frac{1}{24} q_1 - \frac{1}{6} q_2 = 0$

Linear system

$-1 + q_1 = p_1 \Rightarrow P_1 + 0.P_2 + 0.P_3 - q_1 + 0.q_2 = -1 \quad E_1$

$q_2 - q_1 + \frac{1}{2} = p_2 \Rightarrow 0.P_1 + P_2 + 0.P_3 + q_1 - q_2 = 1/2 \quad E_2$

$-q_2 + \frac{1}{2} q_1 - \frac{1}{6} = P_3 \Rightarrow 0.P_1 + 0.P_2 + P_3 + \frac{1}{2} q_1 + q_2 = -1/6 \quad E_3$

$+\frac{1}{2} q_2 - \frac{1}{6} q_1 + \frac{1}{24} = 0 \Rightarrow 0.P_1 + 0.P_2 + 0.P_3 + 1/6 q_1 - \frac{1}{2} q_2 = 1/24 \quad E_4$

$\frac{1}{24} q_1 - \frac{1}{6} q_2 - \frac{1}{120} = 0 \Rightarrow +0.P_1 + 0.P_2 + 0.P_3 - \frac{1}{24} q_1 + \frac{1}{6} q_2 = -1/120 \quad E_5$

Matrix of the system:

$$A = \left(\begin{array}{ccccc|c} 1 & 0 & 0 & -1 & 0 & -1 \\ 0 & 1 & 0 & 1 & -1 & 1/2 \\ 0 & 0 & 1 & -1/2 & 1 & -1/6 \\ 0 & 0 & 0 & 1/6 & -1/2 & 1/24 \\ 0 & 0 & 0 & -1/24 & 1/6 & -1/120 \end{array}\right)$$

factorize this matrix using the methods from the previous lectures

(*) Please note:

- Coefficients in front of p_i:

a_{ij} , $1 \le i \le N$, $1 \le j \le M$

$a_{ii} = 1$ (main diagonal; goes until $i = j = \mu$)

$a_{ij} = 0$, $j = i+1, \mu$ (above main diagonal)

$a_{ij} = 0$; $j = 1, 2 \cdots, i-1$ (below main diagonal)

- Coefficients in front of q_i:

a_{ij} , $1 \le i \le N$, $\mu < j \le N$

ν columns $\rightarrow j = \mu + k$, $k = 1, \cdots, \nu$

$a_{ij} = -a_{i-k}$ $(k = j - \mu)$, $i > k$

Examples $a_{1,\mu+1} = -a_0$

$a_{2,\mu+1} = -a_1$

$a_{ij} = 0$, $k \ge i+1$ $\cdots\cdots$ $(j = \mu+i+1, \cdots, N)$

- Terms on the RHS: $a_{i,N+1} = a_i$

Answers:

$p_0 = 1$

$p_1 = -3/5$

$p_2 = 3/20$

$p_3 = -1/60$

$q_1 = 2/5$

$q_2 = 1/20$

$$R(x) = \frac{1 - (3/5)x + \frac{3}{20}x^2 - \frac{1}{60}x^3}{1 + \frac{2}{5}x + \frac{1}{20}x^2}$$

(the errors obtained using $R(x)$ for approximating $f(x) = e^{-x}$ are much smaller than those obtained with its 5th-order Taylor expansion)

Algorithm (Padé Rational Approximation $R(x) = \frac{\sum_{i=0}^{\mu} p_i x^i}{\sum_{i=0}^{\nu} q_i x^i}$ for a function $f(x)$)

Input: nonnegative integers μ, ν

Output: Coefficients $q_0, q_1, \dots, q_\nu$ and $p_0, p_1, \dots, p_\mu$

Step 1 Set $N = \mu + \nu$

Step 2 For $i = 0, 1, \dots, N$ set $a_i = \frac{f^{(i)}(0)}{i!}$ (coeff. of Taylor expansion
(Ok if you know the function; otherwise they will enter as an input)

Step 3 Set $q_0 = 1$
$p_0 = q_0$

Step 4 For $i = 1, 2, \dots, N$ do Steps 5-10 (linear system with matrix B)

Step 5 For $j = 1, 2, \dots, i-1$
if $j \le \mu$ then $b_{ij} = 0$ (fills below main diagon

Step 6 : If $i = 1, \dots, \mu$ then $b_{ii} = 1$ (fills main diagona up to μ

Step 7 : For $j = i+1, i+2, \dots, N$ set $b_{ij} = 0$ (fills above ma diagonal)

Step 8 : For $j = 1, 2, \dots, i$
if $j \le \nu$ then set $b_{i,\mu+j} = -a_{i-j}$

Step 9 : For $j = \mu+i+1, \mu+i+2, \dots, N$ set $b_{ij} = 0$

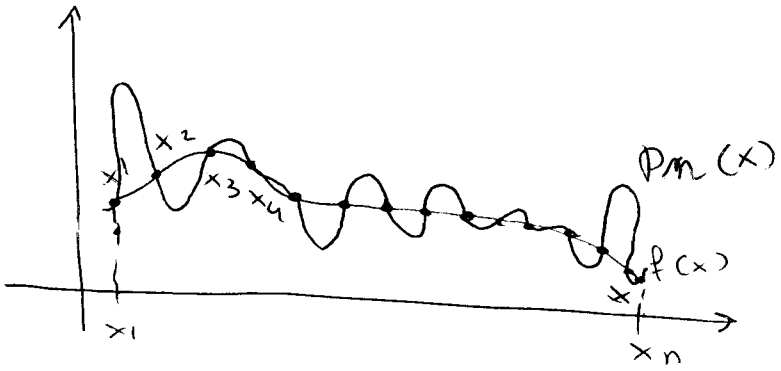
Step 10 : Set $b_{i,N+1} = a_i$

; Solve linear system

4 - Cubic spline Interpolation

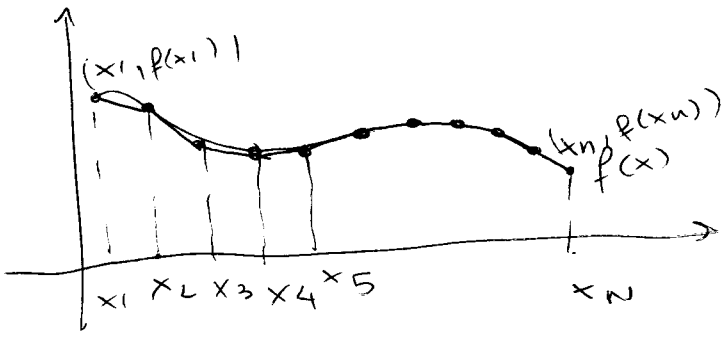
4.1. Preliminaries

- Problem:
  - Polynomial approximations may lead to spurious oscillations



- Solution: divide the interval into a set of sub-interv and construct a different approximating polynomial on each sub-interval ("piecewise polynomial approxim

- Piecewise linear interpolation: joins a set of data points by straight lines



Problems: no differentiability at the endpoints of the subintervals (interpolating function is not "smooth").

- Cubic spline interpolation: uses cubic polynomials between each successive pair of nodes

4.2 - Definition

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

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

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

(c) $S_{j+1}(x_{j+1}) = S_j(x_{j+1})$ for each $j = 0, 1, \dots, n-2$

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

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

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

(i) $S''(x_0) = S''(x_n) = 0$ (free or natural boundary; S is a "natural spline")

(ii) $S'(x_0) = f'(x_0)$ and $S'(x_n) = f'(x_n)$ (clamped boundary; S is a "clamped cubic spline" interpolant)

4.3 - Method

• Apply the conditions in 4.2 to the cubic polynomial

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

for $j = 0, 1, \dots, n-1$

(b) $S_j(x_j) = f(x_j) = a_j$

(c) $S_{j+1}(x_{j+1}) = S_j(x_{j+1})$

$$S_{j+1}(x) = a_{j+1} + b_{j+1}(x - x_{j+1}) + c_{j+1}(x - x_{j+1})^2 + d_{j+1}(x - x_{j+1})^3$$

$$S_{j+1}(x_{j+1}) = a_{j+1}$$

$$S_j(x_{j+1}) = a_j + b_j(x_{j+1} - x_j) + c_j(x_{j+1} - x_j)^2 + d_j(x_{j+1} - x_j)^3$$

$x_{j+1} - x_j = h_j$

$$\boxed{a_{j+1} = a_j + b_j h_j + c_j h_j^2 + d_j h_j^3} \quad \text{for each } j = 0, \dots, n$$

(d) $S'_{j+1}(x_{j+1}) = S'_j(x_{j+1})$

$$S'_j(x) = b_j + 2c_j(x - x_j) + 3d_j(x - x_j)^2$$

$$S'_j(x_j) = b_j \quad (*)$$

$$S'_j(x_{j+1}) = b_j + 2c_j h_j + 3d_j h_j^2 \quad (**)$$

$$S'_{j+1}(x_{j+1}) = b_{j+1} \quad (***)$$

$(*) = (***) \Rightarrow$ $\boxed{b_{j+1} = b_j + 2c_j h_j + 3d_j h_j^2}$

for each $j = 0, \dots, n-1$

(e) $S''_{j+1}(x_{j+1}) = S''_j(x_{j+1})$

$$S''_j(x) = 2c_j + 6d_j(x - x_j) \Rightarrow S''_j(x_j) = 2c_j$$

$$S''_{j+1}(x_{j+1}) = 2c_{j+1}$$

$$S_j''(x_{j+1}) = 2c_j + 6h_j d_j = 2c_{j+1}$$

$$\Rightarrow \boxed{c_{j+1} = c_j + 3h_j d_j} \qquad j = 0, 1, \ldots, n-1$$

$$d_j = \frac{c_{j+1} - c_j}{3h_j}$$

This gives

$$a_{j+1} = a_j + b_j h_j + \frac{h_j^2}{3}(2c_j + c_{j+1}) \qquad (*)$$

$$b_{j+1} = b_j + h_j(c_j + c_{j+1}) \qquad (**)$$

Solving (*) for b_j and b_{j+1} we have

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

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

$$j \to j-1 : \quad b_{j-1} = \frac{a_j - a_{j-1}}{h_{j-1}} - \frac{h_{j-1}}{3}(2c_{j-1} + c_j)$$

Inserting into (**) (with $j \to j-1$):

$$b_j = b_{j-1} + h_{j-1}(c_{j-1} + c_j)$$

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

Rearranging one gets

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

known (a_j's known, h_j's known)

$j = 1, 2, \dots, n-1$

- Task: determine $\{c_j\}_{j=0}^{n}$

Values of f at nodes

spacing

(*) Please note: together with either of the boundary conditions in (f) one will obtain a tridiagonal system (easy to solve!)

(f) Boundary conditions:

i) Free or natural boundary:

$S''(a) = 0$ and $S''(b) = 0$

This implies that

$S_0''(x_0) = 0 \Rightarrow c_0 = 0$

$S_n''(x_n) = 0 \Rightarrow c_n = 0$

$\Rightarrow$ One needs to solve the tridiagonal system $A\vec{x} = \vec{b}$ with from boundary conditions

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

from boundary conditions

$$\vec{b} = \begin{bmatrix} \textcircled{0} \\ \frac{3}{h_1}(a_2-a_1) - \frac{3}{h_0}(a_1-a_0) \\ \vdots \\ \frac{3}{h_{n-1}}(a_n-a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1}-a_{n-2}) \\ \textcircled{0} \end{bmatrix}$$

from boundary conditions

$$\vec{x} = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix}$$

ii) clamped cubic spline:

$$f'(a) = S'(a) = S_0'(x_0) = b_0$$
$$f'(b) = S'(b) = S_n'(x_n) = b_n$$

$$b_0 = \frac{1}{h_0}(a_1 - a_0) - \frac{h_0}{3}(2c_0 + c_1)$$

$$\Rightarrow \boxed{h_0(2c_0 + c_1) = -3f'(a) + \frac{3}{h_0}(a_1 - a_0)}$$

$$f'(b) = b_n = b_{n-1} + h_{n-1}(c_{n-1} + c_n)$$

$$b_{n-1} = \frac{1}{h_{n-1}}(a_n - a_{n-1}) - \frac{h_{n-1}}{3}(2c_{n-1} + c_n)$$

$$f'(b) = \frac{1}{h_{n-1}}(a_n - a_{n-1}) - \frac{h_{n-1}}{3}(2c_{n-1} + c_n) + h_{n-1}(c_{n-1} + c_n)$$

$$\Rightarrow \boxed{h_{n-1}(2c_{n-1} + c_n) = -3f'(b) + \frac{3}{h_{n-1}}(a_n - a_{n-1})}$$

This implies changes in the matrix $\vec{A}$ and in the vector $\vec{b}$

with respect to case i) :

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

depends on the boundary conditions

depends on the boundary conditions

$$\vec{b} = \begin{bmatrix} \frac{3}{h_0}(a_1 - a_0) - 3f'(a) \\ \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ \vdots \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ 3f'(b) - \frac{3}{h_{n-1}}(a_n - a_{n-1}) \end{bmatrix}$$

depends on the boundary conditions

depends on the boundary conditions

(*) In both cases, the matrices are also strictly diagonally dominant. This means that, if one solves the above linear systems i) No row interchanges are necessary.

ii) The round-off error is under control