Funzioni spline

Dato l'intervallo $[a, b]$, si consideri una successione finita di numeri reali (nodi) appartenenti all'intervallo, tali che $a \leq x_0 < x_1 < x_2 < ... < x_m < x_{m+1} \leq b$ (a e b possono assumere i valori $-\infty$ e $+\infty$). Si individua in tal modo *una partizione* dell'intervallo $[a, b]$ in $m + 1$ sottointervalli $I_i = [x_i, x_{i+1})$, $I_m = [x_m, x_{m+1}]$.

Definizione

Si dice *funzione spline* di grado n o di ordine $n + 1$ relativa alla partizione $\{x_i\}_{i=0,\dots,m+1}$ di $[a, b]$ una funzione $s(x)$ che soddisfa le seguenti due proprietà:

1. $s(x)$ è un polinomio $s_i(x)$ di grado non superiore a n in ciascun sottointervallo I_i della partizione, $i = 0, ..., m$;
2. $s(x) \in C^{n-1}[a, b]$, ossia la funzione e le sue derivate fino all'ordine $n - 1$ sono continue sull'intervallo $[a, b]$; ciò significa che per ogni nodo interno alla partizione valgono le seguenti mn condizioni:

$$s_i^{(k)}(x_{i+1}) = s_{i+1}^{(k)}(x_{i+1}), \quad i = 0, ..., m-1; k = 0, 1, .., n-1$$

In altre parole, una spline $s(x)$ entro ciascun intervallo I_i, $i = 0, \dots, m$ è un polinomio di grado al più n che in ogni punto interno all'intervallo è C^∞ e agli estremi coincide con il polinomio relativo all'intervallo precedente (se esiste) e con quello dell'intervallo successivo (se esiste) fino alla derivata $n - 1$–esima.

L'insieme delle spline di grado n relative alla partizione $\{x_i\}_{i=0,\dots,m+1}$ si denota con $S_n\{x_1, ..., x_m\}$. Tale insieme contiene l'insieme dei polinomi di grado non superiore ad n.

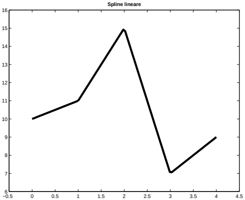
Si osserva che la somma di funzioni spline è ancora una funzione spline e che il prodotto di uno scalare reale per una funzione spline è ancora una funzione spline. **Pertanto l'insieme delle spline $S_n\{x_1, ..., x_m\}$ è uno spazio funzionale lineare.**

Inoltre, la derivata di una spline di grado n è una spline di grado $n-1$ relativa alla medesima partizione e l'integrale di una spline di grado n è una spline di grado $n+1$ relativa alla medesima partizione.

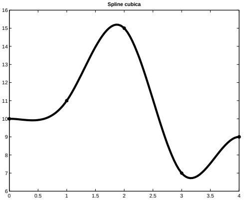
Ogni spline dipende da $(n+1)(m+1)$ parametri che devono soddisfare nm condizioni nei nodi interni (uguaglianza dei valori della funzione e delle derivate fino all'ordine $n-1$). Pertanto ogni spline dipende da $m+n+1$ parametri.

Esempi

Spline di grado 1: $s_i(x)$ è il segmento che unisce (x_i, y_i) con (x_{i+1}, y_{i+1}) e vale che $s_i(x_{i+1}) = s_{i+1}(x_{i+1})$. E' una funzione di classe C^0 e dipende da $2.(m+1) - m = m+2$ parametri.



Spline di grado 3 o cubica: $s_i(x)$ è un polinomio di grado al più 3 in $[x_i, x_{i+1}]$ e vale che $s_i(x_{i+1}) = s_{i+1}(x_{i+1})$, $s_i{}'(x_{i+1}) = s_{i+1}{}'(x_{i+1})$, $s_i{}^{(2)}(x_{i+1}) = s_{i+1}{}^{(2)}(x_{i+1})$. Dipende da $4(m+1) - 3m = m+4$ parametri.



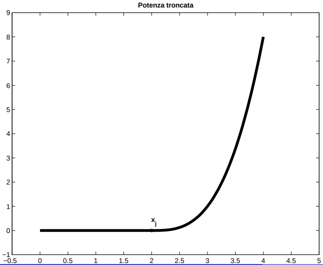
Teorema

Lo spazio delle funzioni spline $S_n\{x_1, ..., x_m\}$ è uno spazio lineare di dimensione $m + n + 1$ e ogni funzione spline $s(x)$ è univocamente rappresentabile nella forma

$$s(x) = p_n(x) + \sum_{j=1}^{m} \alpha_j (x - x_j)^n{}_+$$

ove $p_n(x)$ è un polinomio di grado n e

$$(x - x_j)^n{}_+ = \begin{cases} (x - x_j)^n & \text{per } x \geq x_j \\ 0 & \text{per } x \leq x_j \end{cases}$$



Dimostrazione.

- Una combinazione lineare di funzione spline è una funzione spline: perciò lo spazio $S_n\{x_1, ..., x_m\}$ è lineare.
- Per $j = 0, ..., m$, sia $s_j(x)$ il polinomio di grado n che rappresenta la spline nell'intervallo $[x_j, x_{j+1}]$.
  Si consideri la funzione $s_j(x) - s_{j-1}(x)$ che, nell'intervallo $[x_{j-1}, x_{j+1}]$ è di classe C^{n-1}. Poichè in x_j questa funzione si annulla insieme a tutte le derivate fino all'ordine $n-1$, x_j è uno zero di molteplicità n. Inoltre, essendo la differenza tra polinomi di grado n, $s_j(x) - s_{j-1}(x)$ è un polinomio di grado n dato da

$$s_j(x) - s_{j-1}(x) = \alpha_j(x - x_j)^n, \quad j = 1, ..., m$$

Allora il polinomio che rappresenta la spline in un intervallo generico $[x_i, x_{i+1}]$ può essere scritto come:

$$s_i(x) = s_0(x) + (s_1(x) - s_0(x)) + (s_2(x) - s_1(x)) + ... + (s_i(x) - s_{i-1}(x))$$

Ponendo $s_0(x) = p_n(x)$, si ha:

$$s_i(x) = p_n(x) + \alpha_1(x - x_1)^n + \alpha_2(x - x_2)^n + ... + \alpha_i(x - x_i)^n$$

Poichè per $x > x_{i+1}$, $\sum_{j=i+1}^{m} \alpha_j(x - x_j)^n{}_+ = 0$, segue la tesi. Dunque ogni spline si scrive come il polinomio di grado n che rappresenta la spline nel primo sottointervallo più una combinazione di potenze troncate.

Dimostrazione. (cont.)

- Resta da provare che la rappresentazione è unica, cioè che, per una assegnata spline, $p_n(x)$ e i coefficienti α_j, $j = 1, ..., m$ sono univocamente determinati. Per $x \leq x_1$ le potenze troncate sono nulle e dunque $s(x) = p_n(x)$. Questo polinomio è la rappresentazione della spline nel primo intervallo e quindi è univocamente determinato.
  Per $x > x_1$, consideriamo $s_j(x) - s_{j-1}(x) = \alpha_j(x - x_j)^n$. Differenziando n volte l'uguaglianza si ha:
  $$s_j^{(n)}(x) - s_{j-1}^{(n)}(x) = n!\alpha_j.$$
  Allora per $x = x_j$, $j = 1, 2, ..., m$, si ha:
  $$\alpha_j = \frac{s_j^{(n)}(x_j+) - s_{j-1}^{(n)}(x_j-)}{n!}$$
  Quindi α_j sono univocamente determinati.
- Poichè allora ogni spline si scrive in modo unico come combinazione lineare delle funzioni:
  $$1, (x - x_0), (x - x_0)^2, ..., (x - x_0)^n, (x - x_1)^n{}_+, ..., (x - x_m)^n{}_+$$
  che sono linearmente indipendenti, segue che la dimensione dello spazio delle spline è $n + 1 + m$.

Spline periodiche

Definizione. Una funzione spline di grado n relativa ai nodi $x_1, ..., x_{m+1}$ si dice **periodica** di periodo $x_{m+1} - x_1$ se è una spline che soddisfa le ulteriori n condizioni:

$$s^{(k)}(x_1) = s^{(k)}(x_{m+1}), \qquad k = 0, ..., n-1$$

Lo spazio delle funzioni spline periodiche è uno spazio lineare di dimensione m $((n+1)m - mn = m)$.

Spline naturali

Definizione. Una funzione spline di grado dispari $n = 2k - 1$ relativa ai nodi $x_0, x_1, ..., x_{m+1}$ si dice **naturale** se è una spline che negli intervalli $[x_0, x_1]$ e $[x_m, x_{m+1}]$ diventa un polinomio di grado $k - 1$. Di conseguenza $s^{(j)}(x_0) = s^{(j)}(x_{m+1}) = 0$, $j = k, k + 1, ..., 2k - 2$.

Per esempio una spline cubica è naturale se nel primo e nell'ultimo intervallo è un segmento. Pertanto $s''(x_0) = s''(x_{m+1}) = 0$.

In questo caso i parametri da cui dipende la spline diventano:

$$m + n + 1 - (2k - 2) = m + 2k - 1 + 1 - 2k + 2 = m + 2$$

ossia dal numero totale di nodi della suddivisione.

Spline naturali (cont.)

Teorema

Una funzione spline naturale è univocamente rappresentabile nella forma

$$s(x) = p_{k-1}(x) + \sum_{j=1}^{m} \alpha_j (x - x_j)^n{}_+$$

ove $p_{k-1}(x)$ è un polinomio di grado $k-1$ e i coefficienti α_j soddisfano le k condizioni:

$$\sum_{j=1}^{m} \alpha_j x_j^r = 0, \quad r = 0, .., k-1$$

Dimostrazione. Poichè una spline naturale è una spline, si può rappresentare come:

$$s(x) = p_n(x) + \sum_{j=1}^{m} \alpha_j (x - x_j)^n{}_+$$

ove, in questo caso, $p_n(x)$ deve essere un polinomio di grado $k-1$, poichè è il polinomio che rappresenta la spline nel primo intervallo $[x_0, x_1]$. Dunque $p_n(x) = p_{k-1}(x)$. Inoltre per $x > x_m$, la spline deve essere un polinomio di grado

Spline naturali (cont.)

$k-1$ e quindi tutti i coefficienti dei termini x^r, con $r=k,..,n$ devono essere nulli. Poichè $(x-x_j)^n = \sum_{r=0}^{n} \frac{n!}{r!(n-r)!} x^{n-r}(-1)^r x_j^r$, segue che deve essere nullo il coefficiente di x^{n-r}, $r=0,...,k-1$, ossia deve valere $\sum_{j=1}^{m} \alpha_j x_j^r = 0$:

$$\begin{aligned}
\sum_{j=1}^{m} \alpha_j (x-x_j)^n &= \sum_{j=1}^{m} \sum_{r=0}^{n} \alpha_j \frac{n!}{r!(n-r)!} x^{n-r}(-1)^r x_j^r \\
&= \sum_{r=0}^{n} \frac{n!}{r!(n-r)!} x^{n-r}(-1)^r \sum_{j=1}^{m} \alpha_j x_j^r
\end{aligned}$$

La rappresentazione di una spline può essere fatta in modo analogo partendo da sinistra verso destra e considerando le potenze troncate ${(x_j-x)^n}_+$.

Interpolazione con Spline Lineari

Teorema

Dati $n+2$ nodi distinti in $[a,b]$ denotati con $x_0, x_1, \ldots, x_{n+1}$, tali che $x_0 \geq a$, $x_{n+1} \leq b$ e $x_i < x_{i+1}$ $(i = 0, \ldots, n)$ e assegnati $y_0, \ldots, y_{n+1}$, esiste una e una sola spline lineare $s(x)$ tale che $s(x_i) = y_i$, $i = 0, \ldots, n+1$.
È

$$s(x) = \sum_{i=0}^{n+1} y_i l_i(x)$$

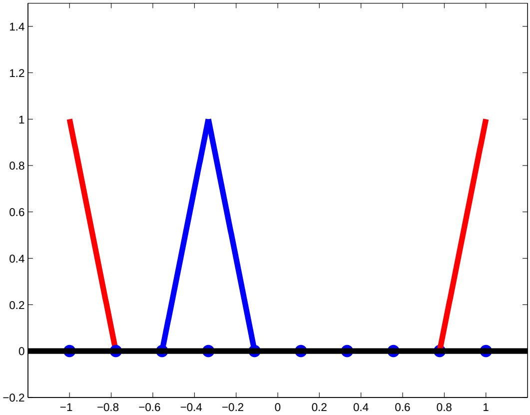
con

$$l_i(x) = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}}, & x \in [x_{i-1}, x_i]; \\ \frac{x - x_{i+1}}{x_i - x_{i+1}}, & x \in [x_i, x_{i+1}]; \\ 0, & \text{altrove.} \end{cases} \qquad i = 1, \ldots, n$$

per $i = 1, \ldots, n$.

$$l_0(x) = \begin{cases} \frac{x - x_1}{x_0 - x_1}, & x \in [x_0, x_1]; \\ 0, & \text{altrove;} \end{cases} \qquad l_{n+1} = \begin{cases} \frac{x - x_n}{x_{n+1} - x_n}, & x \in [x_n, x_{n+1}]; \\ 0, & \text{altrove} \end{cases}$$

Interpolazione con Spline Lineari (cont.)



Dimostrazione. Detta $s_i(x)$ la spline in $[x_i, x_{i+1}]$, deve essere $s_i(x) = m^{(i)}x + q^{(i)}$, $i = 0, \ldots, n$.
Inoltre deve essere $s(x_i) = y_i$, $s(x_{i+1}) = y_{i+1}$.
Da cui

$$\begin{cases} m^{(i)}x_i + q^{(i)} = y_i \\ m^{(i)}x_{i+1} + q^{(i)} = y_{i+1} \end{cases}$$

e poiché $\begin{vmatrix} x_i & 1 \\ x_{i+1} & 1 \end{vmatrix} \neq 0$
il sistema ammette una e una sola soluzione

$$\begin{aligned} s_i(x) &= y_i \frac{x - x_{i+1}}{x_i - x_{i+1}} + y_{i+1} \frac{x - x_i}{x_{i+1} - x_i} = \frac{y_i(x - x_{i+1}) - y_{i+1}(x - x_i)}{x_i - x_{i+1}} \\ &= y_i l_i(x) + y_{i+1} l_{i+1}(x) \end{aligned}$$

per $x \in [x_i, x_{i+1}]$. In tale intervallo

$$y_0 l_0(x) + \ldots + y_{i-1} l_{i-1}(x) + y_{i+2} l_{i+2}(x) + \ldots + y_{n+1} l_{n+1}(x) = 0$$

Da cui $s(x) = \sum_{i=0}^{n+1} y_i l_i(x)$.

Analisi dell'errore

In $[x_i, x_{i+1}]$, l'errore commesso è pari a quello di interpolazione con un polinomio di grado 1.
Se $f \in C^2([a,b])$,
$\Big(f(x) - s_i(x)\Big) = \frac{f^{''}(\xi_i)}{2!}(x - x_i)(x - x_{i+1}), \quad x \in [x_i, x_{i+1}], \quad x_i < \xi < x_{i+1}.$
Se $|f^{''}(x)| \leq M$ per $x \in [a,b]$, segue

$$|f(x) - s_i(x)| \leq \frac{M}{2}\max|(x - x_i)(x - x_{i+1})| = \frac{M}{2}\frac{(x_{i+1} - x_i)^2}{4}$$

Posto $h = \max_{i=0,\ldots,n}(x_{i+1} - x_i)$

$$\max_{x\in[a,b]}|f(x) - s(x)| = \|f(x) - s(x)\|_\infty \leq \frac{M}{8}h^2$$

La complessità del metodo è di 3 prodotti e 4 addizioni.

Codice Matlab

Calcolo di una spline lineare interpolante relativa a (x, y) nei punti z.

```
function s=spline_lineare(x,y,z);
%
% si assume che x sia un vettore ordinato
%  x vettore dei nodi
%  y vettore delle osservazioni
%  z vettore dei valori in cui calcolo la spline lineare
%
for i=1:length(z)
   if z(i)<x(1) | z(i)>x(end)
      error('punto non interno all''intervallo di interpolazione');
   end;
   ij=find(x==z(i));
   if ~isempty(ij)
      s(i)=y(ij(1));
   else
   ij=find(x>z(i));
   if length(ij)==0
      error('punto non interno all''intervallo di interpolazione');
```

Codice Matlab (cont.)

```
   else
      k=ij(1);
      temp=x(k-1)-x(k);
      s(i)=(y(k-1)* (z(i)-x(k)) -y(k)*(z(i)-x(k-1)))/temp;
   end;
   end;
end;
```

Interpolazione di Hermite tramite polinomi cubici a tratti

Assegnati nodi $a \leq x_0 < x_1 < \ldots < x_n < x_{n+1} \leq b$, si vuol determinare una funzione $g(x)$ che in ogni intervallo $[x_i, x_{i+1}]$, $i = 0, \ldots, n$, è un polinomio cubico e tale che, se $\widetilde{s}_i(x)$ rappresenta tale funzione in $[x_i, x_{i+1}]$ vale che

$$\begin{array}{ll} \widetilde{s}_i(x_i) = y_i \quad \widetilde{s}_i(x_{i+1}) = y_{i+1} & \\ & i = 0, \ldots, n \\ \widetilde{s}_i'(x_i) = z_i \quad \widetilde{s}_i'(x_{i+1}) = z_{i+1} & \end{array}$$

Si assume che $y_0, y_1, \ldots, y_{n+1}$ e $z_0, z_1, \ldots, z_{n+1}$ siano assegnati.
Dall'interpolazione di Hermite, si ha

$$\begin{array}{llllll}
x_i \rightarrow y_i & & & & & \\
x_i \rightarrow y_i & \searrow & z_i & & & \\
x_{i+1} \rightarrow y_{i+1} & \searrow & f[x_i x_{i+1}] = \frac{y_{i+1}-y_i}{x_{i+1}-x_i} & \searrow & \frac{f[x_i x_{i+1}]-z_i}{x_{i+1}-x_i} & \\
x_{i+1} \rightarrow y_{i+1} & \searrow & z_{i+1} & \searrow & \frac{z_{i+1}-f[x_i x_{i+1}]}{x_{i+1}-x_i} & \searrow \quad \frac{z_{i+1}+z_i-2f[x_i x_{i+1}]}{(x_{i+1}-x_i)^2}
\end{array}$$

Interpolazione di Hermite tramite polinomi cubici a tratti (cont.)

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

$$\widetilde{s}_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^2(x - x_{i+1})$$

$$\begin{aligned}
a_i &= y_i \\
b_i &= z_i \\
c_i &= \frac{f[x_i, x_{i+1}] - z_i}{h_i} \\
d_i &= \frac{z_{i+1} + z_i - 2f[x_i x_{i+1}]}{h_i^2}
\end{aligned}$$

Interpolazione cubica a tratti di Bessel

Si approssimano i valori delle derivate prime nei nodi con la formula

$$z_i = \frac{h_{i-1} f[x_i x_{i+1}] + h_i f[x_{i-1}, x_i]}{h_i + h_{i-1}}$$

ove si aggiungono due nodi

$$\begin{aligned} x_{-1} &= x_0 - h_0 \\ x_{n+2} &= x_{n+1} + h_n \end{aligned}$$

È un modo per approssimare le derivate prime di una funzione incognita nei nodi.

Il polinomio si valuta in ξ con lo schema di Horner. Se $\xi \in (x_i, x_{i+1})$.

$$\begin{cases} p = d_i(\xi - x_{i+1}) + c_i \\ p = p(\xi - x_i) + b_i \\ p = p(\xi - x_i) + a_i \end{cases}$$

Osservazione

L'interpolazione mediante spline lineare determina una funzione di approssimazione non derivabile nei nodi x_i $(i = 0, \ldots, n+1)$.
Nelle applicazioni, spesso si richiede che l'approssimazione sia derivabile. Dunque si preferisce interpolare in ogni intervallo $[x_i, x_{i+1}]$ con polinomi di Hermite di grado 3 essendo noti $f(x_i), f(x_{i+1}), f'(x_i), f'(x_{i+1})$, ottenendo una funzione continua e derivabile.
Tuttavia in questo caso è necessario conoscere $f'(x_i)$, $i = 0, \ldots, n+1$, il che non è sempre possibile.
Ci si pone il problema di trovare un approssimante dato da un polinomio di grado 3 a tratti, che sia almeno continuo e derivabile senza conoscere i valori delle derivate prime nei nodi x_i.
Il problema è risolto dall'interpolazione mediante spline cubiche interpolanti.

Interpolazione con spline cubiche

Teorema

Sia data $\{x_i\}_{i=0,\dots,n+1}$ una partizione dell'intervallo chiuso e limitato $[a, b]$ con $a \leq x_0 < x_1 < \dots < x_n < x_{n+1} \leq b$ e siano $y_0, y_1, ..., y_{n+1}$ assegnati.
Esiste una e una sola spline cubica interpolante in $S\{x_1, \dots, x_n\}$ tale che $s(x_i) = y_i$, $i = 0, 1, \dots, n+1$ e tale che vale una delle seguenti condizioni:

a) $s'(x_0) = z_0$, $s'(x_{n+1}) = z_{n+1}$, con z_0, z_{n+1} assegnati (spline cubica)

b) $s^{''}(x_0) = s^{''}(x_{n+1}) = 0$ (spline naturale)

c)

$$\begin{aligned} y_0 &= y_{n+1} = s(x_0) = s(x_{n+1}) \\ s^{'}(x_0) &= s^{'}(x_{n+1}) \\ s^{''}(x_0) &= s^{''}(x_{n+1}) \end{aligned}$$

(spline periodica di periodo $(x_{n+1} - x_0)$)

Sia $s_i(x)$ il polinomio di grado 3 definito in $[x_i, x_{i+1}]$, $i = 0, 1, \dots, n$.
Sia $h_i = x_{i+1} - x_i$.
Allora

$$s_i(x) = \alpha_i + \beta_i(x - x_i) + \gamma_i(x - x_i)^2 + \delta_i(x - x_i)^3$$

Di conseguenza, si ha

$$\begin{aligned} s_i^{'}(x) &= \beta_i + 2\gamma_i(x - x_i) + 3\delta_i(x - x_i)^2 \\ s_i^{''}(x) &= 2\gamma_i + 6\delta_i(x - x_i) \end{aligned}$$

Perchè $s_i(x)$ e $s_{i+1}(x)$ siano parti di una spline cubica, deve valere nel nodo di raccordo x_{i+1} che $\begin{cases} s_i(x_{i+1}) = s_{i+1}(x_{i+1}) \\ s_i^{'}(x_{i+1}) = s_{i+1}^{'}(x_{i+1}) \\ s_i^{''}(x_{i+1}) = s_{i+1}^{''}(x_{i+1}) \end{cases} \quad i = 0, \dots, n-1$
Se sono noti i valori di y_i e di z_i $(i = 0, \dots, n+1)$ nei nodi allora $s_i(x)$ è completamente determinato, come si è già visto:

$$\left.\begin{aligned} s_i(x_i) = \alpha_i = y_i \\ s_i^{'}(x_i) = \beta_i = z_i \end{aligned}\right. \qquad i = 0, \dots, n+1$$

(cont.)

Infatti vale che

$$\begin{aligned} s_i(x) &= \alpha_i + \beta_i(x-x_i) + \gamma_i(x-x_i)^2 + \delta_i(x-x_i)^3 \\ &= a_i + b_i(x-x_i) + c_i(x-x_i)^2 + d_i(x-x_i)^2(x-x_{i+1}) = \\ &= a_i + b_i(x-x_i) + c_i(x-x_i)^2 + d_i(x-x_i)^3 - d_i h_i (x-x_i)^2 \end{aligned}$$

con

$$a_i = y_i \qquad b_i = z_i$$

$$c_i = \frac{f[x_i, x_{i+1}] - z_i}{h_i} \qquad d_i = \frac{z_{i+1} + z_i - 2f[x_i x_{i+1}]}{h_i^2}$$

Poichè $x - x_{i+1} = x - x_i + x_i - x_{i+1} = x - x_i - h_i$, vale che

$$\begin{aligned} \alpha_i &= a_i = y_i \\ \beta_i &= b_i = z_i \\ \gamma_i &= c_i - d_i h_i = \frac{f[x_i, x_{i+1}] - z_i}{h_i} - d_i h_i \\ \delta_i &= d_i = \frac{z_{i+1} + z_i - 2f[x_i x_{i+1}]}{h_i^2} \end{aligned}$$

Allora per conoscere $s_i(x)$ basta determinare z_i, $i = 0, \dots, n+1$.

(cont.)

Si ricorda che $s_i^{''}(x) = 2\gamma_i + 6\delta_i(x - x_i)$, e che vale

$$\begin{array}{rcl} s_i^{''}(x_{i+1}) = s_{i+1}^{''}(x_{i+1}) & & i = 0, 1, \ldots, n-1 \\ 2\gamma_i + 6\delta_i h_i & = & 2\gamma_{i+1} \\ c_i - d_i h_i + 3 d_i h_i & = & c_{i+1} - d_{i+1} h_{i+1} \end{array}$$

$$\begin{array}{rcl} \dfrac{f[x_i x_{i+1}] - z_i}{h_i} & + & 2\dfrac{z_i + z_{i+1} - 2f[x_i x_{i+1}]}{h_i} \\ & = & \dfrac{f[x_{i+1} x_{i+2}] - z_{i+1}}{h_{i+1}} - \dfrac{z_{i+1} + z_{i+2} - 2f[x_{i+1} x_{i+2}]}{h_{i+1}} \end{array}$$

$$\frac{z_i}{h_i} + 2z_{i+1}\left(\frac{1}{h_i} + \frac{1}{h_{i+1}}\right) + \frac{z_{i+2}}{h_{i+1}} = \frac{3}{h_i} f[x_i x_{i+1}] + \frac{3}{h_{i+1}} f[x_{i+1}, x_{i+2}]$$

Facendo il comune denominatore, si ottiene un sistema di n equazioni nelle incognite $z_0, \ldots, z_{n+1}$.

$$\boxed{\begin{array}{l} h_{i+1} z_i + 2(h_i + h_{i+1}) z_{i+1} + h_i z_{i+2} = 3(h_{i+1} f[x_i x_{i+1}] + h_i f[x_{i+1} x_{i+2}]) = b_{i+1} \\ i = 0, \ldots, n-1 \end{array}}$$

Caso a: z_0, z_{n+1} assegnati

Le equazioni diventano

$$\begin{array}{rcl} 2(h_0+h_1)z_1 + h_0 z_2 & = & b_1 - h_1 z_0 \\ h_2 z_1 + 2(h_1+h_2)z_2 + h_1 z_3 & = & b_2 \\ & \cdots & \\ h_{n-1}z_{n-2} + 2(h_{n-2}+h_{n-1})z_{n-1} + h_{n-1}z_n & = & b_{n-1} \\ h_n z_{n-1} + 2(h_{n-1}+h_n)z_n & = & b_n - h_{n-1}z_{n+1} \end{array}$$

$$Tz = b \qquad z = [z_1, \ldots, z_n]^T$$

$$T = \begin{pmatrix} 2(h_0+h_1) & h_0 & & & \\ h_2 & 2(h_1+h_2) & h_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & h_{n-2} \\ & & & h_n & 2(h_{n-1}+h_n) \end{pmatrix} \qquad b = \begin{pmatrix} -h_1 z_0 + b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ -h_{n-1}z_{n+1} + b_n \end{pmatrix}$$

T è di ordine n, strettamente diagonale dominante e quindi non singolare $\rightarrow$ esiste una e una sola spline cubica interpolante.

Caso b

$$\begin{aligned} s''(x_0) &= 0 \quad s''(x_{n+1}) = 0 \\ 2\gamma_0 &= 0 \quad 2\gamma_n + 6\delta_n h_n = 0 \end{aligned}$$

$$\frac{f[x_0x_1] - z_0}{h_0} - \frac{z_0 + z_1 - 2f[x_0x_1]}{h_0} = 0$$
$$\frac{f[x_nx_{n+1}] - z_n}{h_n} + 2\frac{z_n + z_{n+1} - 2f[x_nx_{n+1}]}{h_n} = 0$$

Allora

$$\begin{aligned} 2z_0 + z_1 &= 3f[x_0x_1] \\ z_n + 2z_{n+1} &= 3f[x_nx_{n+1}] \end{aligned}$$

$$Tz = b \qquad z = [z_0, z_1, \dots, z_n, z_{n+1}]^T$$

$$T = \begin{pmatrix} 2 & 1 & & & \\ h_1 & 2(h_1+h_0) & h_0 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \\ & & h_n & 2(h_{n-1}+h_n) & h_{n-1} \\ & & & 1 & 2 \end{pmatrix} \qquad b = \begin{pmatrix} 3f[x_0x_1] \\ b_1 \\ \vdots \\ b_n \\ 3f[x_nx_{n+1}] \end{pmatrix}$$

Caso b (cont.)

T è di ordine $n + 2$, strettamente diagonale dominante e dunque non singolare.
Esiste una e una sola spline naturale.

Caso c

$$
\begin{aligned}
y_0 &= y_{n+1} \\
s^{'}(x_0) &= s^{'}(x_{n+1}) = z_0 = z_{n+1} \\
s^{''}(x_0) &= s^{''}(x_{n+1}) \\
\gamma_0 &= \gamma_n + 3\delta_n h_n
\end{aligned}
$$

$$
\frac{f[x_0x_1]-z_0}{h_0} - \frac{z_0+z_1-2f[x_0x_1]}{h_0} = \frac{f[x_nx_{n+1}]-z_n}{h_n} + 2\frac{z_n+z_{n+1}-2f[x_nx_{n+1}]}{h_n}
$$

$$
2(h_0+h_n)z_0 + h_nz_1 + h_0z_n = \underbrace{3h_0f[x_nx_{n+1}] + 3h_nf[x_0x_1]}_{=b_0}
$$

$$
Tz = b \qquad z = [z_0, z_1, \ldots, z_n]^T
$$

$$
T = \begin{pmatrix}
2(h_0+h_n) & h_n & & & h_0 \\
h_1 & 2(h_1+h_0) & h_0 & & \\
 & \ddots & \ddots & \ddots & \\
 & & \ddots & \ddots & \\
h_{n-1} & & & h_n & 2(h_{n-1}+h_n)
\end{pmatrix}
\qquad
b = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ \vdots \\ b_n \end{pmatrix}
$$

T è di ordine $n+1$ ed è strettamente diagonale dominante e non singolare. Esiste una sola spline periodica.

Codice Matlab per spline cubiche generali

```
function C=splinemia(x,y,zeta1,zeta2);
%
% Costruzione dei coefficienti di una spline cubica interpolante
% relativa ai nodi x, ai valori della funzione y
% zeta1 e zeta2 sono i valori della derivata prima
% nei nodi inziale e finale
% C e' una matrice con tante righe quanti sono i sottointervalli e
% 4 colonne contenenti i coefficienti dei polinomi
% di grado 3 da quello della potenza 3 a quello della potenza 0
%
% C(i,1)*(x-x_i)^3+C(i,2)*(x-x_i)^2+C(i,3)*(x-x_i)+C(i,4)
%
%    rappresentazione pp
%
n=length(x);
h=x(2:n)-x(1:n-1); % n-1 valori di h
t0=2*(h(1:n-2)+h(2:n-1));  % sistema di ordine n-2
t1=[0;h(1:n-3)];
t_1=[h(3:n-1);0];
```

Codice Matlab per spline cubiche generali (cont.)

```
T=spdiags([t_1,t0,t1],[-1,0,1],n-2,n-2);
b=(y(2:n-1)-y(1:n-2))./h(1:n-2).*h(2:n-1)+...
    h(1:n-2).*(y(3:n)-y(2:n-1))./h(2:n-1);
b=3*b;
b(1)=b(1)-zeta1*h(2);
b(n-2)=b(n-2)-zeta2*h(n-2);
z=T\b; % risoluzione di un sistema tridiagonale
z=[zeta1;z;zeta2];
C=zeros(n-1,4);
C(:,4)=y(1:n-1); % coeff del termine omogeneo
C(:,3)=z(1:n-1); % coeff di x-x_i
C(:,1)=(z(1:n-1)+z(2:n)-2*(y(2:n)-y(1:n-1))./h(1:n-1))./h(1:n-1).^2;
C(:,2)=((y(2:n)-y(1:n-1))./h(1:n-1)-z(1:n-1))./h(1:n-1)-...
    C(:,1).*h(1:n-1);
%%
function p=valspline(C,x,xf);
%%%
% valutazione di una spline basata sulla suddivisione x in xf
% i coefficienti dei polinomi che formano la spline sono i C
%
```

Codice Matlab per spline cubiche generali (cont.)

```
[n,l]=size(C);
for i=1:length(xf)
   ij=find(x>xf(i));
   if isempty(ij) & x(end)~=xf(i)
      error('un elemento di xf cade fuori');
   elseif xf(i)==x(end)
      k=n;
   else
      k=ij(1)-1;
   end;
   p(i)=C(k,1)*(xf(i)-x(k))+C(k,2);
   p(i)=p(i)*(xf(i)-x(k))+C(k,3);
   p(i)=p(i)*(xf(i)-x(k))+C(k,4);
end;
```

La nozione di curvatura

La circonferenza offre il modello più semplice di misura della curvatura: circonferenze con raggio maggiore hanno una curvatura minore, e viceversa. La curvatura della circonferenza viene allora definita come il reciproco del suo raggio R.

$$k = \frac{1}{R}$$

La retta, che si può identificare con la circonferenza di raggio infinito, ha curvatura nulla.
Questa definizione può venire estesa a oggetti più complessi.
Per una generica curva piana, la curvatura varia da punto a punto, e viene definita tramite la costruzione del cerchio osculatore, che è tangente alla curva e la approssima fino al secondo ordine: dato un punto P della curva, la curvatura è il vettore che ha come direzione e verso la retta orientata che va da P al centro del cerchio osculatore e come intensità la curvatura del cerchio osculatore.
Per una curva in forma esplicita $y = f(x)$

$$k = \frac{\frac{d^2 f(x)}{dx^2}}{(1 + (\frac{df(x)}{dx})^2)^{3/2}}$$

se la pendenza della funzione è trascurabile rispetto all'unità, è possibile utilizzare l'approssimazione

$$k \approx \frac{d^2 f(x)}{dx^2}$$

Teoremi sulle spline cubiche

Teorema

Tra tutte le funzioni $f \in C^2[a,b]$ tali che

$$f(x_i) = y_i \quad i = 0, \ldots, n+1, \quad x_0 \equiv a, x_{n+1} \equiv b$$

e tali che vale una di queste condizioni

a) $f^{'}(x_0) = z_0, \qquad f^{'}(x_{n+1}) = z_{n+1}$

b) $f^{''}(x_0) = f^{''}(x_{n+1}) = 0$

c) $f^{(k)}(x_0) = f^{(k)}(x_{n+1}) \qquad k = 0, 1, 2$

la spline cubica di interpolazione per cui vale a) o b) o c) è quella per cui vale la proprietà di minimo

$$\int_a^b (s^{''}(x))^2 dx \leq \int_a^b (f^{''}(x))^2 dx$$

l'uguaglianza vale se e solo se $f \equiv s$.

Dimostrazione. Usando la relazione $(u-v)^2 = u^2 - v^2 - 2v(u-v)$, si ha:

$$\int_a^b (f''(x) - s''(x))^2 dx = \int_a^b (f''(x))^2 dx - \int_a^b (s''(x))^2 dx - 2\int_a^b s''(x)(f''(x) - s''(x))dx$$

Integrando per parti l'ultimo termine, si ha:

$$\int_a^b s''(x)(f''(x) - s''(x))dx = \underbrace{s''(x)(f'(x) - s'(x))\Big|_a^b}_{=0 \quad per \quad a)b)c)} - \int_a^b s'''(x)(f'(x) - s'(x))dx$$

$$= \quad -\sum_{k=0}^{n} s'''(x)\underbrace{(f(x) - s(x))}_{=0}\Big|_{x_k}^{x_{k+1}} + \int_a^b \underbrace{s^{IV}(x)}_{=0}(f(x) - s(x))dx = 0$$

$$\int_a^b (s''(x))^2 dx \leq \int_a^b (s''(x))^2 dx + \int_a^b (f''(x) - s''(x))^2 dx = \int_a^b (f''(x))^2 dx$$

Inoltre, se $f = s \rightarrow$ vale l'uguaglianza.
Viceversa, se vale l'uguaglianza $\int_a^b (f''(x))^2 dx = \int_a^b (s''(x))^2 dx$, allora
$\int_a^b (f''(x) - s''(x))^2 dx = 0$ e dunque
$f(x) - s(x) = mx + q$.

Poiché $f(x_0) = s(x_0)$ e $f(x_{n+1}) = s(x_{n+1}) \rightarrow \begin{cases} mx_0 + q = 0 \\ mx_{n+1} + q = 0 \end{cases} \rightarrow m = q = 0.$

Teorema

Sia $f \in C^2[a,b]$ e sia $s(x)$ la spline cubica interpolante f.

$$\left.\begin{array}{l} |f(x) - s(x)| \leq h^{3/2}\left(\int_a^b (f''(x))^2 dx \right)^{1/2} \\ \\ |f'(x) - s'(x)| \leq h^{1/2}\left(\int_a^b (f''(x))^2 dx \right)^{1/2} \end{array}\right. \qquad x \in [a,b]$$

$h = \max_{i=0,\dots,n}(x_{i+1} - x_i)$.

Poiché $|f''(x)| \leq M$, $x \in [a,b]$, $\forall f \in C^2[a,b]$, segue che

$$|f(x) - s(x)| \leq h^{3/2} M \sqrt{b-a}$$

Pertanto fissato $\epsilon \geq h^{3/2} M \sqrt{b-a}$, $s(x)$ è approssimazione di $f(x)$ in $[a,b]$ entro la tolleranza ϵ.

Uso della funzione spline di Matlab

```
y=spline(x,y,x1);
```

- La funzione spline con tre parametri di input valuta la spline cubica relativa ai dati (x, y) nel vettore di punti $x1$. La spline che viene calcolata è quella per cui il secondo nodo e il penultimo nodo della suddivisione su cui è basata la spline godono di una condizione "not-a-knot"; in tali nodi la derivata terza da destra e da sinistra coincidono e quindi la spline nei due intervalli interessati è rappresentata dallo stesso polinomio di grado 3.
- Se la funzione spline viene usata con due parametri $S = spline(x, y)$, il parametro di ritorno S è una struttura che permette di descrivere la spline (pp-representation):

Uso della funzione spline di Matlab (cont.)

```
x=0:5;
y=x.*exp(-(x-1).^2);
S=spline(x,y)
   S = form: 'pp'
   breaks: [0 1 2 3 4 5]
   coefs: [5x4 double]
   pieces: 5
   order: 4
   dim: 1
```

breaks è il vettore dei nodi, *coefs* è una matrice con tante righe quanti sono i sottointervalli e un numero di colonne pari al numero dei coefficienti del polinomio che rappresenta la spline in ogni sottointervallo, nella forma:

$$coefs(i,1)*(x-breaks(i))^3+coefs(i,2)*(x-breaks(i))^2+coefs(i,3)*(x-breaks(i))+coefs(i,4)$$

ordinati dal coefficiente della potenza di grado più alto a quello del termine di grado 0; *pieces* è il numero di sottointervalli, *dim* è il numero di variabili indipendenti e *order* è l'ordine della spline.

Uso della funzione spline di Matlab (cont.)

Per estrarre le componenti della rappresentazione *pp* si usa la funzione:

```
[x,C,l,k]=unmkpp(S);
```

ove x è il vettore dei nodi della suddivisione, C è la matrice $l \times k$ dei coefficienti, k è l'ordine della spline e l è il numero dei sottointervalli.
Viceversa, fornendo x e C si costruisce la rappresentazione *pp* della spline corrispondente:

```
S=mkpp(x,C);
```

Infine per valutare una spline di cui si conosce la rappresentazione *pp* in un vettore *xf* di nodi si usa:

```
pf=ppval(S,xf)
```