SPLINE LINEARI

Dati i vettori di punti

$$x_0 < x_1 \cdots < x_n$$

la spline lineare che interpola tali dati è una funzione continua e lineare a tratti del tipo:

$$s_1(x) = \begin{cases} c_{1,1}(x - x_0) + c_{1,2} & \text{se } x \in [x_0, x_1] \\ c_{2,1}(x - x_1) + c_{2,2} & \text{se } x \in [x_1, x_2] \\ \dots & \dots \\ c_{n,1}(x - x_{n-1}) + c_{n,2} & \text{se } x \in [x_{n-1}, x_n] \end{cases}$$

e tale che

$$s_1(x_i) = y_i \quad i = 0, \dots n,$$

Come la calcolo con Matlab?
Possiamo utilizzare la funzione `interp1()` nel modo seguente:

```
>> s1z=interp1(x,y,z,'linear')
```

dove `z` è un vettore di punti di campionamento nell'intervallo $[x_0, x_n]$; allora `s1z` sarà un vettore contenente in ogni componente i valori assunti dalla spline nei punti di `z`, ovvero $s_1(z_i), \quad i = 1 \dots \text{length}(z)$.

SPLINE CUBICHE

Dati i vettori di punti

$$x_0 < x_1 \cdots < x_n$$

una spline cubica che interpola tali dati è una funzione $s \in C^2([x_0, x_n])$ polinomiale a tratti di grado 3 su ogni intervallino $I_{i+1} = [x_i, x_{i+1}]$

$$s_3(x) = \begin{cases} c_{1,1}(x-x_0)^3 + c_{1,2}(x-x_0)^2 + c_{1,3}(x-x_0) + c_{1,4} & x \in I_1 \\ c_{2,1}(x-x_1)^3 + c_{2,2}(x-x_1)^2 + c_{2,3}(x-x_1) + c_{2,4} & x \in I_2 \\ \ldots & \ldots \\ c_{n,1}(x-x_{n-1})^3 + c_{n,2}(x-x_{n-1})^2 + c_{n,3}(x-x_{n-1}) + c_{n,4} & x \in I_n \end{cases} \qquad (1)$$

e tale che

$$s_1(x_i) = y_i \quad i = 0, \ldots n,$$

Esistono diversi tipi di spline cubiche interpolanti: *naturali*, *vincolate*, *periodiche*, *not-a-knot*. Con Matlab si calcolano facilmente le spline cubiche interpolanti *not-a-knot* attraverso la funzione predefinita `spline()` che può essere chiamata in due modi:

```
>> s3z=spline(x,y,z)
```

dove `x,y` sono i vettori dei dati da interpolare mentre `z` è un vettore di punti di campionamento nell'intervallo $[x_0, x_n]$ in cui valutare la spline cubica interpolante ad esempio per farne il grafico.

oppure

```
>> s3=spline(x,y)
```

che restituisce la spline in 'pp-form'.

In questo caso per disegnarne il grafico occore ancora valutare la spline cubica nel vettore di punti di campionamento

```
>>s3z=ppval(s3,z);
```

allora `s3z` sarà un vettore contenente in ogni componente i valori assunti dalla spline nei punti di `z`, ovvero $s_3(z_i), \quad i = 1 \ldots \mathrm{length}(z)$.
Per vedere com'è fatta una spline in formato 'pp-form':

```
>> [x,C,n,k]=unmkpp(s3);
```

le variabili restituite hanno i seguenti significati:

- $x =$ vettore dei nodi $(x_0, x_1, \ldots, x_n)$
- $C = (c_{i,j})_{i=1,\ldots n; j=1,\ldots 4}$ matrice dei coefficienti dei polinomi di grado 3 su ogni sottointervallo in (1)
- $n =$ numero sottointervalli
- $k = 4$ (grado delle funzioni polinomiali a tratti)+1

Viceversa per costruire una spline in 'pp-form' a partire dal vettore dei nodi `x` e dalla matrice dei coefficienti C, se nota, si può usare il comando

```
>> s=mkpp(x,C)
```

VARIABILE STRUTTURATA

$$\begin{array}{ccccccc} & I_1 & & I_2 & & I_3 & \\ x_0 & & x_1 & & x_2 & & x_3 \end{array}$$

form : 'pp'

breaks: $[x_0 \;\; x_1 \;\; x_2 \;\; x_3]$ vettore

coefs : $[\,3 \times 4\,]$ matrice

pieces : 3 ($n°$ intervalli $= n°$ righe)

order : 4 (grado+1), 4 coefficienti $\forall$ intervallo $\rightarrow \mathbb{P}_3$
$= n°$ colonne

dim : 1 $\quad (\mathbb{R} \to \mathbb{R})$

ATTENZIONE: in MATLAB i modi sono numerati da 1 a n

$$s_j(x) = c_{j,1}(x-x_j)^3 + c_{j,2}(x-x_j)^2 + c_{j,3}(x-x_j) + c_{j,4}$$

$$\begin{array}{ccc} x_j & \text{———} & x_{j+1} \end{array}$$

$$\left[\begin{array}{c|c|c|c} c_{1,1} & c_{1,2} & c_{1,3} & c_{1,4} \\ c_{2,1} & c_{2,2} & c_{2,3} & c_{2,4} \\ c_{3,1} & c_{3,2} & c_{3,3} & c_{3,4} \end{array}\right] \begin{array}{l} \rightarrow I_1 = [x_0, x_1] \\ \rightarrow I_2 = [x_1, x_2] \\ \rightarrow I_3 = [x_2, x_3] \end{array}$$

$$\begin{array}{c|c|c|c} (x-x_j)^3 & (x-x_j)^2 & (x-x_j)^1 & (x-x_j)^0 \end{array}$$

RIEPILOGO (x_i, y_i), $i = 1, \dots, n$ z_j, $j = 1, \dots, 1000$

$\downarrow$

zval

SPLINE LINEARI

1) z1 = interp1 (x, y, zval, 'linear'); spline_lineare_a.m

2) s1_funz = interp1 (x, y, 'linear', 'pp');

z1 = ppval (s1_funz, zval); spline_lineare_b.m

SPLINE CUBICHE

1) z3 = spline (x, y, zval) spline_cubiche_a.m

2) s3_funz = spline (x, y);

z3 = ppval (s3_funz, zval); spline_cubiche_b.m