

**CAP. 8d – METODI D'INTERPOLAZIONE**

In un elemento finito la cui formulazione si basa sul metodo degli spostamenti, lo spostamento di ogni suo punto è funzione delle coordinate del punto e degli spostamenti nodali dell'elemento. In altre parole, gli spostamenti  $u(x), v(x), w(x)$  sono funzione degli spostamenti nodali attraverso un meccanismo d'interpolazione. Anche gli elementi finiti che si basano su un'altra formulazione, come per esempio il metodo degli sforzi, definiscono le quantità di campo in termini di valori nodali.

Supponiamo di conoscere i valori che assume la variabile  $u$  in  $n$  punti ( $u_1, u_2, \dots, u_n$ ). Non è necessario che i punti siano disposti nello spazio a passo costante. I punti possono essere usati per definire un polinomio di grado  $n - 1$  imponendo che la curva passi su ognuno di essi. Nel caso più semplice, in cui  $n = 2$ , abbiamo a che fare con un'interpolazione lineare.

L'interpolazione di **Lagrange** consente di generare una curva interpolatrice nel modo seguente:

$$u(x) = \sum_i N_i(x) \cdot u_i = N_1(x) \cdot u_1 + N_2(x) \cdot u_2 + \dots + N_n(x) \cdot u_n \quad [8d.1]$$

in cui

$$N_i(x) = \frac{\prod_{j=1, j \neq i}^n (x - x_j)}{\prod_{j=1, j \neq i}^n (x_i - x_j)} \quad [8d.2]$$

Ognuna delle funzioni  $N_i$ , dette talvolta FUNZIONI DI MISCELAMENTO altre volte FUNZIONI DI FORMA, è un polinomio di grado  $n - 1$ . E' importante notare che, perché la curva risulti un'interpolante, deve avvenire che:

$$\begin{aligned} \text{per } x = x_i, \quad N_i &= 1 \\ \text{per } x = x_j, \quad N_i &= 0 \quad \text{quando } i \neq j \end{aligned}$$

Per una qualsiasi coordinata  $x$  compresa tra due punti dove è noto il valore di  $u$ ,  $u$  è funzione di tutte le  $u_i$ . Per calcolare la pendenza della curva in un punto di coordinata  $x$  qualsiasi, possiamo scrivere:

$$\vartheta(x) = \frac{du(x)}{dx} = \sum_i \frac{dN_i(x)}{dx} \cdot u_i$$

La "curva esatta", se esiste, può non coincidere esattamente con la curva (8d.1). Nel contesto del metodo degli elementi finiti, questa affermazione significa che una soluzione agli elementi finiti può essere leggermente diversa da quella fornita dalla soluzione esatta della teoria dell'elasticità.

**Interpolazione lineare con  $n = 2$ .**

$$u(x) = \sum_i N_i(x) \cdot u_i = N_1(x) \cdot u_1 + N_2(x) \cdot u_2$$

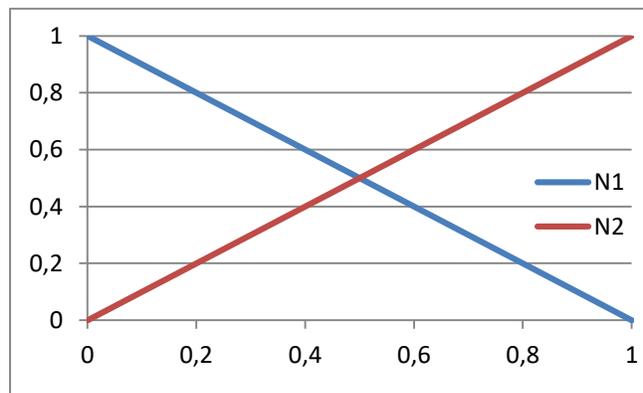
$$\text{dove: } N_1(x) = \frac{\prod_{j=1, j \neq 1}^2 (x - x_j)}{\prod_{j=1, j \neq 1}^2 (x_1 - x_j)} = \frac{x - x_2}{x_1 - x_2}$$

$$N_2(x) = \frac{\prod_{j=1, j \neq 2}^2 (x - x_j)}{\prod_{j=1, j \neq 2}^2 (x_2 - x_j)} = \frac{x - x_1}{x_2 - x_1}$$

da cui:

$$u(x) = \frac{x - x_2}{x_1 - x_2} \cdot u_1 + \frac{x - x_1}{x_2 - x_1} \cdot u_2$$

$$\vartheta(x) = \frac{1}{x_1 - x_2} \cdot u_1 + \frac{1}{x_2 - x_1} \cdot u_2 = \frac{u_2 - u_1}{x_2 - x_1}$$



Funzioni di miscelamento per  $n = 2$  con  $x_1 = 0$  e  $x_2 = 1$ .



Abbiamo inoltre che:

$$N_1(x) + N_2(x) = \frac{x - x_2}{x_1 - x_2} + \frac{x - x_1}{x_2 - x_1} = \frac{x_2 - x + x - x_1}{x_2 - x_1} = 1$$

**Interpolazione parabolica con  $n = 3$ .**

$$u(x) = \sum_i N_i \cdot u_i = N_1(x) \cdot u_1 + N_2(x) \cdot u_2 + N_3(x) \cdot u_3$$

$$N_1(x) = \frac{\prod_{\substack{j=1 \\ j \neq 1}}^3 (x - x_j)}{\prod_{\substack{j=1 \\ j \neq 1}}^3 (x_1 - x_j)} = \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} \quad ; \quad \frac{dN_1(x)}{dx} = \frac{(x - x_2) + (x - x_3)}{(x_1 - x_2)(x_1 - x_3)}$$

$$N_2(x) = \frac{\prod_{\substack{j=1 \\ j \neq 2}}^3 (x - x_j)}{\prod_{\substack{j=1 \\ j \neq 2}}^3 (x_2 - x_j)} = \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)} \quad ; \quad \frac{dN_2(x)}{dx} = \frac{(x - x_1) + (x - x_3)}{(x_2 - x_1)(x_2 - x_3)}$$

$$N_3(x) = \frac{\prod_{\substack{j=1 \\ j \neq 3}}^3 (x - x_j)}{\prod_{\substack{j=1 \\ j \neq 3}}^3 (x_3 - x_j)} = \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)} \quad ; \quad \frac{dN_3(x)}{dx} = \frac{(x - x_1) + (x - x_2)}{(x_3 - x_1)(x_3 - x_2)}$$

Abbiamo inoltre che:

$$\begin{aligned} N_1(x) + N_2(x) + N_3(x) &= \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} + \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)} + \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)} = \\ &= \frac{(x - x_2)(x - x_3)(x_3 - x_2) - (x - x_1)(x - x_3)(x_3 - x_1) + (x - x_1)(x - x_2)(x_2 - x_1)}{(x_2 - x_1)(x_3 - x_1)(x_3 - x_2)} \end{aligned}$$

Sviluppiamo i tre termini del numeratore abbiamo:

$$(x - x_2)(x - x_3)(x_3 - x_2) = x_3x^2 - (x_2x_3 + x_3^2)x + x_2x_3^2 - x_2x^2 + (x_2^2 + x_2x_3)x - x_2^2x_3$$

$$(x - x_1)(x - x_3)(x_3 - x_1) = x_3x^2 - (x_1x_3 + x_3^2)x + x_1x_3^2 - x_1x^2 + (x_1^2 + x_1x_3)x - x_1^2x_3$$

$$(x - x_1)(x - x_2)(x_2 - x_1) = x_2x^2 - (x_1x_2 + x_2^2)x + x_1x_2^2 - x_1x^2 + (x_1^2 + x_1x_2)x - x_1^2x_2$$

Ricordando che il secondo termine deve essere moltiplicato per -1, eseguiamo la somma e cerchiamo i coefficienti che moltiplicano  $x^2$ ,  $x$  ed il termine noto:

$$[x_3 - x_2 - x_3 + x_1 + x_2 - x_1]x^2 = 0 \quad \text{per ogni } x$$

$$(-x_1^2 + x_1^2 - x_3^2 + x_3^2 + x_2^2 - x_2^2 + x_1x_3 - x_1x_3 - x_1x_2 + x_1x_2 - x_2x_3 + x_2x_3)x = 0 \quad \text{per ogni } x$$

$$-x_2^2x_3 + x_2x_3^2 - x_1x_3^2 + x_1^2x_3 - x_1^2x_2 + x_1x_2^2 = (x_2 - x_1)(x_3 - x_1)(x_3 - x_2)$$

Abbiamo quindi che:

$$\sum_i N_i(x) = N_1(x) + N_2(x) + N_3(x) = 1$$

**Interpolazione cubica con  $n = 4$ .**

$$u(x) = N_1(x) \cdot u_1 + N_2(x) \cdot u_2 + N_3(x) \cdot u_3 + N_4(x) \cdot u_4$$

$$N_1(x) = \frac{(x - x_2)(x - x_3)(x - x_4)}{(x_1 - x_2)(x_1 - x_3)(x_1 - x_4)}$$

$$N_2(x) = \frac{(x - x_1)(x - x_3)(x - x_4)}{(x_2 - x_1)(x_2 - x_3)(x_2 - x_4)}$$

$$N_3(x) = \frac{(x - x_1)(x - x_2)(x - x_4)}{(x_3 - x_1)(x_3 - x_2)(x_3 - x_4)}$$

$$N_4(x) = \frac{(x - x_1)(x - x_2)(x - x_3)}{(x_4 - x_1)(x_4 - x_2)(x_4 - x_3)}$$

Spesso è preferibile utilizzare un sistema di riferimento nella variabile  $\xi$  normalizzata:

$$(\xi_2 - \xi_1) : (x_2 - x_1) = (\xi - \xi_1) : (x - x_1)$$

Il passaggio da un sistema di riferimento all'altro si ottiene usando le seguenti relazioni:

$$\xi = \frac{\xi_2 - \xi_1}{x_2 - x_1} \cdot (x - x_1) + \xi_1 \quad ; \quad x = \frac{(x_2 - x_1)}{(\xi_2 - \xi_1)} (\xi - \xi_1) + x_1$$

Nel caso in cui la coordinata normalizzata  $\xi$  sia definita nell'intervallo  $0 \leq \xi \leq 1$  abbiamo:

$$\xi = \frac{x - x_1}{x_2 - x_1} \quad ; \quad x = (x_2 - x_1)\xi + x_1$$

Nel caso in cui la coordinata normalizzata  $\xi$  sia definita nell'intervallo  $-1 \leq \xi \leq 1$  abbiamo:

$$\xi = \frac{2(x - x_1)}{x_2 - x_1} - 1 \quad ; \quad x = \frac{x_2 - x_1}{2} \xi + \frac{x_2 + x_1}{2}$$

Nel sistema normalizzato possiamo scrivere la funzione interpolante:

$$u(\xi) = \sum_i N_i(\xi) \cdot u_i = N_1(\xi) \cdot u_1 + N_2(\xi) \cdot u_2$$

La pendenza della curva  $u = u(x)$  vale:

$$\vartheta(x) = \frac{du(x)}{dx} = \sum_i \frac{dN_i(x)}{dx} \cdot u_i$$

dove:

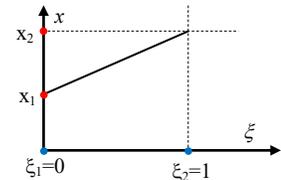
$$\frac{dN_i(\xi)}{d\xi} = \frac{dN_i(x)}{dx} \cdot \frac{dx}{d\xi} \quad \text{da cui} \quad \frac{dN_i(x)}{dx} = \frac{dN_i(\xi)}{d\xi} \cdot \frac{d\xi}{dx}$$

Se la coordinata normalizzata  $\xi$  è definita nell'intervallo  $0 \leq \xi \leq 1$  abbiamo:

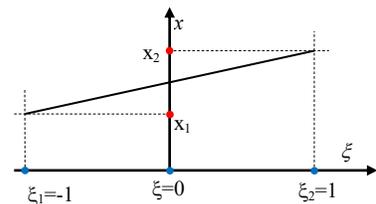
$$N_1(\xi) = 1 - \xi \quad ; \quad N_2(\xi) = \xi \quad ; \quad \frac{dx}{d\xi} = x_2 - x_1$$

$$\frac{dN_1(x)}{dx} = \frac{dN_1(\xi)}{d\xi} \cdot \frac{d\xi}{dx} = \frac{1 - \xi}{x_2 - x_1} \quad ; \quad \frac{dN_2(x)}{dx} = \frac{dN_2(\xi)}{d\xi} \cdot \frac{d\xi}{dx} = \frac{\xi}{x_2 - x_1}$$

se invece la coordinata normalizzata  $\xi$  è definita nell'intervallo  $-1 \leq \xi \leq 1$  abbiamo:



Mapping da  $\xi [0 \leq \xi \leq 1]$  a  $x$ .



Mapping da  $\xi [-1 \leq \xi \leq 1]$  a  $x$ .



$$N_1(\xi) = \frac{1-\xi}{2} \quad ; \quad N_2(\xi) = \frac{1+\xi}{2} \quad ; \quad \frac{dx}{d\xi} = \frac{x_2-x_1}{2}$$

$$\frac{dN_1(x)}{dx} = \frac{dN_1(\xi)}{d\xi} \frac{d\xi}{dx} = \frac{1-\xi}{x_2-x_1} \quad ; \quad \frac{dN_2(x)}{dx} = \frac{dN_2(\xi)}{d\xi} \frac{d\xi}{dx} = \frac{1+\xi}{x_2-x_1}$$

**Interpolazione parabolica con n = 3.**

$$u(\xi) = \sum_i N_i \cdot u_i = N_1(\xi) \cdot u_1 + N_2(\xi) \cdot u_2 + N_3(\xi) \cdot u_3$$

Ponendo  $-1 \leq \xi \leq 1$ , abbiamo:

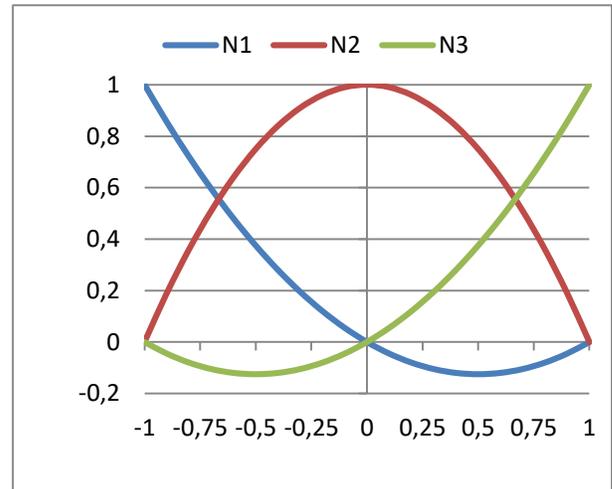
$$\xi_1 = -1 \quad ; \quad \xi_2 = 0 \quad ; \quad \xi_3 = 1 \quad \text{da cui:}$$

$$N_1(\xi) = \frac{\prod_{j \neq 1}^3 (\xi - \xi_j)}{\prod_{j \neq 1}^3 (\xi_1 - \xi_j)} = \frac{(\xi - \xi_2)(\xi - \xi_3)}{(\xi_1 - \xi_2)(\xi_1 - \xi_3)} = \frac{\xi^2 - \xi}{2}$$

$$N_2(\xi) = \frac{\prod_{j \neq 2}^3 (\xi - \xi_j)}{\prod_{j \neq 2}^3 (\xi_2 - \xi_j)} = \frac{(\xi - \xi_1)(\xi - \xi_3)}{(\xi_2 - \xi_1)(\xi_2 - \xi_3)} = 1 - \xi^2$$

$$N_3(\xi) = \frac{\prod_{j \neq 3}^3 (\xi - \xi_j)}{\prod_{j \neq 3}^3 (\xi_3 - \xi_j)} = \frac{(\xi - \xi_1)(\xi - \xi_2)}{(\xi_3 - \xi_1)(\xi_3 - \xi_2)} = \frac{\xi^2 + \xi}{2}$$

$$u(\xi) = \frac{\xi^2 - \xi}{2} \cdot u_1 + (1 - \xi^2) \cdot u_2 + \frac{\xi^2 + \xi}{2} \cdot u_3$$



Funzioni di miscelamento per n = 3 con  $-1 \leq \xi \leq 1$

**Interpolazione cubica con n = 4.**

$$u(\xi) = N_1(\xi) \cdot u_1 + N_2(\xi) \cdot u_2 + N_3(\xi) \cdot u_3 + N_4(\xi) \cdot u_4$$

Ponendo  $-1 \leq \xi \leq 1$ , abbiamo:

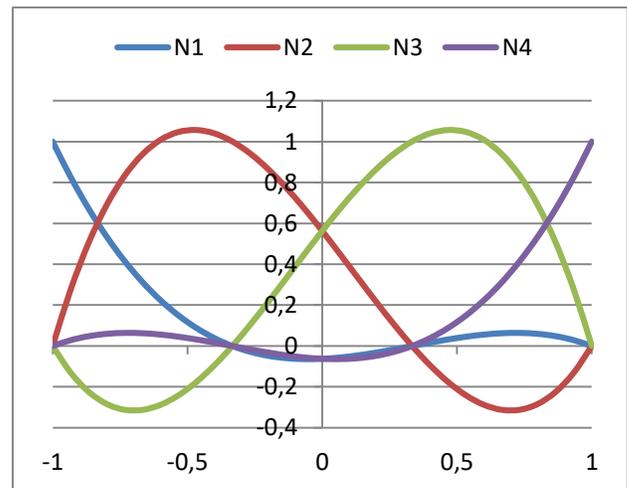
$$\xi_1 = -1 \quad ; \quad \xi_2 = -\frac{1}{3} \quad ; \quad \xi_3 = \frac{1}{3} \quad ; \quad \xi_4 = 1 \quad \text{da cui:}$$

$$N_1(\xi) = \frac{(\xi - \xi_2)(\xi - \xi_3)(\xi - \xi_4)}{(\xi_1 - \xi_2)(\xi_1 - \xi_3)(\xi_1 - \xi_4)} = \frac{(-9\xi^3 + 9\xi^2 + \xi - 1)}{16}$$

$$N_2(\xi) = \frac{(\xi - \xi_1)(\xi - \xi_3)(\xi - \xi_4)}{(\xi_2 - \xi_1)(\xi_2 - \xi_3)(\xi_2 - \xi_4)} = \frac{9}{16}(3\xi^3 - \xi^2 - 3\xi + 1)$$

$$N_3(\xi) = \frac{(\xi - \xi_1)(\xi - \xi_2)(\xi - \xi_4)}{(\xi_3 - \xi_1)(\xi_3 - \xi_2)(\xi_3 - \xi_4)} = \frac{(-9)}{16}(3\xi^3 + \xi^2 - 3\xi - 1)$$

$$N_4(\xi) = \frac{(\xi - \xi_1)(\xi - \xi_2)(\xi - \xi_3)}{(\xi_4 - \xi_1)(\xi_4 - \xi_2)(\xi_4 - \xi_3)} = \frac{(9\xi^3 + 9\xi^2 - \xi - 1)}{16}$$



Funzioni di miscelamento per n = 4 con  $-1 \leq \xi \leq 1$

**ESEMPIO DI INTERPOLAZIONE DI LAGRANGE**

Immaginiamo di estrarre dalla seguente funzione 4 punti a passo  $x$  costante:

$$y(x) = x + \sin\left(\frac{2\pi}{L}x\right) \quad \text{con} \quad 0 \leq x < L$$

in cui abbiamo posto  $L = 4$  e  $\Delta L = 1$ .

| i | x | $\xi$ | y |
|---|---|-------|---|
| 1 | 0 | -1    | 0 |
| 2 | 1 | -1/3  | 2 |
| 3 | 2 | 1/3   | 2 |
| 4 | 3 | 1     | 2 |

Usiamo l'interpolazione cubica di Lagrange con  $n = 4$ .

L'equazione interpolante sarà la seguente:

$$u(\xi) = \sum_i N_i \cdot u_i = N_1(\xi) \cdot u_1 + N_2(\xi) \cdot u_2 + N_3(\xi) \cdot u_3 + N_4(\xi) \cdot u_4$$

Sostituendo le funzioni di miscelamento abbiamo:

$$u(\xi) = \frac{(-9\xi^3 + 9\xi^2 + \xi - 1)}{16} \cdot u_1 + \frac{9}{16}(3\xi^3 - \xi^2 - 3\xi + 1) \cdot u_2 - \frac{9}{16}(3\xi^3 + \xi^2 - 3\xi - 1) \cdot u_3 + \frac{(9\xi^3 + 9\xi^2 - \xi - 1)}{16} \cdot u_4$$

La sua derivata vale:

$$\vartheta(\xi) = \frac{du(\xi)}{d\xi} = \frac{(-27\xi^2 + 18\xi + 1)}{16} \cdot u_1 + \frac{9}{16}(9\xi^2 - 2\xi - 3) \cdot u_2 - \frac{9}{16}(9\xi^2 + 2\xi - 3) \cdot u_3 + \frac{(27\xi^2 + 18\xi - 1)}{16} \cdot u_4$$

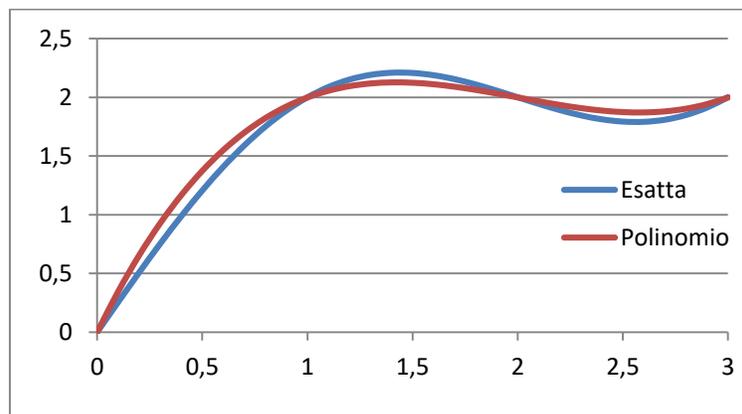
La tangente alla curva si trova calcolando la derivata della funzione:  $\vartheta(x) = \frac{du(x)}{dx} = \sum_i \frac{dN_i(x)}{dx} \cdot u_i$

dove: 
$$\frac{dN_i(x)}{dx} = \frac{dN_i(\xi)}{d\xi} \cdot \frac{d\xi}{dx}$$

da cui:

$$\frac{dN_1(x)}{dx} = \frac{(-27\xi^2 + 18\xi + 1)}{8(x_2 - x_1)} \quad ; \quad \frac{dN_2(x)}{dx} = \frac{9(9\xi^2 - 2\xi - 3)}{8(x_2 - x_1)} \quad ; \quad \frac{dN_3(x)}{dx} = -\frac{9(9\xi^2 + 2\xi - 3)}{8(x_2 - x_1)} \quad ; \quad \frac{dN_4(x)}{dx} = \frac{(27\xi^2 + 18\xi - 1)}{8(x_2 - x_1)}$$

$$\vartheta(x) = \frac{du(x)}{dx} = \frac{(-27\xi^2 + 18\xi + 1)}{8(x_2 - x_1)} \cdot u_1 + \frac{9(9\xi^2 - 2\xi - 3)}{8(x_2 - x_1)} \cdot u_2 - \frac{9(9\xi^2 + 2\xi - 3)}{8(x_2 - x_1)} \cdot u_3 + \frac{(27\xi^2 + 18\xi - 1)}{8(x_2 - x_1)} \cdot u_4$$

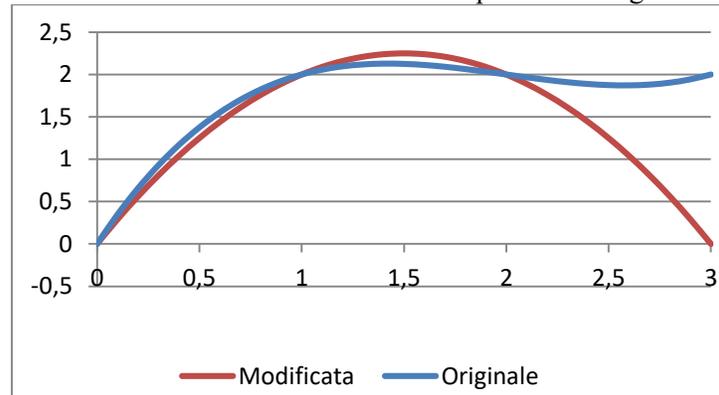


Funzione esatta ed interpolata

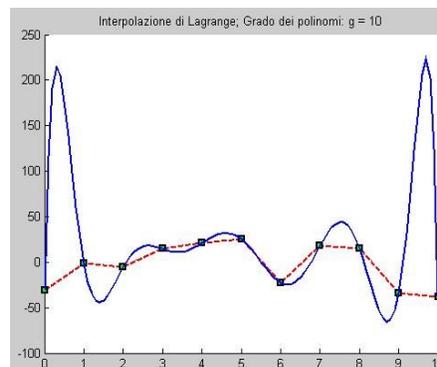
Il valore che assume la funzione in una qualsiasi coordinata  $x$  dipende dai valori di tutti i dati, anche quelli più lontani, ognuno pesato per la relativa funzione di miscelamento o funzione peso. La variazione di un qualsiasi dato  $u_i$ , modifica tutta la curva, anche a grande distanza dal punto considerato.



Modifichiamo per esempio il valore che la funzione assume alla coordinata  $x = 3$  ponendo  $y(x = 3) = 0$  e disegniamo l'intera curva mettendola a confronto con l'interpolazione originale.



E' importante notare che man mano che i punti  $n$  da interpolare aumentano, cresce di conseguenza l'ordine del polinomio ( $n - 1$ ) il che dà origine a funzioni poco regolari, ovvero molto oscillanti. Per questo motivo spesso vengono interpolati pochi punti per volta: si parla allora d'interpolazione polinomiale per tratti (*piecewise*). Nell'esempio precedente avremmo potuto utilizzare un'interpolazione lineare, collegando successivamente copie di punti.



### Codice in MATLAB per l'interpolazione di una funzione con i polinomi di Lagrange

```
% =====  
% Interpolazione di Lagrange per tratti(piecewise)  
% =====  
disp('Interpolazione di Lagrange');  
disp('Scegli la funzione da interpolare');  
disp('f = 0 Casuale');  
disp('f = 1 Sinusoidale');  
disp('f = 2 Esponenziale');  
disp('f = 3 Logaritmica');  
funzione = input('f = ');  
funzione = max(0,min(funzione,3));  
if funzione == 0  
    xdati = 0:1:10;  
    ydati = 100*(rand(size(xdati)) - 0.5);  
elseif funzione == 1  
    xdati = 0:pi/45:10*pi;  
    ydati = sin(xdati);  
elseif funzione == 2  
    xdati = 0:0.05:2;  
    ydati = 100*exp(xdati);  
elseif funzione == 3  
    xdati = 1:1:20;  
    ydati = 100*log(xdati);
```



```
end
[i,Np] = size(xdati);

disp('Scegli il grado dei polinomi: grado < 10');
disp('N = 0 interpolazione di grado Np-1');
disp('N = 1 interpolazione lineare');
disp('N = 2 interpolazione quadratica');
disp('N = 3 interpolazione cubica');
grado = input('N = ');
if grado < 0
    grado = 1;
elseif grado == 0
    grado = Np - 1;
end
grado = min(10,min(grado,Np-1));
np = grado + 1;
Grado = grado;
Ndiv = 10;
NpTot = (Np-1)*Ndiv+1;
x = zeros(NpTot,1);
y = zeros(NpTot,1);
clf;
titolo = sprintf('Interpolazione di Lagrange; Grado dei polinomi: g = %2u',grado);
title(titolo);
hold on
n = 0;
for k = 1:grado:Np-1
    k1 = k + grado;
    if k1 > Np
        k1 = Np;
        np = k1 - k + 1; % se necessario riduci il grado del polinomio
        grado = np - 1;
    end
    xp = xdati(k:k1);
    yp = ydati(k:k1);
    plot(xp,yp,'--rs','LineWidth',2,'MarkerEdgeColor','k',...
        'MarkerFaceColor','g','MarkerSize',5)
    for i = 1:np-1 % Interpolazione nell'intervallo
        dx = (xp(i+1)-xp(i))/Ndiv;
        xinc = xp(i) - dx; % Coordinata incrementale
        for j = 1:Ndiv+1
            n = n + 1;
            xinc = xinc + dx;
            N = Lagrange(np, xp, xinc);
            x(n) = xinc;
            y(n) = yp(1:np)*N(1:np);
        end
    end
end
end
plot(x(1:n),y(1:n),'LineWidth',2);
input('... <Return> per continuare ...');
shg;

% Calcola e disegna le funzioni di miscelamento
np = Grado + 1;
if np == 2
    Ndiv = 1;
else
    Ndiv = 20;
end
xN = 0:1/Grado:1;
NpFF = Grado*Ndiv + 1;
passo = 1/(Ndiv*Grado);
xF = zeros(NpFF,1);
FF = zeros(NpFF,np);
xFinc = -passo;
for n = 1:NpFF
    xFinc = xFinc + passo;
    xF(n) = xFinc;
    FF(n,:) = Lagrange(np, xN, xFinc);
end
```



```

end
hold off
clf;
hold on
title('Funzioni di miscelamento');
ymin = min(min(FF));
ymax = max(max(FF));
axis ([0 1 ymin ymax]);
hold on
for i = 1:np
    ic = mod(i,2);
    if ic == 0
        colore = 'r';
    elseif ic == 1
        colore = 'b';
    end
    plot(xF,FF(:,i), 'Color',colore);
    shg;
    input('... <Return> per continuare ...');
end
hold off
close;

function N = Lagrange (Np, xp, x)
%Lagrange Calcola le funzioni di forma secondo Lagrange
% Np: N.punti
% xp: ascissa
% yp: ordinata
% x: coordinata dove valutare le funzioni di forma
N = zeros(Np,1);
for i = 1:Np
    num = 1;
    den = 1;
    for j=1:Np
        if j ~= i
            num = num*(x-xp(j));
            den = den*(xp(i)-xp(j));
        end
    end
    N(i) = num/den;
end
end

```

L'analogo bi-dimensionale dell'eq.8d.1 può essere spiegato attraverso il seguente esempio. Supponiamo che lo spostamento  $u$  debba essere interpolato a partire dai valori nodali  $u_1, u_2, \dots, u_9$  in Fig.8d.1.

L'applicazione dell'eq.8d.1 dà luogo alle seguenti equazioni:

$$\text{sulla linea 1-3 } u = u_{13} = N_1(x) \cdot u_1 + N_2(x) \cdot u_2 + N_3(x) \cdot u_3$$

$$\text{sulla linea 4-6 } u = u_{46} = N_1(x) \cdot u_4 + N_2(x) \cdot u_5 + N_3(x) \cdot u_6 \quad [8d.3]$$

$$\text{sulla linea 7-9 } u = u_{79} = N_1(x) \cdot u_7 + N_2(x) \cdot u_8 + N_3(x) \cdot u_9$$

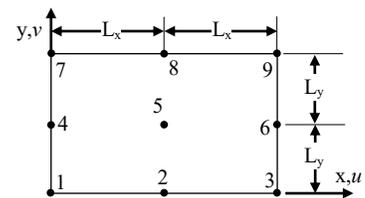


Fig.8d.1

Dopo di che, lungo una qualsiasi linea parallela all'asse  $y$ :

$$u = N_1(y) \cdot u_{13} + N_2(y) \cdot u_{46} + N_3(y) \cdot u_{79} \quad [8d.4]$$

dove sono state usate le seguenti funzioni  $N$  (che si possono ricavare dall'eq.8d.2), in cui la coordinata generica  $s$  può rappresentare sia la coordinata  $x$  che la coordinata  $y$ :



$$N_1(s) = \frac{\prod_{\substack{j=1 \\ j \neq 1}}^3 (s-s_j)}{\prod_{\substack{j=1 \\ j \neq 1}}^3 (s_1-s_j)} = \frac{(s-s_2) \cdot (s-s_3)}{(s_1-s_2) \cdot (s_1-s_3)} = \frac{(s-L_s) \cdot (s-2L_s)}{(-L_s) \cdot (-2L_s)} \quad [8d.5]$$

$$N_2(s) = \frac{\prod_{\substack{j=1 \\ j \neq 2}}^3 (s-s_j)}{\prod_{\substack{j=1 \\ j \neq 2}}^3 (s_2-s_j)} = \frac{(s-s_1) \cdot (s-s_3)}{(s_2-s_1) \cdot (s_2-s_3)} = \frac{(s) \cdot (s-2L_s)}{(L_s) \cdot (-L_s)} \quad [8d.5]$$

$$N_3(s) = \frac{\prod_{\substack{j=1 \\ j \neq 3}}^3 (s-s_j)}{\prod_{\substack{j=1 \\ j \neq 3}}^3 (s_3-s_j)} = \frac{(s) \cdot (s-L_s)}{(2L_s) \cdot (L_s)} \quad [8d.5]$$

La sostituzione delle eq.8d.3 nell'eq.8d.4 fornisce la formula cercata. Questa formula può essere generalizzata per prendere in considerazione  $m$  linee nodali parallele all'asse  $y$  e  $n$  linee nodali parallele all'asse  $x$ . In particolare:

$$u = \sum_{i=1}^m \sum_{j=1}^n N_i(x) \cdot N_j(y) \cdot u_{ij} \quad [8d.6]$$

in cui  $u_{ij}$  è il valore noto di  $u$  nelle coordinate  $x = x_i$  e  $y = y_j$ . Se necessario è possibile un'ulteriore generalizzazione al caso tridimensionale.

Lo spostamento in direzione  $y$  può essere definito sostituendo le  $u$  con le  $v$  nelle eq.8d.3 e 8d.4. E' quindi possibile definire il campo di spostamenti  $u = u(x, y)$  e  $v = v(x, y)$  in funzione degli spostamenti nodali, e generare un elemento finito che abbia i nodi che compaiono nella Fig.8d.1 utilizzando i metodi che discuteremo in seguito.

La formula di Lagrange è solo uno dei metodi d'interpolazione: se in corrispondenza dei nodi è necessario imporre oltre che gli spostamenti anche le pendenze della curva, è necessario ricorrere all'interpolazione di **Hermite**.

Se agli estremi dell'intervallo sono imposti due spostamenti e due rotazioni, si può ottenere l'interpolazione per mezzo di un polinomio cubico che per essere definito richiede quattro parametri:

$$u(x) = \{1 \quad x \quad x^2 \quad x^3\} \cdot \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{Bmatrix} = a_1 + a_2 \cdot x + a_3 \cdot x^2 + a_4 \cdot x^3$$

Poiché deve essere imposta anche la continuità delle rotazioni ai nodi, scriviamo:

$$\vartheta(x) = \frac{du(x)}{dx} = \{0 \quad 1 \quad 2x \quad 3x^2\} \cdot \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{Bmatrix}$$

Ponendo  $0 \leq x \leq L$ , imponiamo le condizioni al contorno e cioè:

$$\text{per } x=0 \quad u(x=0) = u_1 \quad ; \quad \vartheta(x=0) = \vartheta_1$$

$$\text{per } x=L \quad u(x=L) = u_2 \quad ; \quad \vartheta(x=L) = \vartheta_2$$

abbiamo:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & L & L^2 & L^3 \\ 0 & 1 & 2L & 3L^2 \end{bmatrix} \cdot \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{Bmatrix} = [A] \cdot \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{Bmatrix} = \begin{Bmatrix} u_1 \\ \vartheta_1 \\ u_2 \\ \vartheta_2 \end{Bmatrix}$$

Pre moltiplicando per l'inversa della matrice  $[A]$  possiamo ricavare il vettore dei parametri:



$$\begin{Bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{Bmatrix} = [A]^{-1} \cdot \begin{Bmatrix} u_1 \\ \vartheta_1 \\ u_2 \\ \vartheta_2 \end{Bmatrix}$$

$$[A]^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -3/L^2 & -2/L & 3/L^2 & -1/L \\ 2/L^3 & 1/L^2 & -2/L^3 & 1/L^2 \end{bmatrix}$$

da cui

$$u(x) = \{1 \quad x \quad x^2 \quad x^3\} \cdot \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{Bmatrix} = \{1 \quad x \quad x^2 \quad x^3\} \cdot [A]^{-1} \cdot \begin{Bmatrix} u_1 \\ \vartheta_1 \\ u_2 \\ \vartheta_2 \end{Bmatrix} = \{N_1 \quad N_2 \quad N_3 \quad N_4\} \cdot \begin{Bmatrix} u_1 \\ \vartheta_1 \\ u_2 \\ \vartheta_2 \end{Bmatrix}$$

Quindi le funzioni di forma o di miscelamento sono le seguenti:

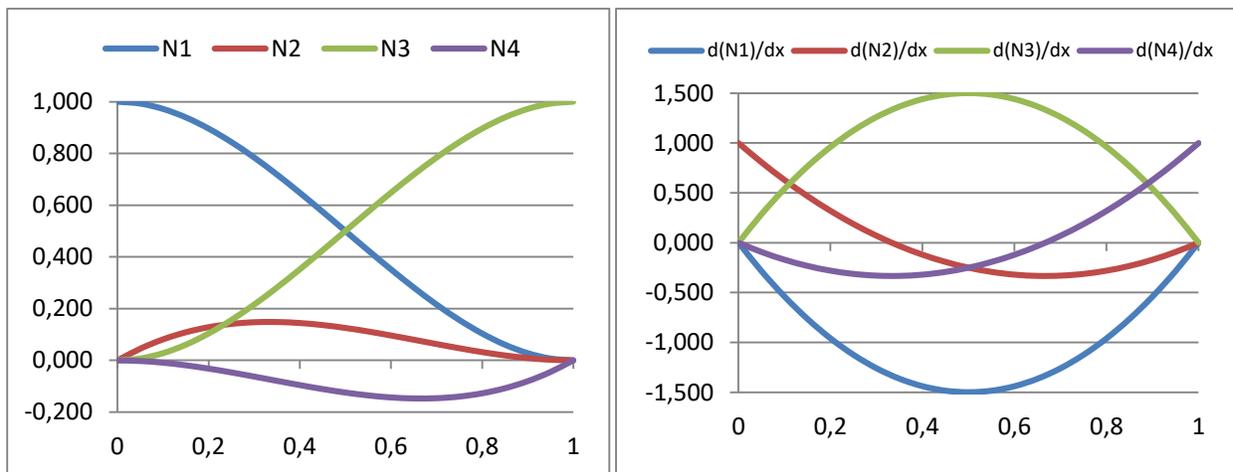
$$\{N_1 \quad N_2 \quad N_3 \quad N_4\} = \{1 \quad x \quad x^2 \quad x^3\} \cdot [A]^{-1} = \{1 \quad x \quad x^2 \quad x^3\} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -3/L^2 & -2/L & 3/L^2 & -1/L \\ 2/L^3 & 1/L^2 & -2/L^3 & 1/L^2 \end{bmatrix}$$

da cui:

$$\{N_1 \quad N_2 \quad N_3 \quad N_4\} = \left\{ 1 - \frac{3}{L^2}x^2 + \frac{2}{L^3}x^3; \quad x - \frac{2}{L}x^2 + \frac{1}{L^2}x^3; \quad \frac{3}{L^2}x^2 - \frac{2}{L^3}x^3; \quad -\frac{1}{L}x^2 + \frac{1}{L^2}x^3 \right\}$$

$$\vartheta(x) = \frac{du(x)}{dx} = \{0 \quad 1 \quad 2x \quad 3x^2\} \cdot \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{Bmatrix} = \left\{ \frac{dN_1}{dx} \quad \frac{dN_2}{dx} \quad \frac{dN_3}{dx} \quad \frac{dN_4}{dx} \right\} \cdot \begin{Bmatrix} u_1 \\ \vartheta_1 \\ u_2 \\ \vartheta_2 \end{Bmatrix}$$

$$\left\{ \frac{dN_1}{dx} \quad \frac{dN_2}{dx} \quad \frac{dN_3}{dx} \quad \frac{dN_4}{dx} \right\} = \left\{ -\frac{6}{L^2}x + \frac{6}{L^3}x^2; \quad 1 - \frac{4}{L}x + \frac{3}{L^2}x^2; \quad \frac{6}{L^2}x - \frac{6}{L^3}x^2; \quad -\frac{2}{L}x + \frac{3}{L^2}x^2 \right\}$$



Funzioni di miscelamento dei polinomi di Hermite e loro derivate



---

**Codice in MATLAB per l'interpolazione di una funzione con i polinomi di Hermite**

```
% =====  
% Interpolazione di Hermite  
% Ai nodi sono imposti ordinate e pendenze  
% =====  
disp('Interpolazione di Hermite');  
disp('Scegli la funzione da interpolare');  
disp('f = 0 Casuale');  
disp('f = 1 Sinusoidale');  
disp('f = 2 Esponenziale');  
disp('f = 3 Logaritmica');  
funzione = input('f = ');  
funzione = max(0,min(funzione,3));  
if funzione == 0  
    Np = 11;  
    xdati = 0:1:Np-1;  
    ydati = 100*(rand(Np) - 0.5);  
    theta = zeros(Np,1);  
    theta(1) = (ydati(2)-ydati(1))/(xdati(2)-xdati(1));  
    for i=2:Np-1  
        theta(i) = (ydati(i+1)-ydati(i-1))/(xdati(i+1)-xdati(i-1));  
    end  
    theta(Np) = (ydati(Np)-ydati(Np-1))/(xdati(Np)-xdati(Np-1));  
elseif funzione == 1  
    xdati = 0:pi/15:10*pi;  
    ydati = sin(xdati);  
    theta = cos(xdati);  
elseif funzione == 2  
    xdati = 0:0.1:2;  
    ydati = 100*exp(xdati);  
    theta = ydati;  
elseif funzione == 3  
    xdati = 1:1:20;  
    ydati = 100*log(xdati);  
    theta = zeros(20,1);  
    for i = 1:20  
        theta(i) = 100/xdati(i);  
    end  
end  
[i,Np] = size(xdati);  
  
Ndiv = 10;  
NpTot = (Np-1)*Ndiv+1;  
x = zeros(NpTot,1);  
y = zeros(NpTot,1);  
clf;  
title('Interpolazione di Hermite');  
hold on  
n = 0;  
for k = 1:Np-1  
    k1 = k + 1;  
    plot(xdati(k:k1),ydati(k:k1),'--rs', ...  
         'LineWidth',2,'MarkerEdgeColor','k',...  
         'MarkerFaceColor','g','MarkerSize',5);  
    Lenght = xdati(k1)-xdati(k);  
    passo = (xdati(k1)-xdati(k))/Ndiv;  
    xinc = xdati(k) - passo ;  
    yp = [ydati(k); theta(k); ydati(k1); theta(k1)];  
    xL =-passo;  
    for j = 1:Ndiv+1  
        xL = xL + passo;  
        N = Hermite (Lenght, xL);  
        n = n + 1;  
        xinc = xinc + passo;  
        x(n) = xinc;  
        y(n) = N*yp;  
    end  
end  
plot(x(1:n),y(1:n),'LineWidth',2);
```



```

input('... <Return> per continuare ...');
shg;
hold off
close;

function N = Hermite( Lenght, x)
%Hermite Calcola il valore delle funzioni di miscelamento alla coordinata x
L2 = Lenght^2;
L3 = L2*Lenght;
x2 = x^2;
x3 = x*x2;
N(1) = 1 - 3*x2/L2 + 2*x3/L3;
N(2) = x - 2*x2/Lenght + x3/L2;
N(3) = 3*x2/L2 - 2*x3/L3;
N(4) = -x2/Lenght + x3/L2;
end

```

### ESEMPIO DI INTERPOLAZIONE DI HERMITE A TRATTI

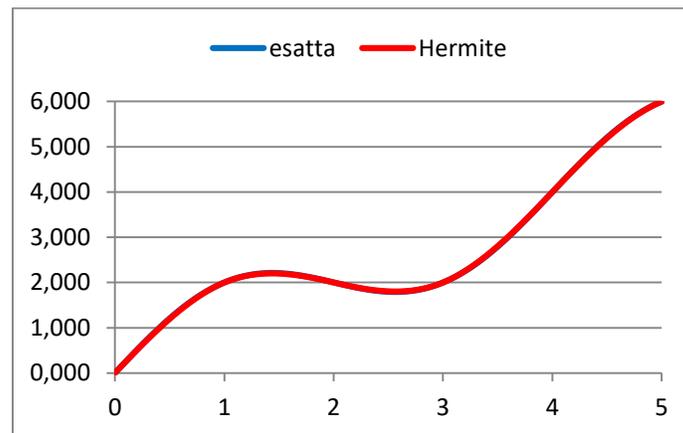
Immaginiamo di estrarre 6 punti a passo costante  $\Delta x = 1$  dalla seguente funzione:

$$y(x) = x + \sin\left(\frac{\pi}{2}x\right) \quad ; \quad \vartheta(x) = \frac{dy(x)}{dx} = 1 + \frac{\pi}{2} \cdot \cos\left(\frac{\pi}{2}x\right)$$

Usiamo l'interpolazione cubica di Hermite per tratti, imponendo in ogni nodo sia il valore della funzione che la sua pendenza. I dati da interpolare sono indicati nella tabella riportata qui a destra. Si può procedere all'interpolazione utilizzando una diversa cubica per ogni tratto compreso tra due punti consecutivi. Quindi tra due punti, lo spostamento  $v$  è determinato dai valori nodali agli estremi dell'intervallo e non dipende dai valori nodali esterni all'intervallo.

| i | x | y | $\theta$            |
|---|---|---|---------------------|
| 1 | 0 | 0 | $1 + \frac{\pi}{2}$ |
| 2 | 1 | 2 | 1                   |
| 3 | 2 | 2 | $1 - \frac{\pi}{2}$ |
| 4 | 3 | 2 | 1                   |
| 5 | 4 | 4 | $1 + \frac{\pi}{2}$ |
| 6 | 5 | 6 | 1                   |

Questa è una caratteristica del comportamento degli elementi finiti: gli spostamenti all'interno dell'elemento dipendono solo dagli spostamenti dei nodi appartenenti all'elemento stesso. In altre parole, il modello di una struttura agli elementi finiti comporta un'interpolazione polinomiale per tratti (*piecewise*); gli spostamenti nodali definiscono diversi campi di spostamento che giacciono su lati opposti adiacenti; non c'è un singolo polinomio che definisca gli spostamenti dell'intera struttura.



Nota che gli spostamenti e le pendenze sono continue sui nodi, ciò che non è vero per la curvatura, derivata seconda dello spostamento. Questo comportamento è tipico dell'interpolazione per parti.

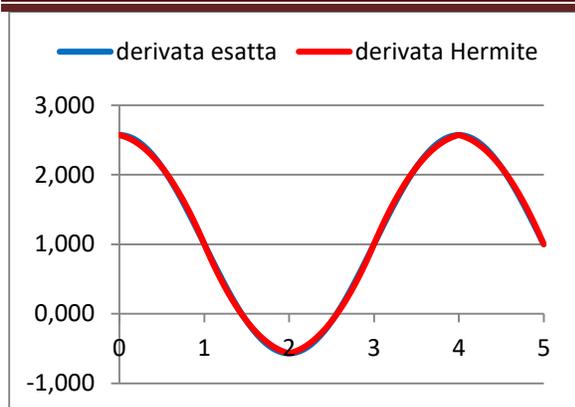


Fig.8d.2a

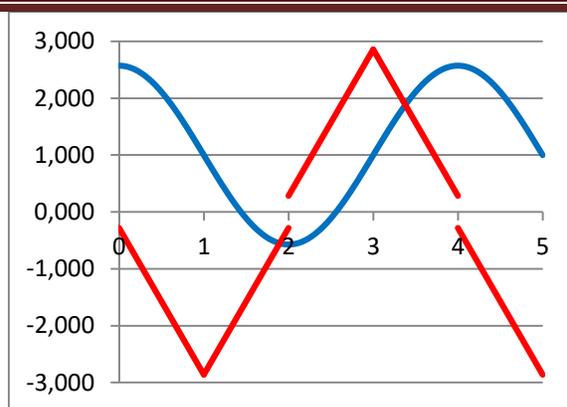


Fig.8d.2b

Fig.8d.2a - Confronto tra la derivata prima della funzione di partenza  $y(x) = x + \sin(\frac{\pi}{2}x)$  da cui sono stati estratti 5 punti (curva blu) e la derivata prima del polinomio di Hermite per tratti (curva rossa).

Fig.8d.2b - Confronto tra la derivata seconda della funzione di partenza  $y(x) = x + \sin(\frac{\pi}{2}x)$  da cui sono stati estratti 5 punti (curva blu) e derivata seconda del polinomio di Hermite per tratti (curva rossa).

Le **SERIE DI FOURIER** possono essere usate per rappresentare forze e spostamenti periodici. Una loro particolare applicazione è nell'analisi di solidi e gusci di rivoluzione sottoposti a carichi non assialsimmetrici, nel qual caso il periodo è pari a  $2\pi$  radianti.

Se  $f(x)$  è una funzione a valori reali, periodica, con periodo  $2\pi$ , è possibile rappresentarla per mezzo della serie di Fourier:

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left[ a_n \cdot \cos\left(n \frac{2\pi}{\lambda} x\right) + b_n \cdot \sin\left(n \frac{2\pi}{\lambda} x\right) \right]$$

dove:

$$a_0 = \frac{1}{\pi} \cdot \int_{-\pi}^{\pi} f(x) \cdot dx \quad ; \quad a_n = \frac{1}{\pi} \cdot \int_{-\pi}^{\pi} f(x) \cdot \cos(nx) \cdot dx \quad ; \quad b_n = \frac{1}{\pi} \cdot \int_{-\pi}^{\pi} f(x) \cdot \sin(nx) \cdot dx$$

I coefficienti  $a_n$  e  $b_n$  esprimono le ampiezze, ovvero i pesi delle sinusoidi e cosinusoidi,  $a_0/2$  corrisponde al valor medio in un periodo della funzione  $f(x)$  e  $\lambda$  rappresenta il periodo.

Le funzioni seno sono chiamate dispari o antisimmetriche, in quanto  $u(x) = -u(-x)$ ; le funzioni coseno sono dette pari o simmetriche perché  $u(x) = u(-x)$ .

Vediamo di rappresentare nel seguente esempio la funzione periodica:

$$\begin{aligned} \text{per } 0 \leq x \leq L & \quad q(x) = q_0 = \text{costante} \\ \text{per } L \leq x \leq 2L & \quad q(x) = -q_0 = \text{costante} \end{aligned}$$

Si tratta di un'onda quadra che si ripete con un periodo  $\lambda = 2L$  e che potrebbe essere utilizzata per rappresentare un carico distribuito uniformemente di intensità  $q_0$  su una trave di lunghezza pari a  $L$ . Si può dimostrare che lo sviluppo in serie di Fourier di tale funzione è la seguente:

$$q(x) = \frac{4q_0}{\pi} \sum_{n \text{ dispari}} \frac{1}{n} \cdot \sin\left(\frac{n\pi}{L} x\right)$$

Infatti abbiamo:



$$a_0 = \frac{1}{\pi} \cdot \int_{-\pi}^{\pi} f(x) \cdot dx = \frac{1}{\pi} \cdot \int_{-\pi}^0 (-q_0) \cdot dx + \frac{1}{\pi} \cdot \int_0^{\pi} q_0 \cdot dx = \frac{-q_0}{\pi} \cdot [x]_{-\pi}^0 + \frac{q_0}{\pi} \cdot [x]_0^{\pi} = -q_0 + q_0 = 0$$

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos(nx) dx = \frac{1}{\pi} \int_{-\pi}^0 (-q_0) \cdot \cos(nx) \cdot dx + \frac{1}{\pi} \int_0^{\pi} q_0 \cdot \cos(nx) \cdot dx$$

da cui:

$$a_n = \frac{-q_0}{n\pi} \cdot [\sin(nx)]_{-\pi}^0 + \frac{q_0}{n\pi} \cdot [\sin(nx)]_0^{\pi} = 0$$

$$b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin(nx) dx = \frac{1}{\pi} \int_{-\pi}^0 (-q_0) \cdot \sin(nx) \cdot dx + \frac{1}{\pi} \int_0^{\pi} q_0 \cdot \sin(nx) \cdot dx$$

da cui:

$$b_n = \frac{q_0}{n\pi} \cdot [\cos(nx)]_{-\pi}^0 - \frac{q_0}{n\pi} \cdot [\cos(nx)]_0^{\pi} = \frac{q_0}{n\pi} \cdot \{1 - \cos[n(-\pi)] - \cos[n\pi] + 1\}$$

Se  $n$  è dispari abbiamo  $\cos[n(-\pi)] = \cos[n\pi] = -1$  da cui:

$$b_n = \frac{4q_0}{n\pi}$$

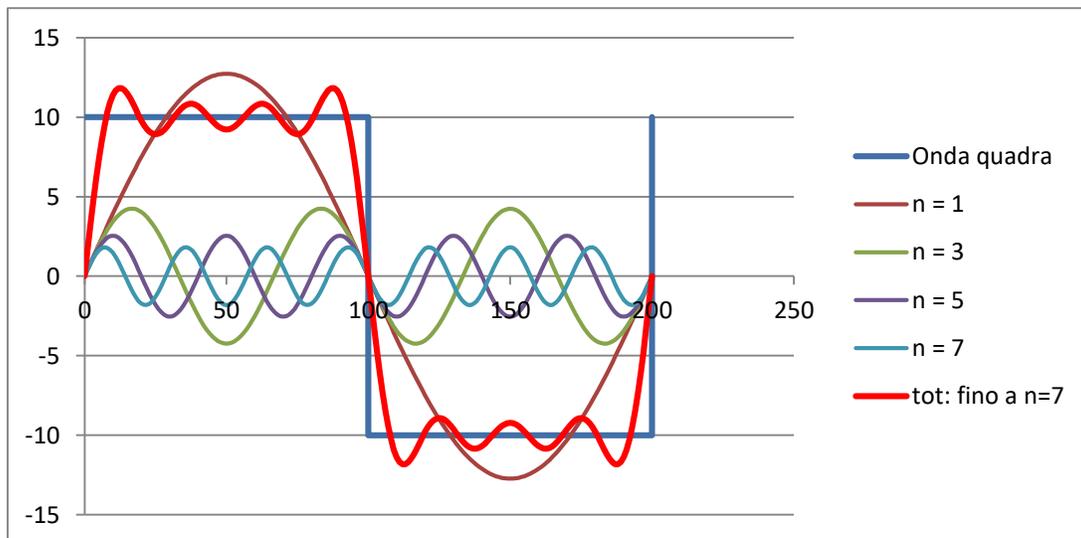
mentre se  $n$  è pari abbiamo  $\cos[n(-\pi)] = \cos[n\pi] = 0$  da cui:

$$b_n = 0$$

La funzione  $q = q(x)$  vale quindi:

$$q(x) = \sum_{n=1}^{\infty} b_n \cdot \sin\left(n \frac{2\pi}{\lambda} x\right) = \frac{4q_0}{\pi} \sum_{n \text{ dispari}} \frac{1}{n} \cdot \sin\left(\frac{n\pi}{L} x\right)$$

Come si può intuire osservando la seguente figura, aggiungendo un numero crescente di termini, la serie approssima sempre meglio la funzione esatta di partenza.





---

**Codice in MATLAB per l'analisi di una funzione (in questo esempio, l'onda quadra)**

```
% Genera la funzione da esaminare
q0 = 100.0;
nx = 1024;
half = fix(nx/2);
f(1:half) = q0;           % onda quadra
f(half+1:nx) = -q0; ;    % onda quadra
NFFT = 2^nextpow2(nx); % Next power of 2 from length of f

Y = fft(f,NFFT);        % FFT: Fast Fourier Transform

stampa = sprintf('Numero di termini per la ricostruzione (< %4d)',nx/2);
disp(stampa);
Ncoef = input('n = ');
Ncoef = max(1,min(Ncoef,nx/2));

p = zeros(nx,1);
euler = exp(2*pi*1i/nx);
for j = 1:nx
    omega = euler^(1-j);
    wrk = complex(0,0);
    for n = 1:Ncoef
        wrk = wrk + Y(n)*omega^(n-1);
    end
    p(j) = -2*real(wrk)/nx;
end

stampa = sprintf('n      f(x)_esatta   f(x)_Fourier      Err');
disp(stampa);
for i = 1:nx
    stampa = sprintf('%4d %14.7f %14.7f %14.7f',i,f(i),p(i),p(i) - f(i));
    disp(stampa);
end

figure;
hold on;
stampa = sprintf('Ricostruzione con le prime %4d frequenze',Ncoef);
title(stampa);
plot(f, 'b-');
plot(p, 'r-');
input('... <Return> per continuare ...');
close;
```