

### 13. Interpolazione trigonometrica e trasformata discreta di Fourier

Le funzioni trigonometriche  $\sin kx$  e  $\cos kx$ , con  $k$  intero, pur non essendo funzioni razionali, vengono ugualmente utilizzate per l'interpolazione in quanto facilmente calcolabili. Le loro proprietà di periodicità e di ortogonalità ne fanno la classe ideale per approssimare funzioni periodiche. Inoltre esse hanno in comune con la classe delle funzioni razionali la proprietà, utilissima per le applicazioni, che le loro derivate e le loro primitive sono ancora funzioni della stessa classe.

**5.55 Definizione.** Una funzione della forma

$$F(x) = \sum_{j=0}^m (\alpha_j \cos jx + \beta_j \sin jx) \quad (69)$$

è detta *polinomio trigonometrico* di grado  $m$ . ■

Data una funzione  $f(x)$ , reale e definita nell'intervallo  $[0, 2\pi)$ , il problema dell'*interpolazione trigonometrica* consiste nel determinare il polinomio trigonometrico  $F(x)$  di minimo grado tale che

$$F(x_k) = y_k, \quad \text{per } k = 0, \dots, n-1, \quad (70)$$

dove  $y_k = f(x_k)$  sono i valori assunti da  $f(x)$  negli  $n$  punti equidistanti  $x_k = 2k\pi/n$ ,  $k = 0, \dots, n-1$ .

Il problema dell'interpolazione trigonometrica è riconducibile a un problema di interpolazione polinomiale sul cerchio unitario del piano complesso. A questo scopo si introducono le radici  $n$ -esime dell'unità.

**5.56 Definizione.** Sia  $n$  un intero. Si definisce *radice  $n$ -esima dell'unità* ogni numero complesso  $z$  tale che  $z^n = 1$ . Una radice  $n$ -esima  $\omega$  è detta *primitiva* se l'insieme  $\{\omega^i, i = 0, \dots, n-1\}$  è costituito da  $n$  elementi distinti. In particolare, indicata con  $i$  l'unità immaginaria ( $i^2 = -1$ ), il numero complesso

$$\omega_n = e^{i2\pi/n}$$

è una radice primitiva  $n$ -esima dell'unità. ■

Poiché

$$\omega_n^r = e^{i2\pi r/n},$$

le successive potenze di  $\omega_n$  sono nell'ordine tutte e sole le radici  $n$ -esime dell'unità, e

$$\omega_n^k = \omega_n^p \quad \text{se e solo se } k \equiv p \pmod{n}.$$

**5.57 Teorema.** Vale la relazione di ortogonalità:

$$\sum_{j=0}^{n-1} \omega_n^{jk} = \begin{cases} n & \text{se } k \equiv 0 \pmod{n}, \\ 0 & \text{altrimenti.} \end{cases}$$

**Dim.** Se  $k \equiv 0 \pmod{n}$ , esiste un intero  $s$  per cui  $k = sn$ , e quindi è  $\omega_n^{jk} = 1$ . Altrimenti, se  $k \not\equiv 0 \pmod{n}$ , è  $\omega_n^k \neq 1$ ; ponendo  $x = \omega_n^k$  e utilizzando la nota relazione

$$(1 + x + x^2 + \dots + x^{n-1})(1 - x) = 1 - x^n,$$

si ottiene

$$\left(\sum_{j=0}^{n-1} \omega_n^{jk}\right)(1 - \omega_n^k) = 1 - \omega_n^{nk},$$

e, ricordando che  $\omega_n^k \neq 1$  e che  $\omega_n^{nk} = 1$ , ne segue la tesi. ■

Si considera ora un problema di interpolazione polinomiale nel campo complesso che, come si vedrà nel teorema 5.59, è equivalente al problema dell'interpolazione trigonometrica: si tratta di determinare i numeri complessi  $z_j$ ,  $j = 0, \dots, n-1$ , coefficienti del polinomio di grado al più  $n-1$

$$p(w) = \sum_{j=0}^{n-1} z_j w^j, \quad (71)$$

tale che

$$p(\omega_n^k) = y_k, \quad \text{per } k = 0, \dots, n-1. \quad (72)$$

I coefficienti  $z_j$  si ottengono risolvendo il sistema (3)

$$V\mathbf{z} = \mathbf{y},$$

dove  $\mathbf{y} = [y_0, y_1, \dots, y_{n-1}]^T$ ,  $\mathbf{z} = [z_0, z_1, \dots, z_{n-1}]^T$  e  $V$  è la matrice di Vandermonde di ordine  $n$ , i cui elementi sono

$$v_{kj} = \omega_n^{kj}, \quad k, j = 0, \dots, n-1.$$

Per il teorema 5.57 la matrice  $V$  soddisfa alla proprietà (si veda l'esercizio 5.54)  $V^H V = nI$  (dove con  $V^H$  si è indicata la matrice trasposta della matrice i cui elementi sono i coniugati degli elementi di  $V$ ). Ne segue che  $V^{-1} = \frac{1}{n} V^H$ , e quindi

$$\mathbf{z} = \frac{1}{n} V^H \mathbf{y}. \quad (73)$$

**5.58 Definizione.** L'applicazione che al vettore  $\mathbf{y}$  associa il vettore  $\mathbf{z}$  definito in (73) è detta *trasformata discreta di Fourier* e viene generalmente indicata con la sigla DFT, mentre il vettore  $\mathbf{z} = \text{DFT}(\mathbf{y})$  è detto *trasformata discreta di Fourier* del vettore  $\mathbf{y}$  e verifica la relazione

$$z_j = \frac{1}{n} \sum_{k=0}^{n-1} y_k \omega_n^{-jk}, \quad j = 0, \dots, n-1. \quad (74)$$

L'applicazione che al vettore  $\mathbf{z}$  associa il vettore  $\mathbf{y}$  è detta *trasformata discreta inversa di Fourier* e viene generalmente indicata con la sigla IDFT, mentre il vettore  $\mathbf{y} = \text{IDFT}(\mathbf{z})$  è detto *trasformata discreta inversa di Fourier* del vettore  $\mathbf{z}$  e verifica la relazione

$$y_k = \sum_{j=0}^{n-1} z_j \omega_n^{jk}, \quad k = 0, \dots, n-1. \quad (75)$$

Se il vettore  $\mathbf{y}$  ha componenti reali, le componenti del vettore  $\mathbf{z}$  sono tali che

$$z_0 \text{ è reale e } \bar{z}_j = z_{n-j}, \quad \text{per } j = 1, \dots, n-1. \quad (76)$$

Infatti dalla (74), essendo  $\omega_n^{-nk} = 1$  per  $k$  intero, si ha che

$$z_0 = \frac{1}{n} \sum_{k=0}^{n-1} y_k \quad \text{e} \quad z_{n-j} = \frac{1}{n} \sum_{k=0}^{n-1} y_k \omega_n^{-(n-j)k} = \frac{1}{n} \sum_{k=0}^{n-1} y_k \omega_n^{jk} = \bar{z}_j.$$

Se  $n$  è pari,  $n = 2m$ , anche  $z_m$  è reale, in quanto  $\bar{z}_m = z_m$ .

**5.59 Teorema.** Il polinomio trigonometrico di interpolazione della funzione  $f(x)$  negli  $n$  punti  $x_k$ ,  $k = 0, \dots, n-1$ , è

$$F_n(x) = \begin{cases} \frac{\alpha_0}{2} + \sum_{j=1}^{m-1} (\alpha_j \cos jx + \beta_j \sin jx), & \text{se } n = 2m-1, \\ \frac{\alpha_0}{2} + \sum_{j=1}^{m-1} (\alpha_j \cos jx + \beta_j \sin jx) + \frac{\alpha_m}{2} \cos mx, & \text{se } n = 2m, \end{cases} \quad (77)$$

in cui i coefficienti  $\alpha_j, \beta_j$  sono reali e sono dati da:

$$\begin{aligned} \alpha_j &= 2 \operatorname{Re}(z_j) = \frac{2}{n} \sum_{k=0}^{n-1} f(x_k) \cos jx_k, \\ \beta_j &= -2 \operatorname{Im}(z_j) = \frac{2}{n} \sum_{k=0}^{n-1} f(x_k) \sin jx_k, \end{aligned} \quad (78)$$

420 Capitolo 5. Interpolazione

dove  $\mathbf{z}$  è la DFT del vettore  $\mathbf{y} = [f(x_0), f(x_1), \dots, f(x_{n-1})]^T$ ; in particolare se  $n = 2m$ , è

$$\alpha_m = 2z_m = \frac{2}{n} \sum_{k=0}^{n-1} (-1)^k f(x_k);$$

e tale polinomio è unico.

**Dim.** Si esamina dapprima il caso  $n = 2m - 1$ . In questo caso la (75) per la (76) si può scrivere

$$\begin{aligned} y_k &= \sum_{j=0}^{2m-2} z_j \omega_n^{jk} = \sum_{j=0}^{m-1} z_j \omega_n^{jk} + \sum_{j=m}^{2m-2} z_j \omega_n^{jk} \\ &= \sum_{j=0}^{m-1} z_j \omega_n^{jk} + \sum_{j=m}^{2m-2} \bar{z}_{n-j} \omega_n^{jk} = \sum_{j=0}^{m-1} z_j \omega_n^{jk} + \sum_{p=1}^{m-1} \bar{z}_p \omega_n^{(n-p)k} \\ &= z_0 + \sum_{j=1}^{m-1} (z_j \omega_n^{jk} + \bar{z}_j \omega_n^{-jk}). \end{aligned}$$

Poiché

$$\omega_n^{jk} = e^{\mathbf{i}jx_k} = \cos jx_k + \mathbf{i} \sin jx_k,$$

risulta

$$y_k = z_0 + \sum_{j=1}^{m-1} [(z_j + \bar{z}_j) \cos jx_k + \mathbf{i}(z_j - \bar{z}_j) \sin jx_k],$$

e ponendo

$$\alpha_j = z_j + \bar{z}_j, \quad \beta_j = \mathbf{i}(z_j - \bar{z}_j), \quad j = 0, \dots, m-1,$$

si ha

$$y_k = \frac{\alpha_0}{2} + \sum_{j=1}^{m-1} (\alpha_j \cos jx_k + \beta_j \sin jx_k).$$

Ne segue che il polinomio trigonometrico della forma (69)

$$F_n(x) = \frac{\alpha_0}{2} + \sum_{j=1}^{m-1} (\alpha_j \cos jx + \beta_j \sin jx)$$

soddisfa alle (70). I coefficienti  $\alpha_j$  e  $\beta_j$ , calcolati per mezzo delle (74), sono

$$\begin{aligned}\alpha_j &= z_j + \bar{z}_j = \frac{1}{n} \sum_{k=0}^{n-1} y_k (\omega_n^{-jk} + \omega_n^{jk}) = \frac{1}{n} \sum_{k=0}^{n-1} y_k (e^{-\mathbf{i}jx_k} + e^{\mathbf{i}jx_k}) \\ &= \frac{2}{n} \sum_{k=0}^{n-1} y_k \cos jx_k, \quad \text{per } j = 0, \dots, n-1, \\ \beta_0 &= 0, \quad \beta_j = \mathbf{i}(z_j - \bar{z}_j) = \frac{2}{n} \sum_{k=0}^{n-1} y_k \sin jx_k \quad \text{per } j = 1, \dots, n-1.\end{aligned}$$

Nel caso in cui  $n = 2m$ , procedendo in modo analogo, dalla (75) si ha

$$y_k = \sum_{j=0}^{2m-1} z_j \omega_n^{jk} = z_0 + \sum_{j=1}^{m-1} (z_j \omega_n^{jk} + \bar{z}_j \omega_n^{-jk}) + z_m \omega_n^{mk}.$$

Poiché  $\omega_n^{mk} = \cos mx_k$ , il polinomio trigonometrico cercato è della forma

$$F_n(x) = \frac{\alpha_0}{2} + \sum_{j=1}^{m-1} (\alpha_j \cos jx + \beta_j \sin jx) + \frac{\alpha_m}{2} \cos mx.$$

L'unicità del polinomio di interpolazione trigonometrico segue dall'unicità del polinomio (71) che verifica le (72) e dal fatto che la relazione che lega i coefficienti  $\alpha_j$  e  $\beta_j$  con gli  $z_j$  è biunivoca. ■

**5.60 Esempio.** Sia  $f(x)$  la funzione continua a tratti ottenuta per estensione periodica di periodo  $2\pi$  della funzione  $f(x) = x$ , per  $0 \leq x < 2\pi$ . Assumendo  $n = 3$ , si ha

$x$	0	$\frac{2\pi}{3}$	$\frac{4\pi}{3}$
$f(x)$	0	$\frac{2\pi}{3}$	$\frac{4\pi}{3}$

Il polinomio trigonometrico di interpolazione è dato da

$$F_3(x) = \frac{\alpha_0}{2} + \alpha_1 \cos x + \beta_1 \sin x$$

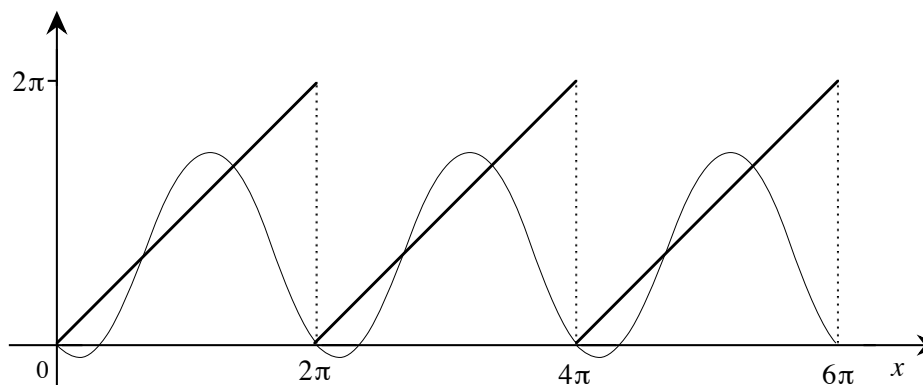
dove

$$\begin{aligned}\alpha_0 &= \frac{2}{3} \left( \frac{2\pi}{3} + \frac{4\pi}{3} \right) = \frac{4\pi}{3}, \\ \alpha_1 &= \frac{2}{3} \left( \frac{2\pi}{3} \cos \frac{2\pi}{3} + \frac{4\pi}{3} \cos \frac{4\pi}{3} \right) = -\frac{2\pi}{3}, \\ \beta_1 &= \frac{2}{3} \left( \frac{2\pi}{3} \sin \frac{2\pi}{3} + \frac{4\pi}{3} \sin \frac{4\pi}{3} \right) = -\frac{2\pi\sqrt{3}}{9},\end{aligned}$$

per cui

$$F_3(x) = \frac{2\pi}{3} \left( 1 - \cos x - \frac{\sqrt{3}}{3} \sin x \right).$$

Nella figura 5.17 sono riportati il grafico su 3 periodi della funzione  $f(x)$  (linea spessa) e il grafico del polinomio trigonometrico di interpolazione  $F_3(x)$  (linea sottile).



**Fig. 5.17** - Grafico del polinomio trigonometrico di interpolazione di  $f(x) = x$ ,  $0 \leq x < 2\pi$ , per  $n = 3$ .

Nella figura 5.18 è riportato il grafico del polinomio trigonometrico ottenuto con  $n = 8$ , i cui coefficienti sono

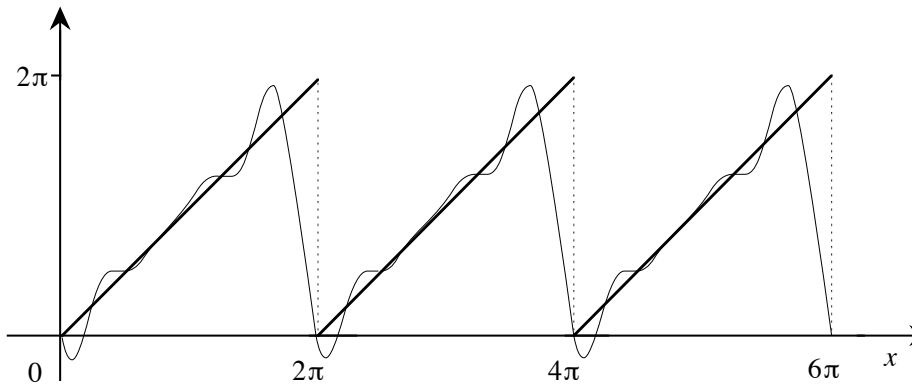
$$\begin{aligned} \alpha_0 &= \frac{7\pi}{4}, & \alpha_1 &= \alpha_2 = \alpha_3 = \alpha_4 = -\frac{\pi}{4}, \\ \beta_1 &= -\frac{\pi}{4}(1 + \sqrt{2}), & \beta_2 &= -\frac{\pi}{4}, & \beta_3 &= \frac{\pi}{4}(1 - \sqrt{2}). \quad \blacksquare \end{aligned}$$

Se nell'intervallo  $[0, 2\pi)$  la funzione  $f(x)$  è simmetrica, oppure antisimmetrica, rispetto al punto  $\pi$  i coefficienti  $\alpha_j$  e  $\beta_j$  del polinomio trigonometrico di interpolazione si semplificano nel modo seguente:

a) se  $f(x)$  è simmetrica, allora

$$\alpha_j = \begin{cases} \frac{2}{n} \left[ f(x_0) + 2 \sum_{k=1}^{m-1} f(x_k) \cos jx_k \right], & \text{se } n = 2m - 1, \\ \frac{2}{n} \left[ f(x_0) + (-1)^j f(x_m) + 2 \sum_{k=1}^{m-1} f(x_k) \cos jx_k \right], & \text{se } n = 2m, \end{cases}$$

$$\beta_j = 0,$$



**Fig. 5.18** - Grafico del polinomio trigonometrico di interpolazione di  $f(x) = x$ ,  $0 \leq x < 2\pi$ , per  $n = 8$ .

b) se  $f(x)$  è antisimmetrica (e quindi è  $f(\pi) = 0$ ), allora

$$\alpha_j = \frac{2}{n} f(x_0),$$

$$\beta_j = \frac{4}{n} \sum_{k=1}^{m-1} f(x_k) \sin jx_k.$$

La dimostrazione delle relazioni precedenti tiene conto del fatto che, se  $f(x)$  è simmetrica, allora  $f(x_k) = f(x_{n-k})$  e se  $f(x)$  è antisimmetrica, allora  $f(x_k) = -f(x_{n-k})$  ed inoltre del fatto che

$$\cos jx_k = \cos jx_{n-k} \quad \text{e} \quad \sin jx_k = -\sin jx_{n-k}.$$

Per esempio se la funzione  $f(x)$  è simmetrica e  $n = 2m - 1$ , allora

$$\begin{aligned} \beta_j &= \frac{2}{n} \left[ \sum_{k=1}^{m-1} f(x_k) \sin jx_k + \sum_{k=m}^{2m-2} f(x_k) \sin jx_k \right] \\ &= \frac{2}{n} \left[ \sum_{k=1}^{m-1} f(x_k) \sin jx_k + \sum_{k=1}^{m-1} f(x_{n-k}) \sin jx_{n-k} \right] = 0. \end{aligned}$$

Le altre relazioni si dimostrano in modo analogo.

**5.61 Esempio.** Sia  $f(x)$  la funzione ottenuta per estensione periodica di periodo  $2\pi$  della funzione  $f(x) = x(2\pi - x)$ , per  $0 \leq x < 2\pi$ . Assumendo  $n = 3$ , si ha

$x$	$0$	$\frac{2\pi}{3}$	$\frac{4\pi}{3}$
$f(x)$	$0$	$\frac{8\pi^2}{9}$	$\frac{8\pi^2}{9}$

Poiché la funzione  $f(x)$  è simmetrica rispetto al punto  $\pi$  il polinomio trigonometrico di interpolazione è dato da

$$F_3(x) = \frac{\alpha_0}{2} + \alpha_1 \cos x$$

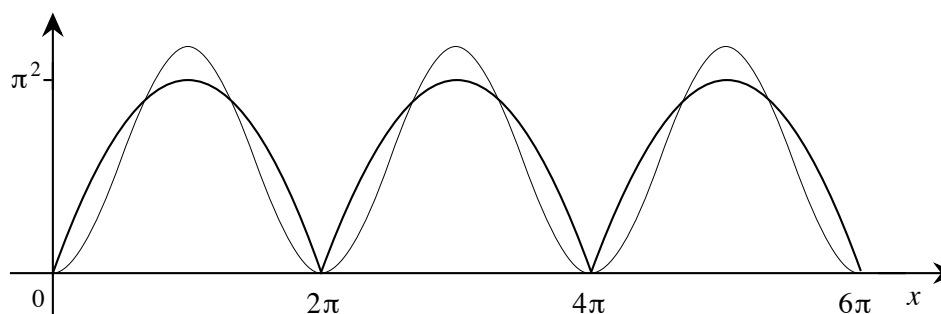
dove

$$\alpha_0 = 2 \frac{2}{3} \frac{8\pi^2}{9} = \frac{32\pi^2}{27}, \quad \alpha_1 = 2 \frac{2}{3} \frac{8\pi^2}{9} \cos \frac{2\pi}{3} = -\frac{16\pi^2}{27},$$

per cui

$$F_3(x) = \frac{16\pi^2}{27}(1 - \cos x).$$

Nella figura 5.19 sono riportati il grafico su 3 periodi della funzione  $f(x)$  (linea spessa) e il grafico del polinomio trigonometrico di interpolazione  $F_3(x)$  (linea sottile).



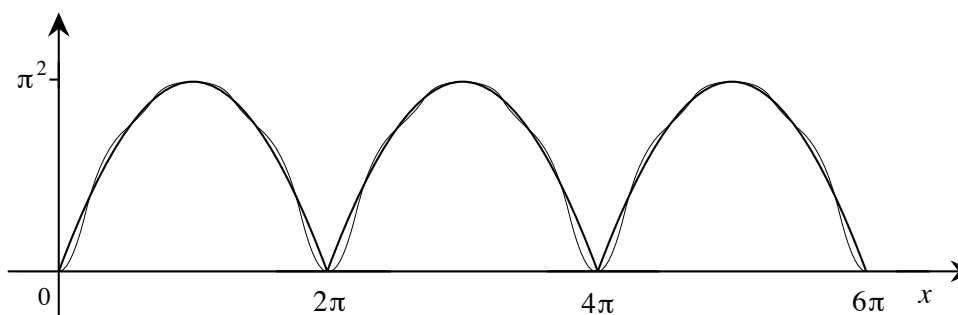
**Fig. 5.19** - Grafico del polinomio trigonometrico di interpolazione di  $f(x) = x(2\pi - x)$ ,  $0 \leq x < 2\pi$ , per  $n = 3$ .

Nella figura 5.20 è riportato il grafico del polinomio trigonometrico ottenuto con  $n = 8$ , i cui coefficienti sono

$$\alpha_0 = \frac{21\pi^2}{16}, \quad \alpha_1 = -\frac{\pi^2}{8}(2 + \sqrt{2}), \quad \alpha_2 = -\frac{\pi^2}{8}, \quad \alpha_3 = -\frac{\pi^2}{8}(2 - \sqrt{2}),$$

$$\alpha_4 = -\frac{\pi^2}{16}. \quad \blacksquare$$





**Fig. 5.20** - Grafico del polinomio trigonometrico di interpolazione di  $f(x) = x(2\pi - x)$ ,  $0 \leq x < 2\pi$ , per  $n = 8$ .

Se la funzione  $f(x)$  è definita sull'intervallo  $[0, \pi]$ , è sempre possibile estenderla all'intervallo  $[0, 2\pi)$  in modo da ottenere una funzione simmetrica rispetto al punto  $\pi$  e un corrispondente polinomio trigonometrico di interpolazione di soli coseni. Se  $y_k = f(x_k)$  per  $x_k = \frac{k\pi}{n+1}$ ,  $k = 0, \dots, n+1$ , si estende la funzione all'intervallo  $[0, 2\pi)$  definendo i valori  $f(x_k) = y_{2n+2-k}$  per  $k = n+2, \dots, 2n+1$ . Dalla (77) risulta che il polinomio di interpolazione è

$$F_{2n+2}(x) = \frac{\alpha_0}{2} + \sum_{j=1}^n \alpha_j \cos jx + \frac{\alpha_{n+1}}{2} \cos(n+1)x, \quad (79)$$

con

$$\alpha_j = \frac{1}{n+1} \left[ y_0 + 2 \sum_{k=1}^n y_k \cos jx_k + (-1)^j y_{n+1} \right], \quad j = 0, \dots, n+1. \quad (80)$$

Un'importante proprietà di cui gode l'interpolazione trigonometrica è quella della convergenza, per  $n$  che tende all'infinito, del polinomio  $F_n(x)$  alla funzione  $f(x)$ , se sono soddisfatte certe ipotesi di regolarità. Vale infatti il seguente teorema, per la cui dimostrazione si veda l'esercizio 5.64.

**5.62 Teorema.** *Sia  $f(x)$  una funzione periodica di periodo  $2\pi$  e derivabile due volte con continuità. Allora per ogni  $\epsilon > 0$  esiste un polinomio trigonometrico di interpolazione  $F_n(x)$  tale che*

$$|f(x) - F_n(x)| \leq \epsilon \quad \text{per ogni } x. \quad \blacksquare$$

Il calcolo del valore  $F_n(x)$  in un punto  $x$  di un polinomio trigonometrico, di cui siano noti i coefficienti  $\alpha_j$  e  $\beta_j$ , sembra a prima vista assai più complesso di quello di una funzione razionale. In realtà un'espressione

della forma (77) può essere calcolata con una sola valutazione di funzione trigonometrica. Poiché valgono le seguenti relazioni:

$$\begin{aligned}\cos^2 x &= 1 - \sin^2 x, \\ \sin jx &= \sin x \cos(j-1)x + \cos x \sin(j-1)x, \\ \cos jx &= \cos x \cos(j-1)x - \sin x \sin(j-1)x,\end{aligned}$$

nel caso di  $n$  dispari,  $n = 2m - 1$ , il valore di  $F_n(x)$  può essere calcolato con il seguente algoritmo, in cui il segno di  $c_1$  viene scelto in base al valore di  $x$ :

$$\left. \begin{aligned}s_1 &= \sin x, & c_1 &= \pm\sqrt{1 - s_1^2}, & F_1 &= \frac{\alpha_0}{2} + \alpha_1 c_1 + \beta_1 s_1, \\ s_j &= s_1 c_{j-1} + c_1 s_{j-1}, \\ c_j &= c_1 c_{j-1} - s_1 s_{j-1}, \\ F_j &= F_{j-1} + \alpha_j c_j + \beta_j s_j,\end{aligned} \right\} \text{ per } j = 2, 3, \dots, m-1,$$

$$F_n(x) = F_{m-1},$$

il cui costo computazionale, a meno di costanti additive, è di  $4m$  addizioni,  $6m$  moltiplicazioni, più una estrazione di radice quadrata e una valutazione di  $\sin x$ . Il calcolo di queste funzioni trascendenti può essere trascurato nel conto totale delle operazioni, perché esse richiedono un numero di operazioni aritmetiche indipendente da  $m$ . Nel caso di  $n$  pari, l'algoritmo è assai simile e ha lo stesso costo computazionale. Infine nel caso che il polinomio di interpolazione sia di soli coseni, il costo computazionale può essere ulteriormente ridotto (si veda l'esercizio 5.61).

I coefficienti  $\alpha_j$  e  $\beta_j$ , dati dalle (78), nella pratica sono calcolati per mezzo della DFT del vettore  $\mathbf{y}$ , il cui calcolo, per la particolare struttura della matrice  $V$ , richiede meno delle  $n^2$  operazioni sufficienti per la moltiplicazione di una matrice di ordine  $n$  per un vettore. Per il calcolo della IDFT e della DFT esistono infatti algoritmi molto efficienti, detti *algoritmi FFT* (*Fast Fourier Transform*), che hanno un costo computazionale dell'ordine di  $n \log_2 n$ . Per semplicità si esamina il caso in cui  $n$  è potenza di 2.

**5.63 Teorema.** *Sia  $n = 2^s$ ; il costo computazionale del calcolo della IDFT di un vettore  $\mathbf{z}$  di ordine  $n$  o del calcolo della DFT di un vettore  $\mathbf{y}$  di ordine  $n$ , a meno di termini di ordine inferiore, è di  $(n/2) \log_2 n$  moltiplicazioni fra numeri complessi e  $n \log_2 n$  addizioni fra numeri complessi, non contando il calcolo delle  $n$  radici  $n$ -esime dell'unità.*

**Dim.** Posto  $n = 2m$ , per la IDFT( $\mathbf{z}$ ) si ha dalla (75)

$$\begin{aligned} y_k &= \sum_{j=0}^{n-1} z_j \omega_n^{jk} = \sum_{j \text{ pari}} z_j \omega_n^{jk} + \sum_{j \text{ dispari}} z_j \omega_n^{jk} \\ &= \sum_{p=0}^{m-1} z_{2p} \omega_n^{2kp} + \sum_{p=0}^{m-1} z_{2p+1} \omega_n^{k(2p+1)}. \end{aligned}$$

Ponendo  $z'_p = z_{2p}$  e  $z''_p = z_{2p+1}$ ,  $p = 0, 1, \dots, m-1$ , si ha

$$y_k = \sum_{p=0}^{m-1} z'_p \omega_n^{2kp} + \sum_{p=0}^{m-1} z''_p \omega_n^{k(2p+1)}.$$

Tenendo presente che  $\omega_n^{2p} = (\omega_{n/2})^p = \omega_m^p$ , è

$$y_k = \sum_{p=0}^{m-1} z'_p \omega_m^{kp} + \omega_n^k \sum_{p=0}^{m-1} z''_p \omega_m^{kp}, \quad k = 0, \dots, n-1. \quad (81)$$

Posto  $\mathbf{y}' = \text{IDFT}(\mathbf{z}')$  e  $\mathbf{y}'' = \text{IDFT}(\mathbf{z}'')$ , cioè

$$y'_q = \sum_{p=0}^{m-1} z'_p \omega_m^{qp}, \quad y''_q = \sum_{p=0}^{m-1} z''_p \omega_m^{qp}, \quad q = 0, \dots, m-1,$$

dalla (81) segue che i primi  $m$  elementi della trasformata sono dati da

$$y_q = y'_q + \omega_n^q y''_q, \quad q = 0, \dots, m-1. \quad (82)$$

Per calcolare i rimanenti  $m$  elementi della trasformata, poiché  $\omega_n^m = -1$  e  $\omega_m^{q+m} = \omega_m^q$ , dalla (81) segue

$$\begin{aligned} y_{q+m} &= \sum_{p=0}^{m-1} z'_p \omega_m^{(q+m)p} + \omega_n^{q+m} \sum_{p=0}^{m-1} z''_p \omega_m^{(q+m)p} \\ &= \sum_{p=0}^{m-1} z'_p \omega_m^{qp} + \omega_n^{q+m} \sum_{p=0}^{m-1} z''_p \omega_m^{qp} \\ &= y'_q - \omega_n^q y''_q, \quad q = 0, \dots, m-1. \end{aligned} \quad (83)$$

La trasformata di ordine  $n$  può quindi essere effettuata con 2 trasformate di ordine  $n/2$  più  $n/2$  moltiplicazioni e  $n$  addizioni. Poiché la trasformata di ordine 1 non richiede operazioni, si possono scrivere le seguenti relazioni

di ricorrenza per il numero di addizioni  $A(n)$  e di moltiplicazioni  $M(n)$  occorrenti per il calcolo della trasformata di ordine  $n$

$$A(1) = 0, \quad A(n) = 2A(n/2) + n, \quad (84)$$

$$M(1) = 0, \quad M(n) = 2M(n/2) + n/2.$$

Posto  $t_s = A(n)$ , dove  $s = \log_2 n$ , dalla (84) si ottiene l'equazione alle differenze

$$t_0 = 0, \quad t_s = 2t_{s-1} + 2^s,$$

la cui soluzione è data da

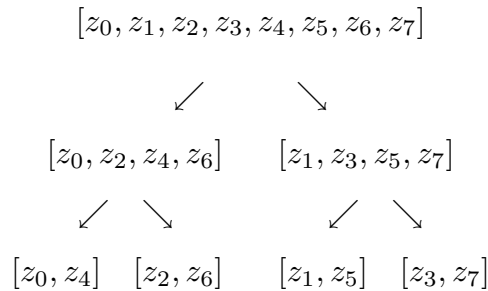
$$t_s = s2^s,$$

da cui  $A(n) = n \log_2 n$ . Analogamente si ottiene  $M(n) = n/2 \log_2 n$ . Si procede nello stesso modo per la DFT( $\mathbf{y}$ ), eseguendo solo al termine le divisioni per  $n$ . ■

Delle  $m$  radici  $n$ -esime dell'unità presenti nelle (82) e (83), basta calcolare le prime  $m/4$ , in quanto le altre si ottengono per simmetria.

L'algoritmo per il calcolo della IDFT( $\mathbf{z}$ ) che si ricava dalla dimostrazione del teorema 5.63 non è direttamente implementabile con un linguaggio che non ammetta la ricorsione, e comunque richiede un notevole ingombro di memoria a causa del procedimento ricorsivo. È però possibile costruire algoritmi non ricorsivi che sfruttano ancora le (82) e (83). Fra questi il più noto (per  $n$  potenza di 2) è quello di *Cooley e Tukey*, che verrà ora esposto.

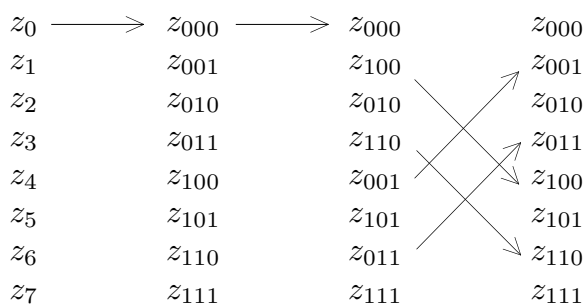
Dalla dimostrazione del teorema 5.63 risulta che la IDFT di ordine  $n$  viene calcolata mediante due IDFT di ordine  $n/2$ , che a loro volta vengono calcolate mediante IDFT di ordine  $n/4$ , e così via fino alle trasformate di ordine 2. Percorrendo il cammino inverso, si parte dalle IDFT di ordine 2 e si calcolano successivamente le IDFT di ordine  $4, \dots, n/2, n$ . La difficoltà del procedimento consiste nel determinare il giusto ordinamento delle componenti del vettore  $\mathbf{z}$ , a cui applicare le IDFT di ordine più basso. Ad esempio, nel caso  $n = 8$  i vettori che vengono considerati successivamente possono essere rappresentati secondo lo schema



Quindi per applicare correttamente il procedimento a partire dalle trasformate di ordine 2, è necessario riordinare il vettore  $\mathbf{z}$  nel vettore

$$\widehat{\mathbf{z}} = [z_0, z_4, z_2, z_6, z_1, z_5, z_3, z_7],$$

in modo da considerare ad ogni passo sottosequenze formate da elementi con indice pari o sottosequenze formate da elementi con indice dispari. Si può facilmente verificare che il riordinamento richiesto può essere realizzato mediante la seguente procedura, indicata come *bit reversal*: gli indici  $j$  delle componenti  $z_j$  vengono rappresentati con notazione in base 2, utilizzando  $\log_2 n$  bit, poi questi indici vengono scritti in ordine inverso e il vettore  $\mathbf{z}$  riordinato secondo questi nuovi indici. Nel caso visto precedentemente di  $n = 8$  risulta



Vengono così scambiati di posto fra loro  $z_1$  e  $z_4$ ,  $z_3$  e  $z_6$ , ottenendo il vettore  $\widehat{\mathbf{z}}$  (nell'implementazione gli scambi non vengono fatti sulle componenti del vettore  $\mathbf{z}$ , ma sulle componenti di un vettore ausiliario di indici). Riapplicando la procedura bit reversal al vettore  $\widehat{\mathbf{z}}$  si riottiene il vettore  $\mathbf{z}$ . In altri termini la permutazione di indici descritta è *involutoria*.

Il passo successivo consiste nell'applicazione delle (82) e (83). Le radici  $n$ -esime dell'unità possono essere calcolate tutte all'inizio, a partire da  $\omega_n = \cos \frac{2\pi}{n} + i \sin \frac{2\pi}{n}$ , sfruttando le formule trigonometriche di addizione.

Questo procedimento richiede però la memorizzazione di  $n/8$  numeri complessi. In pratica conviene calcolare ogni volta le radici  $n$ -esime che servono, così è sufficiente memorizzare solo il vettore  $\mathbf{z}$ , che viene modificato durante l'esecuzione e che al termine è uguale a  $\text{IDFT}(\mathbf{z})$ . Si tenga conto però del fatto che il vettore  $\mathbf{z}$  deve essere definito per numeri complessi anche se i dati iniziali sono reali.

**5.64 Algoritmo FFT di Cooley e Tukey** ( $n = 2^s$ , il vettore  $\mathbf{z}$  di  $n$  componenti complesse deve essere già riordinato con il bit reversal, il risultato  $\text{IDFT}(\mathbf{z})$  viene ancora memorizzato in  $\mathbf{z}$ ).

```

m := 1;
for t := 1 to s do begin
  m := 2m;  θ :=  $\frac{2\pi}{m}$ ;  c := cos θ;  s := sin θ;  ωre := 1;  ωim := 0;
  for j := 0 to  $\frac{m}{2} - 1$  do begin
    for h := j to n - 1 step m do begin
      k := h +  $\frac{m}{2}$ ;  ρ := (ωre + iωim)zk;
      zk := zh - ρ;  zh := zh + ρ
    end;
    σ := ωre;  ωre := σc - ωims;  ωim := ωimc + σs
  end
end;
end;

```

Per il calcolo della DFT si può usare un programma analogo, in cui si sostituisce  $\bar{\omega}$  ad  $\omega$  (cioè  $\omega_{re} - \mathbf{i}\omega_{im}$  al posto di  $\omega_{re} + \mathbf{i}\omega_{im}$ ) e si dividono per  $n$  le componenti ottenute. ■

Ovviamente, nel caso che si debbano calcolare più trasformate dello stesso ordine, conviene calcolare le radici  $n$ -esime dell'unità una sola volta.

Si osservi che la struttura ricorsiva della DFT permette di calcolare i coefficienti del polinomio trigonometrico di grado  $2n$  utilizzando integralmente i calcoli fatti per il polinomio trigonometrico di grado  $n$ .

Esistono altri modi di definire un algoritmo FFT, ad esempio è possibile applicare la trasformazione bit reversal al termine, utilizzando l'*algoritmo di Sande e Tukey*, che sfrutta il fatto che la matrice  $V$  è simmetrica (si veda l'esercizio 5.58).

In altre applicazioni è possibile utilizzare metodi che non effettuano trasformazioni bit reversal: ad esempio nel calcolo di un prodotto di convoluzione (si veda l'esercizio 5.60)

$$\mathbf{y} = \mathbf{u} * \mathbf{v}, \quad \text{eseguito con} \quad \mathbf{y} = \text{IDFT}(\text{DFT}(\mathbf{u}) \cdot \text{DFT}(\mathbf{v})),$$

dove il segno  $\cdot$  indica il prodotto componente per componente, si possono calcolare le due trasformate interne con l'algoritmo di Sande e Tukey e la trasformata esterna con l'algoritmo di Cooley e Tukey, non effettuando le permutazioni perché il bit reversal è involutorio.

Il costo computazionale di un algoritmo di FFT può essere leggermente ridotto se, anziché iniziare il calcolo con le trasformate di ordine 2, si inizia direttamente con trasformate di ordine 4 (che non richiedono moltiplicazioni in quanto le radici quarte dell'unità sono  $1, \mathbf{i}, -1, -\mathbf{i}$ ), o con le trasformate di ordine 8 (che possono essere calcolate con un ridotto numero di moltiplicazioni in quanto le radici ottave di 1 che non siano già radici quarte hanno parte reale e immaginaria di modulo uguale a  $\sqrt{2}/2$ ).

Con una DFT di ordine  $n$  di un vettore complesso è possibile calcolare due DFT di ordine  $n$  di vettori reali (si veda l'esercizio 5.52), riducendo alla metà il costo computazionale. Inoltre, con la stessa riduzione del costo, è possibile calcolare la DFT di ordine  $n$  di un vettore reale con una DFT di ordine  $n/2$  di un vettore complesso (si veda l'esercizio 5.53).

Se  $n$  non è potenza di 2 ma è fattorizzabile, è possibile individuare metodi per il calcolo delle DFT e IDFT, che si basano su proprietà analoghe a quelle usate per costruire l'algoritmo 5.64. Si supponga per semplicità che  $n = n_1 n_2$ , con  $n_1 = 3$ ; procedendo in modo analogo a quanto fatto nella dimostrazione del teorema 5.63, si ha

$$\begin{aligned} y_k &= \sum_{j=0}^{n-1} z_j \omega_n^{jk} = \sum_{p=0}^{n_2-1} z_{3p} \omega_n^{3kp} + \sum_{p=0}^{n_2-1} z_{3p+1} \omega_n^{k(3p+1)} + \sum_{p=0}^{n_2-1} z_{3p+2} \omega_n^{k(3p+2)} \\ &= \sum_{p=0}^{n_2-1} z_{3p} \omega_{n_2}^{kp} + \omega_n^k \sum_{p=0}^{n_2-1} z_{3p+1} \omega_{n_2}^{kp} + \omega_n^{2k} \sum_{p=0}^{n_2-1} z_{3p+2} \omega_{n_2}^{kp}. \end{aligned}$$

Quindi la trasformata di ordine  $n_1 n_2$  può essere calcolata mediante  $n_1$  trasformate di ordine  $n_2$ .

Si possono avere notevoli riduzioni del costo computazionale se  $n$  è scomponibile in un elevato numero di fattori primi. Va però osservato che le permutazioni degli indici dei vettori, conseguenti a tali fattorizzazioni, sono, a differenza del bit reversal, assai più difficilmente implementabili. Esistono altri algoritmi, individuati da Winograd per alcuni valori primi di  $n$ , che sfruttano le proprietà di struttura della matrice  $V$  [25].

Gli algoritmi per il calcolo della trasformata discreta di Fourier, sia quello che implementa direttamente la (74) o la (75), sia l'algoritmo FFT sono numericamente stabili. Vale infatti il seguente teorema per la cui dimostrazione si rimanda a [10].

**5.65 Teorema (di Gentleman e Sande).** *Sia  $n = 2^s$  e sia  $\tilde{\mathbf{z}}$  il vettore effettivamente calcolato operando in un'aritmetica con precisione  $u$ . Se il calcolo viene eseguito mediante la (74) risulta*

$$\frac{\|\mathbf{z} - \tilde{\mathbf{z}}\|_2}{\|\mathbf{z}\|_2} \leq k_1 u (2n)^{3/2},$$

mentre se il calcolo viene eseguito mediante l'algoritmo FFT di Cooley e Tukey risulta

$$\frac{\|\mathbf{z} - \tilde{\mathbf{z}}\|_2}{\|\mathbf{z}\|_2} \leq k_2 u \sqrt{n} \log_2 n,$$

in cui  $k_1$  e  $k_2$  sono due costanti (se l'aritmetica utilizzata è in base 2 è  $k_1 = 1.06$  e  $k_2 = 8.5$ ). ■

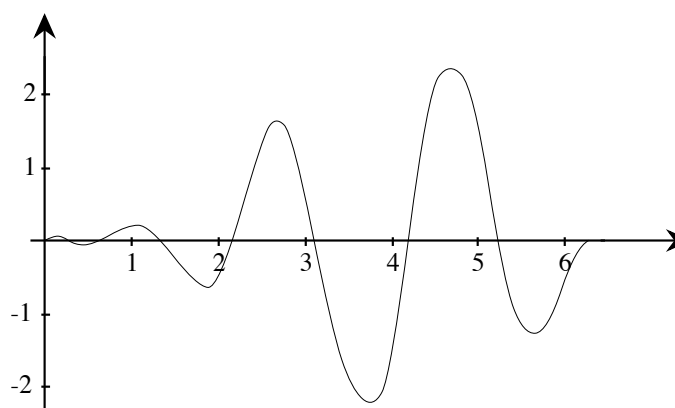
Una tipica applicazione della FFT riguarda il *filtraggio digitale* di un segnale.

**5.66 Esempio.** Sia

$$s(x) = \frac{\alpha_0}{2} + \sum_{j=1}^{\infty} (\alpha_j \cos jx + \beta_j \sin jx)$$

una funzione periodica di periodo  $T$  che rappresenta un *segnale*. Il  $j$ -esimo termine della serie, detto  *$j$ -esima componente o armonica* del segnale, ha modulo massimo  $\rho_j = \sqrt{\alpha_j^2 + \beta_j^2}$ , detto *ampiezza* della  $j$ -esima componente. Si considera una funzione  $r(x)$  che rappresenta un *rumore*. Nella figura 5.21 si riporta il grafico della funzione definita fra  $[0, 2\pi]$  da

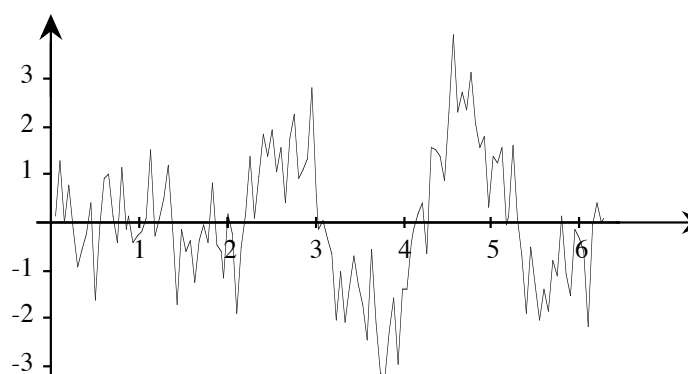
$$s(x) = ((x - 1.2) \sin 3x + \frac{1}{x + 1} \sin(1 + x^2)) \sin \frac{x}{2}$$



**Fig. 5.21** - Segnale  $s(x)$ .

ed estesa per periodicità con periodo  $T = 2\pi$ , nella figura 5.22 si riporta il grafico della funzione  $t(x) = s(x) + r(x)$ , dove la funzione rumore è

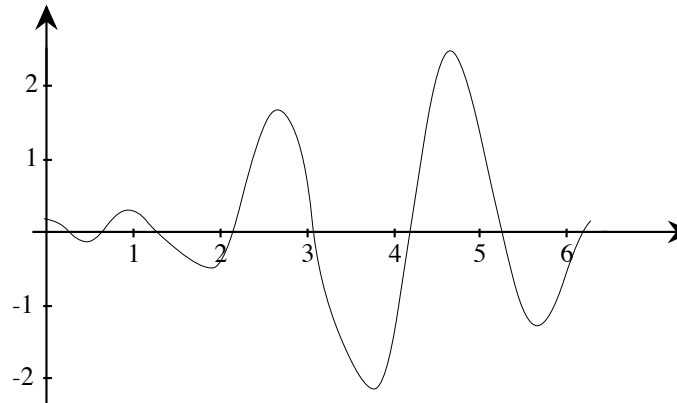
$$r(x) = 0.1 \sin 7x + 2 \sin 23x \cos 31x \sin(1 - 19x).$$



**Fig. 5.22** - Segnale + rumore.

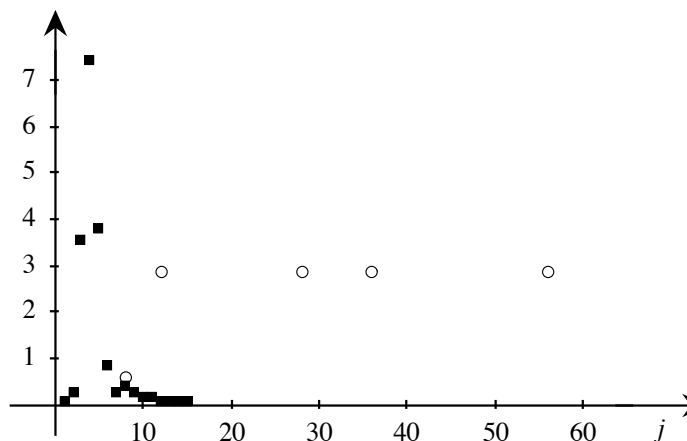


A partire dalla funzione  $t(x)$ , mediante un'operazione di filtraggio è possibile recuperare la maggior parte dell'informazione contenuta nella funzione  $s(x)$ . Infatti, costruendo attraverso la FFT il polinomio trigonometrico  $F_m(x)$  della forma (69) della funzione  $t(x)$  relativo a 128 punti, ed arrestando la sommatoria ad  $m = 8$  (*eliminazione delle alte frequenze*), si ottiene un nuovo polinomio trigonometrico *filtrato*  $F_8(x)$ , il cui grafico è riportato nella figura 5.23.



**Fig. 5.23** - Polinomio trigonometrico filtrato  $F_8(x)$ .

In questo caso la forte rassomiglianza delle funzioni  $s(x)$  e  $F_8(x)$  è dovuta al fatto che le ampiezze massime  $\rho_j$ , con  $j > 8$ , delle componenti di  $s(x)$  sono trascurabili, mentre la funzione  $r(x)$  ha componenti nulle per  $j < 7$ , come illustrato nella figura 5.24 (quadrantini neri per le ampiezze delle componenti di  $s(x)$  e pallini per le ampiezze delle componenti di  $r(x)$ ) ■.



**Fig. 5.24** - Ampiezze massime delle componenti di  $s(x)$  e di  $r(x)$ .