

Per quanto riguarda l'errore di troncamento, ovvero dell'errore introdotto dall'aver assunto che la funzione f sia nulla al di fuori dell'intervallo $[0, T]$ e che quindi coincida con la funzione f_p , fissato h , tale quantità decresce esponenzialmente al crescere di N (e quindi di T) [4]. In generale, in base a tali considerazioni è possibile una scelta opportuna dei parametri N e h in modo da garantire dei risultati accettabili con un costo computazionale non elevato.

5.3 La Trasformata Veloce di Fourier (FFT)

Dalla definizione di DFT di un vettore di N numeri complessi si deduce che la valutazione diretta di ciascuna componente del vettore DFT richiede il calcolo di N moltiplicazioni complesse e $N - 1$ addizioni complesse, ed inoltre la valutazione di N esponenziali complessi. *Supponendo noti gli esponenziali complessi*, il calcolo della DFT di un vettore di lunghezza N richiede, quindi:

$$T(N) = N \cdot [N \text{ molt. complesse} + (N - 1) \text{ add. complesse}] = O(N^2)$$

operazioni floating point. Dunque, a parte la valutazione degli esponenziali complessi, la complessità computazionale del calcolo di una DFT è equivalente a quella di un prodotto matrice-vettore.

Fino al 1965, anno di pubblicazione del primo algoritmo di *Fast Fourier Transform* (FFT) nonostante la già ampia diffusione della DFT, la sua applicazione presentava problemi legati al costo computazionale. L'idea di J. Cooley e J. Tukey, alla base degli algoritmi FFT, fu di *semplificare* opportunamente i calcoli decomponendo il problema, ovvero il calcolo di una DFT di lunghezza N , in sottoproblemi dello stesso tipo ma di dimensione non solo più piccola ma anche tale da consentire di eliminare una buona parte di operazioni inutili e ridondanti. In altre parole, un algoritmo FFT implementa una metodologia di tipo *divide et impera*, applicandola più volte a ciascuno dei sottoproblemi così ottenuti, in modo da riorganizzare efficacemente le operazioni che coinvolgono gli esponenziali complessi.

La difficoltà principale di un algoritmo per il calcolo di una DFT, e più in generale di tutti gli algoritmi numerici utilizzati nell'analisi di Fourier, è sempre stata la valutazione efficiente delle funzioni trigonometriche, indipendentemente dall'ambiente di calcolo a disposizione. Già nel 1925, Witthaker, avendo a disposizione le macchine calcolatrici per eseguire moltiplicazioni, oltre che le tavole trigonometriche, utilizzando la metodologia *divide et impera*, propose uno schema per il calcolo dei coefficienti di Fourier per $N = 6, 12, 24$, che faceva uso solo delle valutazioni negli angoli notevoli. Tale algoritmo, ad esempio per $N = 12$, impiegava solo 44 operazioni. Nel 1958, Goertzel propose un algoritmo basato su formule di ricorrenza per il calcolo di coefficienti di Fourier per un qualsiasi valore di N che utilizzava solo 2 valutazioni trigonometriche (algoritmo di Goertzel (1958), successivamente reso stabile dalla modifica introdotta da Reinsch [13]). Chiaramente questo algoritmo, a differenza dell'algoritmo proposto da

Witthaker, poteva essere eseguito per un *qualsiasi valore di* N , anche se la sua complessità computazionale ($O(N^2)$) ne impediva l'implementazione effettiva per valori di N grandi. Solo l'introduzione degli algoritmi FFT ha consentito l'effettivo sviluppo di quelle applicazioni dell'analisi di Fourier basate sull'approccio computazionale, tra le quali vi sono quelle caratteristiche della disciplina che prende il nome di *digital signal processing* o anche più in generale, di *digital image processing*. In tali applicazioni N è almeno dell'ordine di $O(10^3)$.

Il primo algoritmo FFT, introdotto da Cooley e Tukey, si basa sulla decomposizione degli indici j e k che definiscono rispettivamente le componenti del vettore di input e di quello risultante [4]. Questa decomposizione consente di semplificare le operazioni coinvolte applicando opportunamente le proprietà degli esponenziali complessi. A partire dall'idea introdotta da Cooley e Tukey, l'anno successivo Gentleman e Sande ¹⁹ pubblicarono un articolo che descrive in maniera completa ed esaustiva l'algoritmo FFT, corredandolo di analisi della complessità e della stabilità [11]. Dal 1965 ad oggi, sono disponibili molteplici implementazioni dell'algoritmo FFT, ciascuna specializzata in base a vari fattori come il valore del parametro N , che definisce la lunghezza del vettore DFT, il tipo di fattorizzazione di N , l'applicazione o meno del bit reversal. Il pacchetto software FFTW contiene tali implementazioni, utilizzabili in maniera del tutto *trasparente* perchè sono generate automaticamente da un compilatore interno alla libreria.

5.3.1 L'algoritmo di Cooley e Tukey

L'idea alla base di tutti gli algoritmi FFT è calcolare la DFT di un vettore di lunghezza $N = r \cdot q$ effettuando q DFT di lunghezza r o viceversa.

♣ **Esempio 5.12.** Consideriamo la DFT di un vettore $f = (f_0, f_1, f_2, \dots, f_{11})$ di lunghezza $N = 3 \cdot 4 = 12$:

$$DFT[f]_k := F_k = \sum_{j=0}^{11} f_j w_{12}^{-jk} \quad (5.21)$$

posto:

$$\begin{aligned} k &= 3k_1 + k_0 & k_1 &= 0, 1, 2, 3 & k_0 &= 0, 1, 2 \\ j &= 4j_1 + j_0 & j_1 &= 0, 1, 2 & j_0 &= 0, 1, 2, 3 \end{aligned}$$

allora:

$$F(k) = \sum_{j_0=0}^3 \sum_{j_1=0}^2 f(4j_1 + j_0) w_{12}^{-(4j_1 + j_0)(3k_1 + k_0)} \quad (5.22)$$

Poiché:

$$w_{12}^{-(4j_1 + j_0)(3k_1 + k_0)} = w_{12}^{-12j_1 k_1} w_{12}^{-4j_1 k_0} w_{12}^{-3j_0 k_1} w_{12}^{-j_0 k_0} = w_{12}^{-12j_1 k_1} w_3^{-j_1 k_0} w_4^{-j_0 k_1} w_{12}^{-j_0 k_0}$$

¹⁹Sande era allievo di Tukey.

e $w_{12}^{-12j_1k_1} = 1$, la (5.22) si può scrivere come:

$$F(k) = \sum_{j_0=0}^3 w_4^{-j_0k_1} \underbrace{\sum_{j_1=0}^2 f(4j_1 + j_0)w_3^{-j_1k_0} w_{12}^{-j_0k_0}}_{\substack{\text{DFT di lunghezza 3} \\ 4 \text{ DFT di lunghezza 3}}} \quad (5.23)$$

A parte il fattore esponenziale $w_{12}^{-j_0k_0}$, il calcolo di una DFT di lunghezza 12 si riconduce al calcolo di 4 DFT di lunghezza 3. In particolare, si osserva che i 4 vettori di cui si calcola la DFT, ottenuti da $f(4j_1 + j_0)$ fissando $j_0 = 0, 1, 2, 3$ rispettivamente e facendo variare $j_1 = 0, 1, 2$, sono costruiti considerando le componenti del vettore f i cui indici distano di 4 unità. ♣

Consideriamo la DFT di un vettore $f = (f_0, f_1, f_2, \dots, f_{N-1})$ di lunghezza $N = r_1 \cdot r_2$:

$$F(k) = \sum_{j=0}^{N-1} f_j w_N^{-jk} \quad (5.24)$$

Posto:

$$\begin{aligned} k &= r_1 k_1 + k_0 & k_1 &= 0, 1, \dots, r_2 - 1 & k_0 &= 0, 1, \dots, r_1 - 1 \\ j &= r_2 j_1 + j_0 & j_1 &= 0, 1, \dots, r_1 - 1 & j_0 &= 0, 1, \dots, r_2 - 1 \end{aligned}$$

allora:

$$F(k) = \sum_{j_0=0}^{r_2-1} \sum_{j_1=0}^{r_1-1} f(r_2 j_1 + j_0) w_N^{-(r_2 j_1 + j_0)(r_1 k_1 + k_0)} \quad (5.25)$$

Poiché:

$$w_N^{-(r_2 j_1 + j_0)(r_1 k_1 + k_0)} = w_N^{-N j_1 k_1} w_{12}^{-r_2 j_1 k_0} w_N^{-r_1 j_0 k_1} w_N^{-j_0 k_0} = w_N^{-N j_1 k_1} w_{r_1}^{-j_1 k_0} w_{r_2}^{-j_0 k_1} w_N^{-j_0 k_0}$$

e $w_N^{-N j_1 k_1} = 1$, la (5.25) si può scrivere come:

$$F(k) = \sum_{j_0=0}^{r_2-1} w_{r_1}^{-j_0 k_1} \underbrace{\left[\sum_{j_1=0}^{r_1-1} f(r_2 j_1 + j_0) w_{r_2}^{-j_1 k_0} \right]}_{\substack{\text{DFT di lunghezza } r_1 \\ r_2 \text{ DFT di lunghezza } r_1}} w_N^{-j_0 k_0} \quad (5.26)$$

A meno dell'esponenziale $w_N^{-j_0 k_0}$, il calcolo di una DFT di lunghezza $N = r_1 \cdot r_2$ si riconduce al calcolo di r_2 DFT di lunghezza r_1 . Se poi r_1 si fattorizza nel prodotto $r_2 = r_3 \cdot r_4$, si può applicare tale idea alla DFT di lunghezza r_2 , pervenendo via via a DFT di lunghezza più piccola.

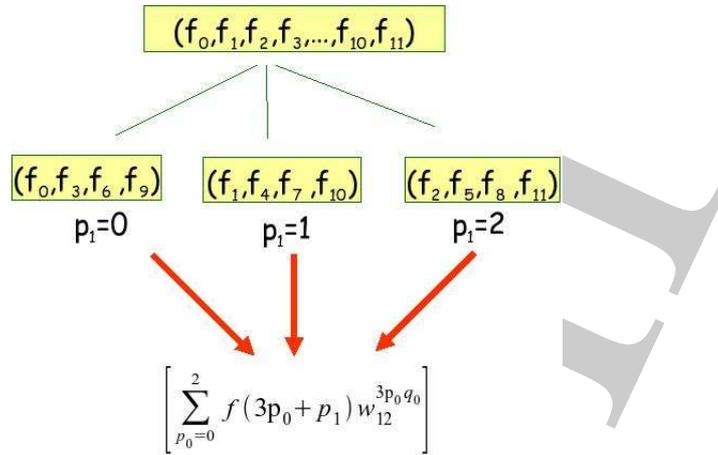


Figura 5.3: Algoritmo di Cooley e Tukey per il calcolo di una DFT di lunghezza 12 mediante la decomposizione in 3 DFT di lunghezza 4. Le componenti del vettore f distano di 3 unità.

♣ **Esempio 5.13.** Consideriamo la DFT di un vettore $f = (f_0, f_1, f_2, \dots, f_{11})$ di lunghezza $N = 4 \cdot 3 = 2^2 \cdot 3 = 12$:

$$DFT[f]_k := F(k) = \sum_{j=0}^{11} f_j w_{12}^{-jk} \quad (5.27)$$

posto:

$$\begin{aligned} k &= 4k_1 + k_0 & k_1 &= 0, 1, 2 & k_0 &= 0, 1, 2, 3 \\ j &= 3j_1 + j_0 & j_1 &= 0, 1, 2, 3 & j_0 &= 0, 1, 2 \end{aligned}$$

si ha:

$$F(k) = \sum_{j_0=0}^2 \sum_{j_1=0}^3 f(3j_1 + j_0) w_{12}^{-(4j_1 + j_0)(3k_1 + k_0)} \quad (5.28)$$

Poiché:

$$w_{12}^{-(3j_1 + j_0)(4k_1 + k_0)} = w_{12}^{-12j_1 k_1} w_{12}^{-3j_1 k_0} w_{12}^{-4j_0 k_1} w_{12}^{-j_0 k_0} = w_{12}^{-12j_1 k_1} w_4^{-j_1 k_0} w_3^{-j_0 k_1} w_{12}^{-j_0 k_0}$$

e $w_{12}^{-12j_1 k_1} = 1$ allora la (5.28) si può scrivere come:

$$F(k) = \sum_{j_0=0}^2 w_3^{-j_1 k_1} \underbrace{\left[\sum_{j_1=0}^3 f(3j_1 + j_0) w_4^{-j_1 k_0} \right]}_{\text{DFT di lunghezza 4}} w_{12}^{-j_0 k_0} \quad (5.29)$$

3 DFT di lunghezza 4

Il calcolo di una DFT di lunghezza 12 si riconduce al calcolo di 3 DFT di lunghezza 4. Come illustrato nella Figura 5.3, in questo caso gli $r_2 = 3$ vettori di lunghezza $r_1 = 4$ di cui si deve calcolare la DFT, ottenuti per $j = 0, 1, 2$ rispettivamente, sono costruiti considerando le componenti del vettore f che distano di $r_2 = 3$ unità.

Applichiamo lo stesso schema alla DFT più interna di lunghezza 4. Introduciamo il vettore F^* di lunghezza 3 così definito:

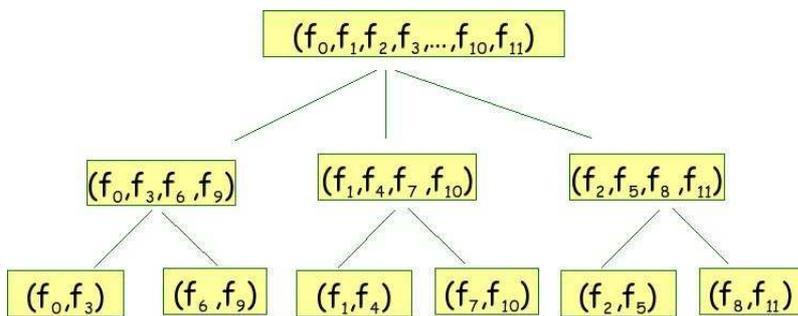


Figura 5.4: Al secondo livello dell'albero, il calcolo della DFT di lunghezza 12 viene ricondotto a quello di 6 DFT di lunghezza 2. La distanza fra le componenti è di $r_1 = 3$ unità.

$$F^*(j_0) = \sum_{j_1=0}^3 f(3j_1 + j_0)w_4^{-j_1 k_0} \quad j_1 = 0, 1, 2, 3 \tag{5.30}$$

Sia:

$$j_1 = 2j'_1 + j'_0 \quad j'_1 = 0, 1, \quad j'_0 = 0, 1$$

allora:

$$F^*(j_0) = \sum_{j'_1=0}^1 w_4^{-j'_0 k_0} \underbrace{\left[\sum_{j'_0=0}^1 f(3j_1 + j_0)w_2^{-j'_1 k_0} \right]}_{\substack{\text{DFT di lunghezza 2} \\ 2 \text{ DFT di lunghezza 2}}} \tag{5.31}$$

ovvero una DFT di lunghezza 4 è espressa come 2 DFT di lunghezza 2. Dalle (5.29) e (5.31) segue quindi che:

$$F(k) = \sum_{j_0=0}^2 w_3^{-j_1 k_1} \underbrace{\sum_{j'_1=0}^1 w_2^{-j'_0 k_0} \left[\sum_{j'_0=0}^1 f(3j_1 + j_0)w_2^{-j'_1 k_0} \right]}_{\substack{\text{DFT di lunghezza 2} \\ 2 \text{ DFT di lunghezza 2}}} w_{12}^{-j_0 k_0} \tag{5.32}$$

$3 \cdot 2 = 6 \text{ DFT di lunghezza 2}$

Si nota che la DFT più interna coinvolge coppie di componenti del vettore f che distano di 3 unità, come anche illustrato in Figura 5.4. In particolare, poiché

$$p_0 = 3j_1 + j_0 = 3(2j'_1 + j'_0) + j_0 = 6j'_1 + 3j'_0 + j_0$$

le 6 DFT vengono effettuate fissando nelle due sommatorie esterne $j_0 = 0$ e $j'_1 = 0$ e quindi utilizzando le coppie di componenti del vettore f in corrispondenza di $p_0 = 0$ e $p_0 = 3$, poi, fissando $j_0 = 0$ e $j'_1 = 1$ si ottiene $p_0 = 6$ e $p_0 = 9$, successivamente fissando $j_0 = 1$ e $j'_1 = 0$, si ha $p_0 = 1$ e $p_0 = 4$, mentre per $j_0 = 1$ e $j'_1 = 1$ si ha $p_0 = 7$ e $p_0 = 10$. Infine, per $j_0 = 2$ e $j'_1 = 0$ segue $p_0 = 2$ e $p_0 = 5$ e per $j_0 = 2$ e $j'_1 = 1$ si ha l'ultima coppia di componenti ovvero $p_0 = 8$ e $p_0 = 11$. ♣

Se assumiamo che N sia divisibile per 2, scegliendo $r_1 = N/2$ e $r_2 = 2$, l'espressione (5.26) si scrive come:

$$F(k) = \sum_{j_0=0}^1 w_{N/2}^{-j_0 k_1} \underbrace{\left[\sum_{j_1=0}^{N/2-1} f(2j_1 + j_0) w_{N/2}^{-j_1 k_0} \right]}_{\substack{\text{DFT di lunghezza } r_1=N/2 \\ 2 \text{ DFT di lunghezza } N/2}} w_N^{-j_0 k_0} \quad (5.33)$$

e lo schema appena descritto porta alla decomposizione del vettore f in 2 vettori di lunghezza $N/2$.

Dalla (5.33) si osserva, in particolare, che fissato $j_0 = 0$ nella sommatoria esterna, l'algoritmo di Cooley e Tukey porta alla decomposizione del vettore f in una parte contenente le componenti con indice $2j_1$, con $j_1 = 0, \dots, N/2$ e quindi le componenti con indice pari, e in un'altra parte, per $j_0 = 1$, contenente le componenti con indice dispari²⁰. Siano quindi $\{f_{2j}\}_{j=0, \dots, N/2}$ e $\{f_{2j+1}\}_{j=0, \dots, N/2-1}$ i due vettori di lunghezza $N/2$ costruiti a partire dal vettore di componenti $\{f_j\}_{j=0, \dots, N-1}$ e contenenti, rispettivamente, il primo le componenti di f corrispondenti a valori pari dell'indice j , il secondo le componenti corrispondenti a valori dispari dell'indice j .

♣ **Esempio 5.14.** Consideriamo il calcolo della DFT di un vettore di lunghezza $N = 16 = 2^4$.

$$DFT[f]_k := F(k) = \sum_{j=0}^{15} f(j) w_{16}^{-jk} \quad k = 0, \dots, 15 \quad (5.34)$$

Fissato k , con $k = 0, \dots, N-1$ si ha:

$$\begin{aligned} F(k) &= \sum_{j=0}^{15} f(j) w_{16}^{-jk} = \sum_{j=0}^7 f(2j) w_{16}^{-2jk} + \sum_{j=0}^7 f(2j+1) w_{16}^{-(2j+1)k} = \\ &= \sum_{j=0}^7 f(2j) w_8^{-jk} + w_8^{-k} \sum_{j=0}^7 f(2j+1) w_8^{-jk} \end{aligned}$$

Osserviamo che, sfruttando la periodicità dell'esponenziale complesso coinvolto, ovvero dei termini w_8^{-jk} e w_8^{-k} , basta calcolare le componenti di $F(k)$ per $k = 0, \dots, 7$, perché le altre 8 si possono ricavare dalle prime 8. Ad esempio,

$$w_8^{-j9} = w_8^{-j(8+1)} = w_8^{-j \cdot 8} \cdot w_8^{-j \cdot 1} = w_8^{-j \cdot 1} \quad (w_8^r = w_8^{r \bmod 8}).$$

²⁰Questa particolare decomposizione fu scoperta già nel 1942 da Danielson e Lanczos [5].

e quindi ci si riconduce al calcolo della componente con indice $k = 1$. Siano, allora,

$$F^E(k) = \sum_{j=0}^7 f(2j)w_8^{-jk}, \quad k = 0, \dots, 7$$

e

$$F^O(k) = \sum_{j=0}^7 f(2j+1)w_8^{-jk}, \quad k = 0, \dots, 7$$

L'algoritmo di Cooley e Tukey, al primo passo, porta alla decomposizione del vettore F come:

$$F(k) = F^E(k) + \omega_8^{-jk}F^O(k), \quad k = 0, \dots, 7$$



Dalla (5.24) si ha:

$$F(k) = \sum_{j=0}^{N-1} f(j)\omega_N^{-jk} = \sum_{j=0}^{N/2-1} f(2j)\omega_N^{-2jk} + \sum_{j=0}^{N/2-1} f(2j+1)\omega_N^{-(2j+1)k}. \quad (5.35)$$

Poiché

$$\omega_N^{-(2j+1)k} = \omega_N^{-2jk} \cdot \omega_N^{-k} = \omega_{N/2}^{-jk} \cdot \omega_N^{-k}$$

segue

$$F(k) = \sum_{j=0}^{N/2-1} f(2j)\omega_{N/2}^{-jk} + \omega_N^{-k} \sum_{j=0}^{N/2-1} f(2j+1)\omega_{N/2}^{-jk}, \quad k = 0, \dots, N-1$$

Siano

$$F^E(k) = \sum_{j=0}^{N/2-1} f(2j)\omega_{N/2}^{-jk}, \quad k = 0, \dots, N/2-1$$

e

$$F^O(k) = \sum_{j=0}^{N/2-1} f(2j+1)\omega_{N/2}^{-jk}, \quad k = 0, \dots, N/2-1$$

le due DFT di lunghezza $N/2$ ottenute a partire dai vettori $\{f_{2j}\}_{j=0, \dots, N/2-1}$ e $\{f_{2j+1}\}_{j=0, \dots, N/2-1}$. La (5.35) diventa quindi:

$$F(k) = F^E(k) + \omega_N^{-k}F^O(k) \quad k = 0, \dots, N/2-1 \quad (5.36)$$

e sfruttando la periodicità dell'esponenziale complesso $\omega_{N/2}$, le altre $N/2$ componenti si ricavano a partire dalle prime $N/2$.

Se N è divisibile per 2^2 , ovvero $r_1 = N/2$ è divisibile per 2, applicando la stessa idea alle due DFT, $\{F^E(k)\}_{k=0, \dots, N/2-1}$ e $\{F^O(k)\}_{k=0, \dots, N/2-1}$, di lunghezza $N/2$, l'algoritmo

di Cooley e Tukey decompone ciascuna DFT in 2 DFT di lunghezza $N/4$, ottenute utilizzando le componenti con indice pari e quelle con indice dispari di ciascuno dei due vettori $\{f_{2j}\}_{j=0,\dots,N/2-1}$ e $\{f_{2j+1}\}_{j=0,\dots,N/2-1}$. Questo comporta che il calcolo del vettore F al primo passo viene ricondotto a quello di 4 DFT di lunghezza $N/4$.

♣ **Esempio 5.15. (Continua l'esempio 5.14).** Consideriamo i due vettori $\{F^E(k)\}_{k=0,\dots,7}$ e $\{F^O(k)\}_{k=0,\dots,7}$ ottenute nell'esempio 5.14. Ciascuna DFT di lunghezza 8 si può decomporre in 2 DFT di lunghezza 4. Ad esempio:

$$\begin{aligned} F^E(k) &= \sum_{j=0}^7 f(2j)w_8^{-jk} = \sum_{j=0}^3 f(4j)w_8^{-2jk} + \sum_{j=0}^3 f(4j+1)w_8^{-(2j+1)k} = \\ &= \sum_{j=0}^3 f(4j)w_4^{-jk} + w_8^{-k} \sum_{j=0}^3 f(4j+1)w_4^{-jk} \end{aligned}$$

Siano

$$F^{EE}(k) = \sum_{j=0}^3 f(4j)w_4^{-jk}$$

e

$$F^{EO}(k) = \sum_{j=0}^3 f(4j+1)w_4^{-jk}$$

le 2 DFT di lunghezza 4 costruite utilizzando i vettori $\{f_{4j}\}_{j=0,\dots,3}$ e $\{f_{4j+1}\}_{j=0,\dots,3}$. Al secondo passo, l'algoritmo di Cooley e Tukey porta alla decomposizione di $\{F^E(k)\}_{k=0,\dots,N/2-1}$ come:

$$F^E(k) = F^{EE}(k) + w_8^{-k} F^{EO}(k) \quad k = 0, \dots, 3$$

e quindi il calcolo del vettore F viene ricondotto al calcolo di 4 DFT di lunghezza 4. ♣

In generale, assumendo che N sia esprimibile come una potenza di 2, ad esempio $N = 2^m$, al passo k il calcolo della DFT di lunghezza N viene ricondotto a quello di k DFT di lunghezza 2^{m-k} , ovvero il vettore f viene decomposto in k vettori di lunghezza 2^{m-k} . Dopo $m - 1$ passi, l'algoritmo di Cooley e Tukey decompone il vettore f in 2^{m-1} coppie di componenti, laddove ciascuna coppia definisce una DFT di lunghezza 2. L'algoritmo così ottenuto è la variante FFT radix-2 di Cooley e Tukey.

♣ **Esempio 5.16. (Continua l'esempio 5.15).**

Riscrivendo ciascuna DFT di lunghezza 4 (ottenuta nell'esempio 5.15) come somma di 2 DFT di lunghezza 2, si ha:

$$F^{EE}(k) = \sum_{j=0}^3 f(j)w_4^{-jk} = \sum_{j=0}^1 f(8j)w_4^{-2jk} + \sum_{j=0}^1 f(8j+1)w_4^{-(2j+1)k} =$$

$$= \sum_{j=0}^1 f(8j)w_2^{-jk} + w_4^k \sum_{j=0}^1 f(8j+1)w_2^{-jk}$$

Siano

$$F^{EEE}(k) = \sum_{j=0}^1 f(8j)w_2^{-jk}$$

e

$$F^{EEO}(k) = \sum_{j=0}^1 f(8j+1)w_2^{-jk}$$

le 2 DFT di lunghezza 2 ottenute considerando rispettivamente le componenti (f_0, f_8) e (f_1, f_9) . In conclusione, per $N = 16 = 2^4$, dopo 3 passi l'algoritmo di Cooley e Tukey porta alla decomposizione seguente:

$$\begin{aligned} F_k &= F_k^E + F_k^O = \\ &= (F_k^{EE} + F_k^{EO}) + (F_k^{OE} + F_k^{OO}) = \\ &= (F_k^{EEE} + F_k^{EEO}) + (F_k^{EOE} + F_k^{EOO}) + (F_k^{OEE} + F_k^{OEO}) + (F_k^{OOE} + F_k^{OOO}) \end{aligned}$$

laddove la prima somma coinvolge 2 DFT di lunghezza 8, la seconda somma coinvolge 4 DFT di lunghezza 4 e l'ultima 8 DFT di lunghezza 2. Questo è lo schema dell'algoritmo FFT radix-2, introdotto da Cooley e Tukey. Si nota la natura ricorsiva dell'algoritmo di Cooley e Tukey. ♣

L'idea di base dell'algoritmo di Cooley e Tukey è *decomporre* ripetutamente il vettore f separando le componenti con indice pari da quelle con indice dispari finché si arriva a 2^{m-1} coppie di componenti. Dopo questa prima fase di decomposizioni successive si passa alla fase di *calcolo* effettivo: partendo dalla DFT più piccola, ovvero quella di lunghezza 2 si calcolano DFT di lunghezza 4 combinando 2 DFT di lunghezza 2, poi, combinando 2 DFT di lunghezza 4, si calcolano DFT di lunghezza 8 e così proseguendo fino a ottenere 1 DFT di lunghezza N .

Questa versione dell'algoritmo prevede da un punto di vista implementativo una fase di *riordinamento* iniziale e una seconda fase di *calcolo* (operazioni butterfly) sul vettore ordinato. In tal modo, questa versione produce il vettore risultante secondo l'ordine naturale.

Questa strategia è anche nota in letteratura con l'appellativo di *decimazione nel tempo*, perché è basata sulla decomposizione del vettore di input, che nell'analisi di Fourier rappresenta usualmente una funzione periodica che evolve nel tempo.

In maniera analoga derivano gli algoritmi FFT radix- r . I più comuni sono quelli per $r = 3$ e per $r = 5$.

♣ **Esempio 5.17.** Consideriamo la DFT di un vettore $f = (f_0, f_1, f_2, \dots, f_8)$ di lunghezza $N = 3 \cdot 3 = 9$:

$$DFT[f]_k := F_k = \sum_{j=0}^8 f_j w_9^{jk} \quad (5.37)$$

Posto:

$$\begin{aligned} k &= 3k_1 + k_0 & k_1, k_0 &= 0, 1, 2 \\ j &= 3j_0 + j_1 & j_1, j_0 &= 0, 1, 2 \end{aligned}$$

si ha

$$F(k) = \sum_{j_0=0}^2 \sum_{j_1=0}^2 f(3j_0 + j_1) w_9^{-(3j_0+j_1)(3k_1+k_0)} \quad (5.38)$$

Poiché:

$$w_9^{-(3j_0+j_1)(3k_1+k_0)} = w_9^{-9j_0k_1} w_9^{-3j_0k_0} w_9^{-3j_1k_1} w_9^{-j_1k_0}$$

e $w_9^{-9j_0k_1} = 1$ la (5.38) si può scrivere come:

$$F(k) = \underbrace{\sum_{j_1=0}^2 w_9^{3j_1k_1} \left[\underbrace{\sum_{j_0=0}^2 f(3j_0 + j_1) w_9^{-3j_0k_0}}_{\text{DFT di lunghezza 3}} \right] w_9^{-j_1k_0}}_{\text{3 DFT di lunghezza 3}} \quad (5.39)$$

Il calcolo di una DFT di lunghezza 9 si riconduce al calcolo 3 DFT di lunghezza 3. ♣

La strategia descritta nel precedente esempio è alla base della classe di algoritmi FFT *radix-3* in cui il vettore f ha lunghezza $N = 3^m$.

5.3.2 L'algoritmo di Gentleman e Sande

Una strategia leggermente diversa da quella alla base dello schema di Cooley e Tukey sottende la versione dell'algoritmo FFT *radix-2* introdotta da Gentleman e Sande. Tale variante si basa sul decomporre ripetutamente il vettore DFT da calcolare. Si osserva che le componenti con indice pari della DFT si possono esprimere come una DFT di lunghezza $N/2$ costruita a partire da un vettore ottenuto combinando le componenti del vettore di input che distano di $N/2$. Analogo ragionamento si fa per le componenti con indice dispari della DFT da calcolare.

♣ **Esempio 5.18.** Consideriamo il calcolo della DFT di un vettore di lunghezza $N = 8 = 2^3$.

$$F(k) = \sum_{j=0}^8 f_j w_8^{-jk} \quad (5.40)$$

Da:

$$F(k) = \sum_{j=0}^8 f_j w_8^{-jk} \leftrightarrow \sum_{j=0}^3 f_j w_8^{-jk} + \sum_{j=0}^3 f_{j+4} w_8^{-(j+4)k} \quad (5.41)$$

separiamo le componenti del vettore F con indice pari da quelle con indice dispari, ovvero consideriamo i due vettori:

$$(F_0, F_2, F_4, F_6) \quad e \quad (F_1, F_3, F_5, F_7)$$

Sfruttando la proprietà di periodicità dell'esponenziale complesso, si ha:

$$F(2k) = \sum_{j=0}^3 [f(j) + f(j+4)] w_4^{-jk}, \quad k = 0, \dots, 3$$

e

$$F(2k+1) = \sum_{j=0}^3 [f(j) - f(j+4)] w_4^{-j} w_4^{-jk}, \quad k = 0, \dots, 3$$

Primo passo: L'espressione di $F(2k)$ e di $F(2k+1)$ suggerisce di costruire i 2 vettori di lunghezza 4:

$$y = (f_0 + f_4, f_1 + f_5, f_2 + f_6, f_3 + f_7) \quad e \quad z = (f_0 - f_4, (f_1 - f_5)w_4^{-1}, (f_2 - f_6)w_4^{-2}, (f_3 - f_7)w_4^{-3})$$

Per cui, i due vettori $\{F(2k)\}_{k=0,1,2,3}$ e $\{F(2k+1)\}_{k=0,1,2,3}$ si ottengono come DFT di lunghezza 4 costruite a partire dai vettori y e z :

$$F(2k) = \sum_{j=0}^3 [f(j) + f(j+4)] w_4^{-jk} = \sum_{j=0}^3 y(j) w_4^{-jk} \quad k = 0, \dots, 3$$

e

$$F(2k+1) = \sum_{j=0}^3 z(j) w_4^{-jk} \quad k = 0, \dots, 3$$

Per ciascuno dei due vettori $\{F(2k)\}_{k=0,1,2,3}$ e $\{F(2k+1)\}_{k=0,1,2,3}$, separiamo il calcolo delle componenti con indice pari da quello delle componenti con indice dispari. Ad esempio, per quanto riguarda $\{F(2k)\}_{k=0,1,2,3}$, consideriamo

$$(F(0), F(4)) = \{F(2k)\}_{k=0,2}$$

e

$$(F(2), F(6)) = \{F(2k)\}_{k=1,3}$$

La coppia $(F(0), F(4))$ si può esprimere come DFT di lunghezza 2 costruita a partire dal vettore di componenti $(y_0 + y_2, y_1 + y_3)$. Cioè:

$$\{F(2k)\}_{k=0,2} = \sum_{j=0}^1 [y(j) + y(j+2)] w_2^{-jk}$$

$$\{F(2k)\}_{k=1,3} = \sum_{j=0}^1 [y(j) + y(j+2)] w_2^{-jk} w_4^{-j}$$

Analogamente, la coppia $(F(2), F(6))$ si può esprimere come una DFT di lunghezza 2 costruita a partire dal vettore di componenti $(y_0 - y_2, (y_1 - y_3)w_4^{-1})$.

Secondo passo: questa analisi suggerisce di costruire i due vettori

$$y' = (y_0 + y_2, y_1 + y_3), \quad y'' = (y_0 - y_2, (y_1 - y_3)w_4^{-1})$$

per cui il vettore di componenti $\{F(2k)\}_{k=0,1,2,3}$ si può ottenere calcolando 2 DFT di lunghezza 2 costruite a partire dai vettori y' e y'' . Cioè:

$$F(2k)_{k=0,2} = \sum_{j=0}^1 y'(j) w_2^{-jk}, \quad F(2k+1)_{k=0,2} = \sum_{j=0}^1 y''(j) w_2^{-jk}.$$

Analogo ragionamento vale per il vettore di componenti $\{F(2k+1)\}_{k=0,1,2,3}$. Separando le componenti con indice pari da quelle con indice dispari, cioè considerando i due sottovettori: $(F(1), F(5))$ e $(F(3), F(7))$, si trova che

$$(F(1), F(5)) = \{F(2k+1)\}_{k=0,2} = \sum_{j=0}^1 [z(j) + z(j+2)] w_2^{-jk} = \sum_{j=0}^1 z'_j w_2^{-jk} \quad k=0,2$$

$$(F(3), F(7)) = \{F(2k+1)\}_{k=1,3} = \sum_{j=0}^1 [z(j) - z(j+2)] w_2^{-jk} w_4^{-j} = \sum_{j=0}^1 z''(j) w_2^{-jk} \quad k=1,3$$

Si costruiscono quindi i vettori:

$$z' = (z_0 + z_2, z_1 + z_3), \quad z'' = (z_0 - z_2, z_1 - z_3)$$

per cui ciascuna coppia di componenti si può esprimere come DFT di lunghezza 2 costruita rispettivamente su z' e z'' .

Il terzo passo comporta infine il calcolo delle DFT di lunghezza 2 costruite al secondo passo, e quindi il calcolo delle quantità:

$$\begin{aligned} y''' &= y'_0 + y'_1, & y^{iv} &= y'_0 - y'_1 \\ y^v &= y''_1 + y''_2, & y^{vi} &= y''_1 - y''_2 \\ z''' &= z'_0 + z'_1, & z^{iv} &= z'_0 - z'_1 \\ z^v &= z''_1 + z''_2, & z^{vi} &= z''_1 - z''_2 \end{aligned}$$

Il vettore $F = DFT[f]$ è:

$$F = (y''', y^{iv}, y^v, y^{vi}, z''', z^{iv}, z^v, z^{vi}) = (F_0, F_4, F_2, F_6, F_1, F_5, F_3, F_7)$$

Come si può osservare, il vettore F ha le componenti non ordinate secondo l'ordine naturale. In effetti, la componente F_4 si trova al posto della componente F_1 e viceversa, la componente F_6 si trova al posto della componente F_3 e viceversa. L'ordinamento con il quale si presentano le componenti del vettore DFT è detto ordinamento del bit inverso (bit reversal), perchè se j e k sono due indici di tale vettore DFT, allora la componente di indice j si trova al posto della componente di indice k se j si ottiene da k invertendo le cifre della sua rappresentazione binaria. ♣

Applicando la proprietà di periodicità dell'esponenziale complesso si ha:

$$F(k) = \sum_{j=0}^{N-1} f(j) w_N^{-jk} = \sum_{j=0}^{N/2-1} f_j w_N^{-jk} + \sum_{j=0}^{N/2-1} f(j+N/2) w_N^{-(j+N/2)k}$$

si ha:

$$F(k) = \sum_{j=0}^{N/2-1} [f(j) w_N^{-jk} + f(j+N/2) w_N^{(-j+N/2)k}] = \sum_{j=0}^{N/2-1} [f(j) + (-1)^k f(j+N/2)] w_{N/2}^{-jk/2}$$

l'ultima eguaglianza sussiste in quanto $w_N^{-N/2} = -1$ e $w_N^{-jk} = w_N^{-jk/2}$

Consideriamo le componenti con indice pari:

$$F(2k) = \sum_{j=0}^{N/2-1} [f(j) + f(j + N/2)w_{N/2}^{-jk}], \quad k = 0, \dots, N/2 - 1$$

e quelle con indice dispari

$$F(2k + 1) = \sum_{j=0}^{N/2-1} [(f(j) - f(j + N/2))w_N^{-j}]w_{N/2}^{-jk}, \quad k = 0, \dots, N/2 - 1$$

Introdotti i vettori $\{y_j\}_{j=0, \dots, N/2-1}$ e $\{z_j\}_{j=0, \dots, N/2-1}$ di componenti

$$y_j = f_j + f_{j+N/2}, \quad z_j = (f_j - f_{j+N/2})w_N^{-j}, \quad j = 0, \dots, N/2 - 1$$

i due vettori di lunghezza $N/2$, $\{F_{2k}\}_{k=0, \dots, N/2}$ e $\{F_{2k+1}\}_{k=0, \dots, N/2}$, si possono riguardare come DFT di lunghezza $N/2$ costruite rispettivamente su $\{y_j\}_{j=0, \dots, N/2-1}$ e $\{z_j\}_{j=0, \dots, N/2-1}$:

$$F = DFT[f] = (\{F_{2k}\}, \{F_{2k+1}\})_{k=0, \dots, N/2} = (DFT[y], DFT[z])$$

Al **primo passo**, l'algoritmo di Gentleman e Sande calcola i due vettori $[y]$ e $[z]$.

Consideriamo ciascuna DFT e separiamo le componenti con indice pari da quelle con indice dispari. I due vettori così costruiti, di lunghezza $N/4$, sono esprimibili in termini di DFT di lunghezza $N/4$:

$$\begin{aligned} DFT[y] &= \sum_{j=0}^{N/2-1} [f(j) + f(j + N/2)w_{N/2}^{-jk}] = \sum_{j=0}^{N/2-1} y_j w_{N/2}^{-jk} = \\ &= \sum_{j=0}^{N/4-1} (y_j + y_{j+N/4})w_{N/4}^{-jk} + \sum_{j=0}^{N/4-1} [(y_j - y_{j+N/4})w_{N/2}^{-j}]w_{N/4}^{-jk} \end{aligned}$$

Se consideriamo le componenti con indice pari del vettore $DFT[y]$:

$$DFT[y](2k) = \sum_{j=0}^{N/4-1} (y_j + y_{j+N/4})\omega_{N/4}^{-jk} \quad k = 0, \dots, N/4 - 1$$

e

$$DFT[y](2k + 1) = \sum_{j=0}^{N/4-1} (y_j - y_{j+N/4})\omega_{N/2}^{-j}\omega_{N/4}^{-jk} \quad k = 0, \dots, N/4 - 1$$

introdotti i vettori di lunghezza $N/4$, $\{y'\}_{j=0, \dots, N/4-1}$ e $\{y''\}_{j=0, \dots, N/4-1}$, di componenti

$$y'_j = (y_j + y_{j+N/4}), \quad y''_j = (y_j - y_{j+N/4})w_{N/2}^{-j},$$

segue che:

$$\{DFT[y]\} = \{DFT[y'], DFT[y'']\},$$

Analogamente, per il vettore $DFT[z]$ contenente le componenti con indice dispari della DFT di f :

$$\begin{aligned} DFT[z](k) &= \sum_{j=0}^{N/2-1} [(f_j - f_{j+N/4})w_{N/2}^{-j}]w_{N/4}^{-jk} = \sum_{j=0}^{N/2-1} z_j w_{N/2}^{-jk} \\ &= \sum_{j=0}^{N/4-1} [(z_j + z_{j+N/4})w_{N/2}^j]w_{N/4}^{jk} + \sum_{j=0}^{N/4-1} [(z_j - z_{j+N/4})w_{N/2}^{-j}]w_{N/4}^{-jk}, \quad k = 0, N/4 - 1 \end{aligned}$$

Introdotti i vettori $\{z'\}_{j=0, \dots, N/4-1}$ e $\{z''\}_{j=0, \dots, N/4-1}$ di componenti

$$z'_j = z_j + z_{j+N/4}, \quad z''_j = (z_j - z_{j+N/4})w_{N/2}^{-j},$$

segue che

$$DFT[z] = \{DFT[z'], DFT[z'']\}$$

Al **secondo passo**, l'algoritmo di Gentleman e Sande calcola i vettori y' , y'' , z' e z'' e il calcolo della DFT di lunghezza N viene ricondotto a quello di 4 DFT di lunghezza $N/4$:

$$\begin{aligned} F &= DFT[f] = \\ &= (DFT[y], DFT[z]) = \\ &= (DFT[y'], DFT[y'']), (DFT[z'], DFT[z'']) \end{aligned}$$

Proseguendo in questo modo, dopo $m = \log(N)$ passi, l'algoritmo di Gentleman e Sande costruisce $N/2$ vettori di lunghezza 2. Ciascuna coppia è una DFT e fornisce una coppia di componenti della DFT del vettore f .

A differenza della variante di Cooley e Tukey, si nota che in questa strategia le DFT calcolate in un certo passo sono ottenute *utilizzando il vettore costruito al passo precedente*, per cui l'algoritmo di Gentleman e Sande prevede inizialmente una fase di calcolo, per un certo numero di passi dipendente dalla dimensione del vettore f , e poi, una volta arrivati a calcolare le $N/2$ DFT di lunghezza 2, è necessaria una fase di riordinamento del vettore DFT, dal momento che tale vettore risulta nell'ordine *scrambled*, ovvero secondo l'ordinamento indotto dal *bit reversal*. Tale ordinamento è così detto perchè la componente di indice j occupa la posizione della componente di indice k e viceversa, se k si ottiene da j invertendo le cifre della rappresentazione binaria di k .

Questa variante è anche nota in letteratura come *decimazione nelle frequenze*, perchè la decomposizione viene effettuata sulle componenti del vettore DFT, e nell'analisi armonica tale vettore fornisce informazioni sulle frequenze contenute nel vettore dato [11].