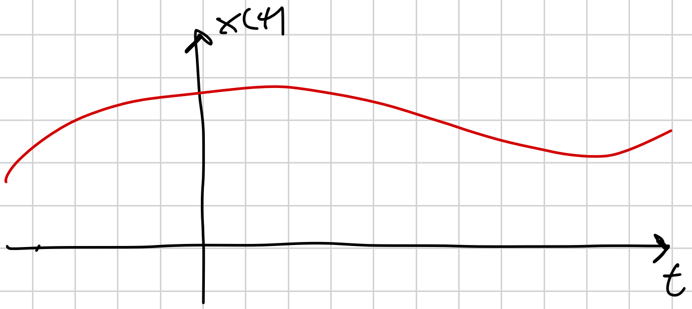
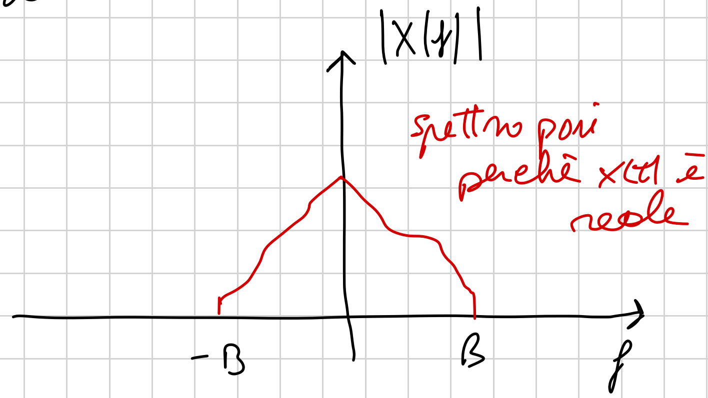
Elaborazione nel dominio della frequenza

Il dominio della frequenza permette di vedere lo stesso oggetto da un punto di vista diverso, permettendo di comprendere meglio il filtraggio fatto nel dominio dello spazio, aggiungere watermark, etc…. Ad esempio è anche possibile riconoscere da quale AI è stata fatta l'immagine vedendo il suo spettro in frequenza.

$$x(t)\in \mathbb{R} \longleftrightarrow X(f)=\int_{-\infty}^{+\infty} x(t)\,e^{-j2\pi f t}\,dt$$



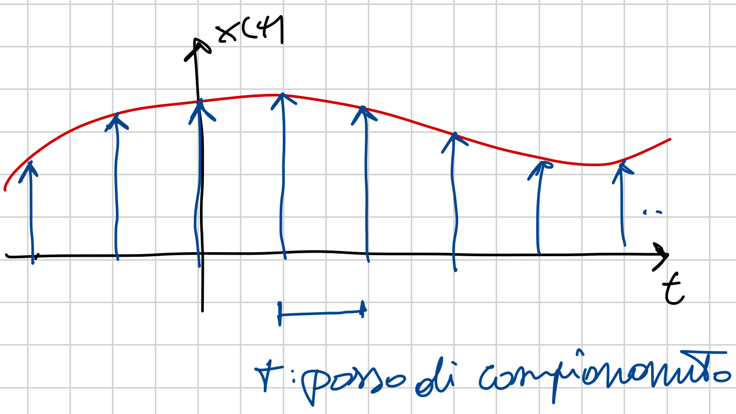


In realtà X(f) non può essere a banda limitata perché lo sarebbe solo se x(t) fosse a durata illimitata. Non esistono nella realtà segnali a banda illimitata. Ciò produce una moltiplicazione per una rect nel tempo e quindi una convoluzione con una sinc in frequenza.

$$x(t)\cdot \Pi\left(\frac{t-t_0}{T}\right) \longleftrightarrow X(f) * \underbrace{\text{sinc}(\dots)\,e^{-j\dots}}_{\text{ILLIMITATA}}$$

Trasformata di Fourier di un segnale Tempo Discreto

I segnali tempo discreto sono quelli che vengono elaborati al computer quindi sono quelli di interesse. In particolare possono essere visti come campionamento di un segnale reale.



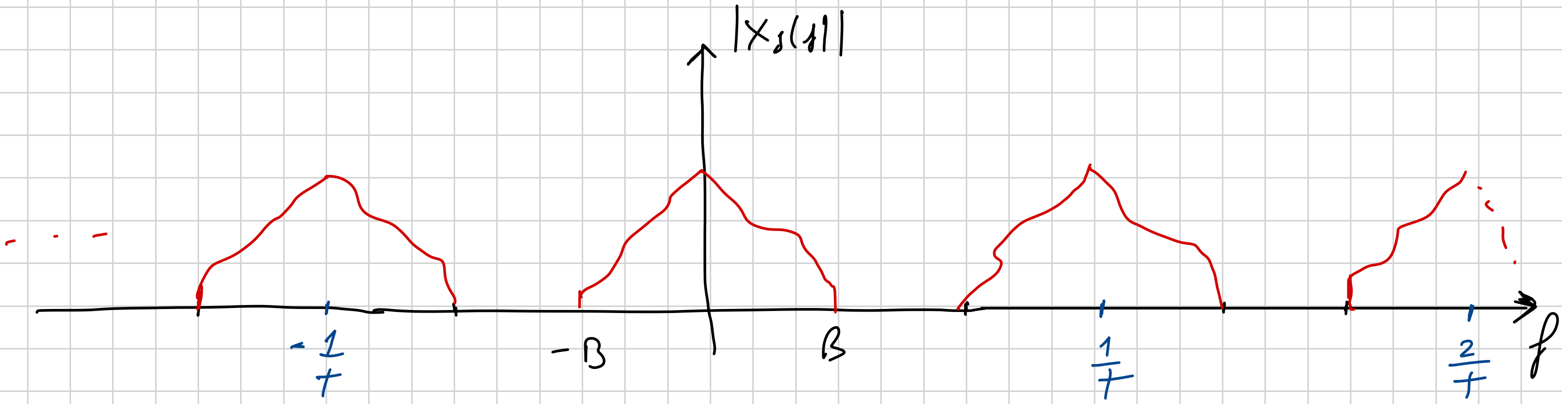
$$x_\delta(t)=x(t)\cdot \sum_n \delta(t-nT)$$

$$x_\delta(t) \longleftrightarrow X_\delta(f) = X(f) * \frac{1}{T}\sum_k \delta\left(f - \frac{k}{T}\right)$$

campionamento nel tempo $\longleftrightarrow$ replica in frequenza

$$X_\delta(f) = \frac{1}{T}\sum_k X\left(f - \frac{k}{T}\right)$$

Quindi sto replicando $X(f)$ nei multipli del periodo.



$\frac{1}{T}$: f_c : frequenza di campionamento

Se f_c è troppo piccolo le repliche andrebbero a sovrapporsi creando l’effetto di *aliasing*, per questo bisogna rispettare la **Condizione di Nyquist** affinché il segnale possa essere ricostruito correttamente:

$$f_c \geq 2B$$

$x_\delta(t)$ può essere riscritto come:

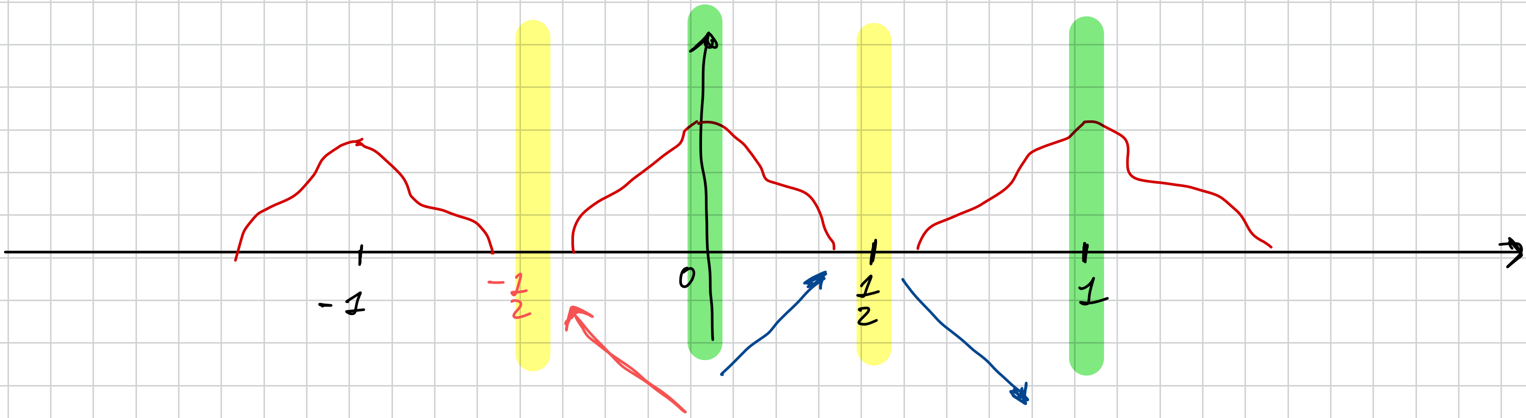
$$x_\delta(t) = x(t)\sum_n \delta(t - nT) = \sum_n x(nT)\,\delta(t - nT)$$

$$\updownarrow$$

$$X_\delta(f) = \sum_n x(nT)\, e^{-j2\pi f nT} = \sum_n x(nT)\, e^{-j2\pi \frac{f}{f_c} n}$$

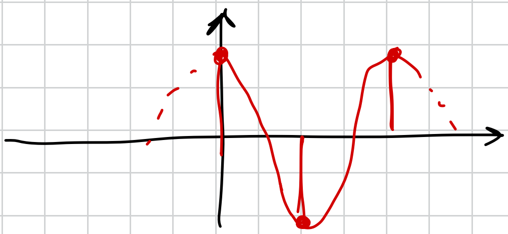
$\frac{f}{f_c}$: frequenza normalizzata : ν

La frequenza normalizzata è meglio perché il segnale in frequenza sarà sempre compreso tra 0 e 1 (tutta l'informazione) perché stiamo dividendo f/f_c.



$$X(\nu) = \sum_n x(n)\, e^{-j2\pi\nu n} \quad : \quad \mathcal{F} \text{ di un segnale TD}$$

E' meglio visualizzarla tra 0 e 1 o tra -1/2 e 1/2? Di default Python lo fa vedere tra 0 e 1. Anzitutto c'è da dire che 1/2 è la **frequenza più alta nel dominio della frequenza per segnali TD** (+infinito nel caso di segnali TC) poiché non esiste qualcosa di più veloce che varia tra -1 e 1(una sinusoide a frequenza massima).



$$\nu_{max} = \frac{1}{2} + k \quad , \quad k \in \mathbb{Z}$$

$$\nu_{min} = 0 + k \quad , \quad k \in \mathbb{Z}$$

Nel TC posso "stretchare" un seno infinitamente.

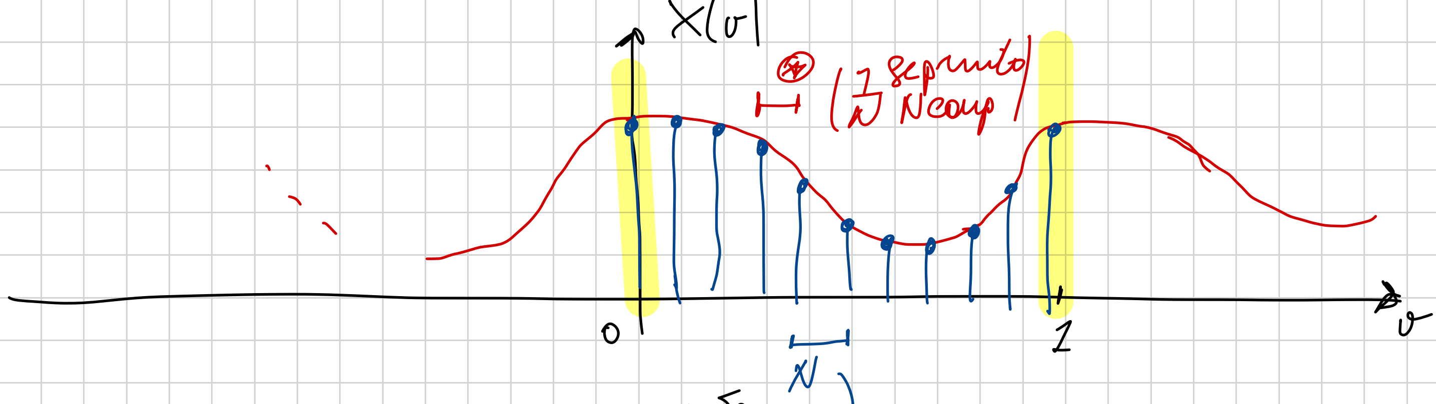
Quindi il contenuto in frequenza che vedrei vedendo da -1/2 e 1/2 sarebbe tipo: prima frequenze alte, poi scendo per frequenze più basse e poi risalgo per frequenze alte. Ho quindi la frequenza più bassa in mezzo. Tra 0 e 1 invece vedrei frequenza bassa poi salgo per frequenze alte e dopo 1/2 riscendo per frequenze basse. **Tra -1/2 e 1/2 risulta comoda cosi da avere al centro le basse e ai lati le alte.**

Problema: per poter rappresentare un segnale TD nel dominio della frequenza servono comunque tutte le frequenze (è un segnale continuo). Quindi non è possibile metterlo all'interno di un computer.

Discrete Fourier Transform (DFT)

E' una trasformata di Fourier discreta applicata ai segnali TD di durata limitata. Serve nel caso in cui vogliamo salvare la trasformata di Fourier su un calcolatore.
Supponiamo di voler **campionare** la trasformata di Fourier di un segnale TD, che supponiamo di durata L:

Vedo tra 0 e 1 perché le formule sono così scritte



Dove $X(v) = \sum_n x(n) e^{-j2\pi v n}$

Quanti campioni prendere?

Il passo di campionamento deve essere abbastanza piccolo (il minimo valore) da poter ricostruire il segnale attraverso l'interpolazione. Quindi dipenderà dall'interpolazione. Suppongo di prendere N campioni di X(v). Nel tempo c'era la condizione di Nyquist che diceva che f_c > 2B con l'interpolazione ideale (quella con la sinc), nel caso attuale sto campionando in frequenza però concettualmente ho sempre bisogno di un numero di campioni tal da poter ricostruire il segnale.

2B rappresenta la durata in frequenza del segnale, quindi Nyquist dice che la frequenza di campionamento deve essere tal da essere almeno pari alla durata in frequenza del segnale. Ragionando dualmente:

N : frequenza di campionamento

$$f_c \geq 2B \Longrightarrow \boxed{N \geq L}$$ Condizione passo di campionamento

Come nel tempo: se prendo $N = L$ mi serve la sinc se faccio $N > L$ posso usare interpolazioni diverse come quella lineare.

DFT: $$X(v)\Big|_{v=\frac{k}{N}} = X\left(\frac{k}{N}\right) \equiv X(k) = \sum_n x(n) e^{-j2\pi \frac{k}{N} n} \qquad k = 0, 1, \dots, N-1$$

campiono X(v)

Se $N < L$ c'è aliasing.

Interpretazione mediante Prodotto scalare e Trasformata lineare

$$X(k) = \sum_{m=0}^{N-1} x(m)\, e^{-j2\pi \frac{k}{N} m}, \qquad k=0,\dots,N-1$$

La formula della DFT è un prodotto scalare? Sì, perché come nel prodotto scalare sto andando a prendere due vettori (x(n) e l'esponenziale) li sto moltiplicando nelle componenti uguali e poi li sommo. Quindi X(v) è il risultato di un prodotto scalare.

Quello che fa la DFT quindi è calcolare la somiglianza tra l'immagine (il segnale) e una sinusoide, il risultato sarà alto quando se somigliano, basso altrimenti. Espandendo la DFT:

$$\begin{cases} ① \; X(0) = \sum_{m=0}^{N-1} x(m)\, e^{0} = \left[x(0) + x(1) + x(2) + \dots + x(N-1) \right] \frac{1}{N} \; (*) \\ ② \; X(1) = \sum_{m=0}^{N-1} x(m)\, e^{-j2\pi \frac{m}{N}} = x(0) + x(1)\, e^{-j\frac{2\pi}{N}} + \dots + x(N-1)\, e^{-j2\pi \frac{(N-1)}{N}} \\ \vdots \\ X(N-1) = \dots = x(0) + x(1)\, e^{-j2\pi \frac{(N-1)}{N}} + \dots + x(N-1)\, e^{-j2\pi \frac{(N-1)(N-1)}{N}} \end{cases}$$

① confronto il segnale con una sinusoide a frequenza nulla (una costante). Se fosse stata la Serie di Fourier ($\frac{1}{N}$ davanti alla formula) avrei avuto la componente continua.

Raggruppo i termini in matrici e posso riscrivere DFT come prodotto matriciale

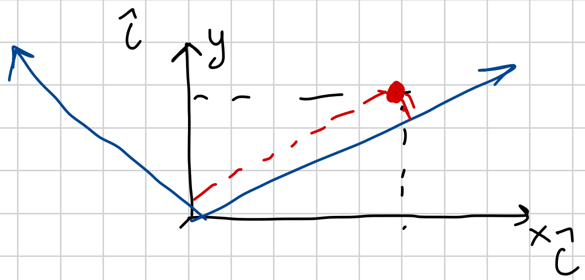
$$\underline{X} = \begin{bmatrix} X(0) \\ \vdots \\ X(N-1) \end{bmatrix} ; \quad x = \begin{bmatrix} x(0) \\ \vdots \\ x(N-1) \end{bmatrix}$$

$$\underline{W} = \begin{bmatrix} 1 & \dots & \dots & \dots & 1 \\ \vdots & e^{-j\frac{2\pi}{N}} & e^{-j2\pi \cdot \frac{2}{N}} & \dots & e^{-j2\pi \frac{(N-1)}{N}} \\ \vdots & & & & \\ 1 & & & & e^{-j2\pi \frac{(N-1)(N-1)}{N}} \end{bmatrix}$$

• $\underline{\underline{W}}$ è simmetrica : $\underline{\underline{W}} = \underline{\underline{W}}^T$

$$\boxed{\underline{X} = \underline{\underline{W}} \cdot \underline{x}}$$ forma matriciale della DFT

La DFT in forma matriciale mostra che i coefficienti della DFT sono ottenuti proprio come proiezione tra i coefficienti del segnale x(n) su una nuova base data dagli elementi della W. W è una matrice costituita da segnali sinusoidali con frequenze via via crescenti quindi ciò che si fa è rappresentare un generico segnale x(n) su una nuova base costituita da sinusoidi. Dato che il prodotto scalare misura il livello di similitudine tra due vettori, il coefficiente X(k) assumerà valori alti se la sinusoide con quella frequenza sarà simile al segnale.



Graficamente è cosi un cambiamento di base, faccio la proiezione sugli assi del vettore da rappresentare. Il cambiamento di base permette di vedere cose che prima non riuscivo a vedere nella vecchia base o vedere meglio cose che prima vedevo male. Non esiste una base migliore in assoluto tant'è che Fourier distrugge i dettagli ad esempio. Le brusche variazioni sono costanti nel dominio della frequenza, e si vedono quindi molto male, nel tempo si vedono meglio.
Anche nelle immagini perdo le brusche variazioni in frequenza, infatti jpeg lavora con blocchetti 8x8.

Perché usare gli esponenziali complessi

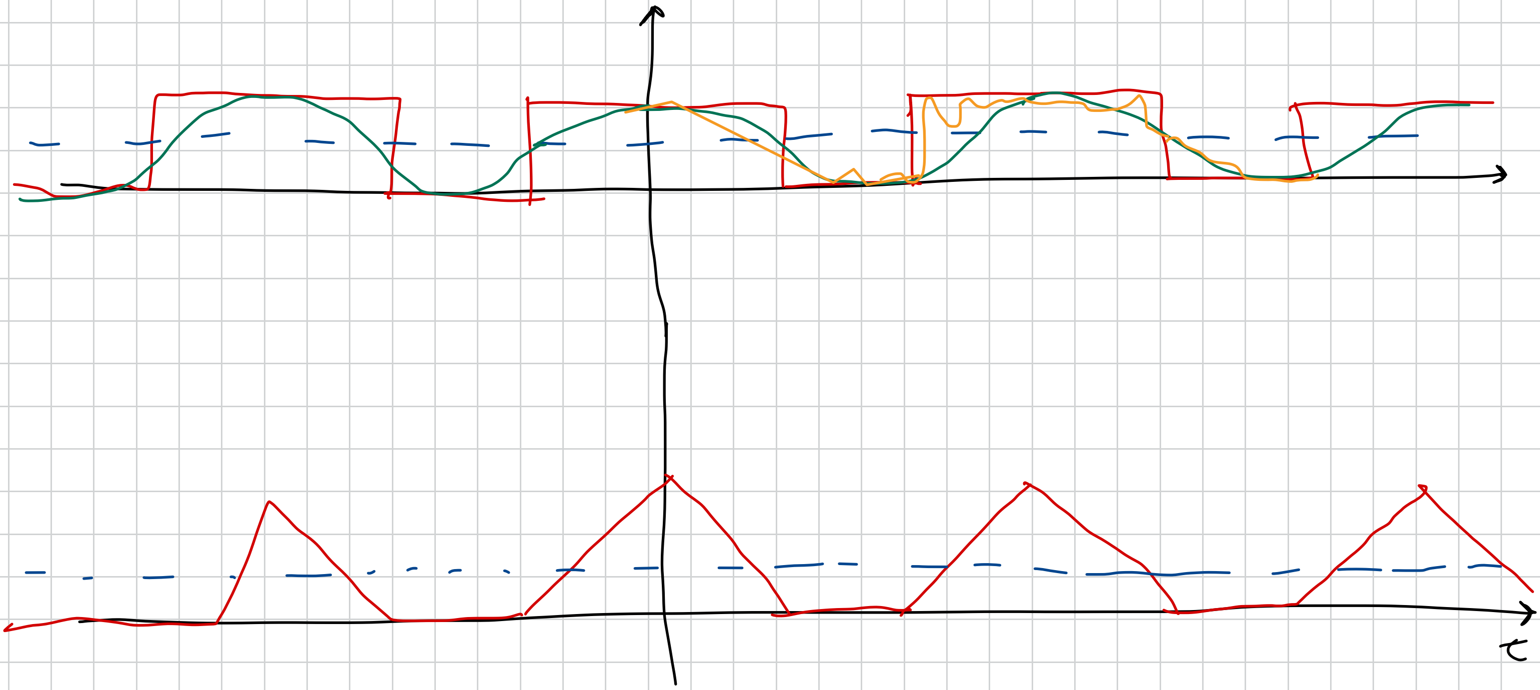
x(t)

$$X(f) = \int_{-\infty}^{+\infty} x(t)\, e^{-j2\pi f t}\, dt \qquad \text{(ANALISI)}$$

$$x(t) = \int_{-\infty}^{+\infty} X(f)\, e^{j2\pi f t}\, df \qquad \text{(SINTESI)}$$

Perché usare esponenziali complessi se x(t) è reale?

Prendo un segnale periodico $x_p(t)$ e la Serie di Fourier:



Fourier afferma che ogni segnale periodo è somma di segnali elementari sinusoidali

$$x_p(t) = A_0 + 2\sum_{k=1}^{+\infty} \tilde{A}_k \underbrace{\cos(2\pi f_0 t + \tilde{\vartheta}_k)}_{\text{Sinusoidi a frequenze multiple della FONDAMENTALE}}$$

A_0: componente continua

$$= A_0 + 2\, A_1 \cos(2\pi f_0 t + \vartheta_1) + A_2 \cos(2\pi (2 f_0) t + \vartheta_2) + \dots$$

- con la prima sinusoide catturo la periodicità del segnale: per questo è detta la fondamentale
- Il termine costante A0 mi serve per traslare su y le sinusoidi (la continua)
- gli altri termini aggiungono mano mano dettagli
- NOTA: anche aggiungendo infinite sinusoidi è impossibile catturare il bordo della rect infatti non si parla di uguaglianza punto punto ma di convergenza.

Ak e thetak sono le grandezze che permettono di effettuare la ricostruzione, sono legate tra loro, rappresentano una **coppia di informazione**, che di solito vengono rappresentare in C perchè è più comodo:

$$A_k e^{j\vartheta_k}$$

$$x_p(t) = A_0 + 2\sum_{k=1}^{+\infty} A_k \cos(2\pi f_0 t + \vartheta_k) \overset{\text{Eulero}}{=} \sum_k X_k e^{+j2\pi f_0 k t}$$

Dove $X_k = A_k e^{j\vartheta_k}$

Le frequenze negative non esistono ovviamente, ma avendo sia quelle positive sia negative nella formula sono sicuro di ritornare a frequenze positive esistenti (formula di Eulero per il coseno)

Quindi esponenziale complesso = FORMA, NON SOSTANZA

$$x(n), \quad X(\nu) = \sum_{n} x(n) e^{-j2\pi\nu n}, \quad \nu = \frac{f}{f_c}$$

CONTINUA perché $x(n)$ è periodico.
PERIODICA di periodo 1 (camp nel tempo)

L'abbiamo campionata:

(DFT) $$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi \frac{k}{N}} \qquad N \geq L, \quad L: \text{lun. segnale}$$

$$\begin{cases} \underline{X} = \underline{\underline{W}} \cdot \underline{x} \\ \underline{x} = \frac{1}{N} \underline{\underline{W}}^* \underline{X} \end{cases}$$

PRODOTTO MATRICIALE (cambio base)

W^{-1} è difficile ma sapendo che $\underline{\underline{W}}$ è ortogonale ho
$W^{-1} = (\underline{\underline{W}}^T)^* = \underline{\underline{W}}^*$

simmetria di $\underline{\underline{W}}$

Perché python non fa prodotto matriciale per la DFT?

Perché W è di dimensioni N-1xN-1 dove N è dipendente dalla lunghezza del segnale in ingresso, quindi ad ogni segnale dovrei ridefinire la matrice e ciò è computazionalmente oneroso.

Trasformata di Fourier bidimensionale

Come nel caso monodimensionale è poco utile definire quella con varaiabili continue, quindi diamo direttamente la definizione della **DFT-2D**

$$X(\mu,\nu) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m,n)\, e^{-j2\pi(\mu m + \nu n)}$$

$$X(k,l) = X(\mu,\nu)\Big|_{\mu=\frac{k}{M},\ \nu=\frac{l}{N}} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m,n)\, e^{-j2\pi\left(\frac{k}{M}m + \frac{l}{N}n\right)}$$

$$k = 0,\dots,N-1 \qquad l = 0,\dots,M-1$$

Forme matriciale

$$\begin{cases} \underline{\underline{X}} = \underline{\underline{W}}\,\underline{\underline{x}}\,W \\ x = \dfrac{1}{MN} W^{*}\,\underline{\underline{X}}\,W^{*} \end{cases}$$

Sinusoidi bidimensionali

Le sinusoidi bidimensionali hanno una direzione al contrario di quelle mono dimensionali.

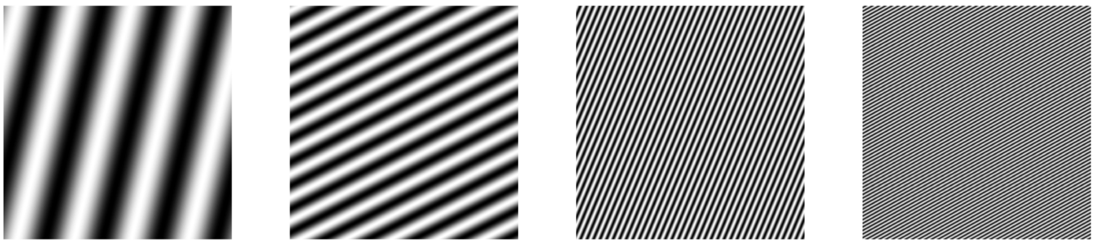


Figura 7: Esempi di sinusoidi bidimensionali $\sin[2\pi(\mu m + \nu n)]$ al crescere di μ e ν.

Nella ricostruzione è più importante la fase rispetto all'ampiezza perché la fase dà il pattern (banalmente se avessi delle fasi sballate nella formula di Fourier verrebbe fuori un segnale che non ci somiglia proprio perché sarebbe sfasato) mentre le ampiezze dicono quanto sia forte la somiglianza con la sinusoide a quella frequenza; Infatti nelle immagini viene visualizzato lo spettro di ampiezza solo perché quello di fase è difficile da interpretare, ma è quello più importante in pratica.

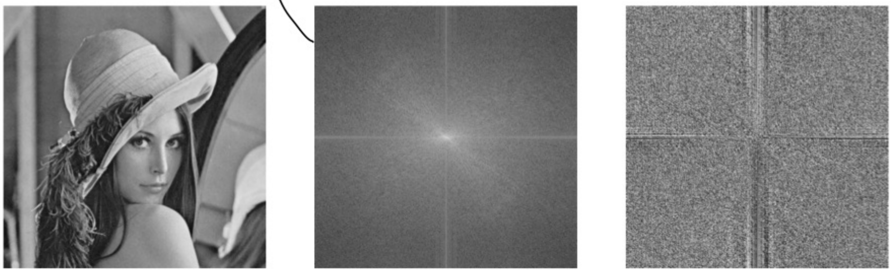


Figura 14: Immagine originale, modulo e fase della trasformata di Fourier.

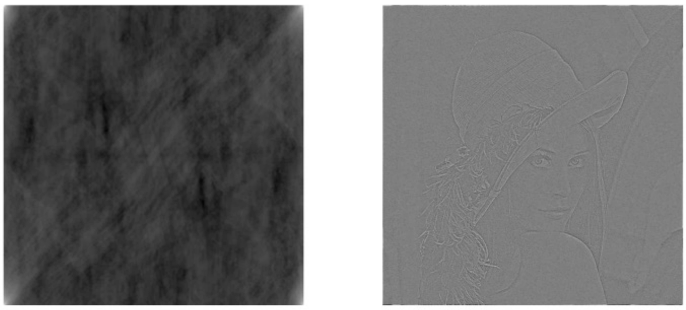


Figura 15: Immagine ricostruita con il solo modulo e con la sola fase della trasformata di Fourier.

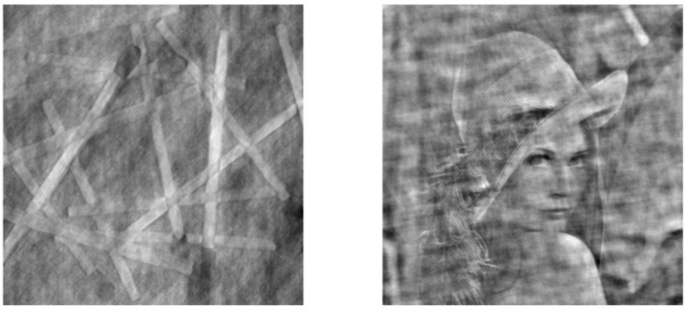
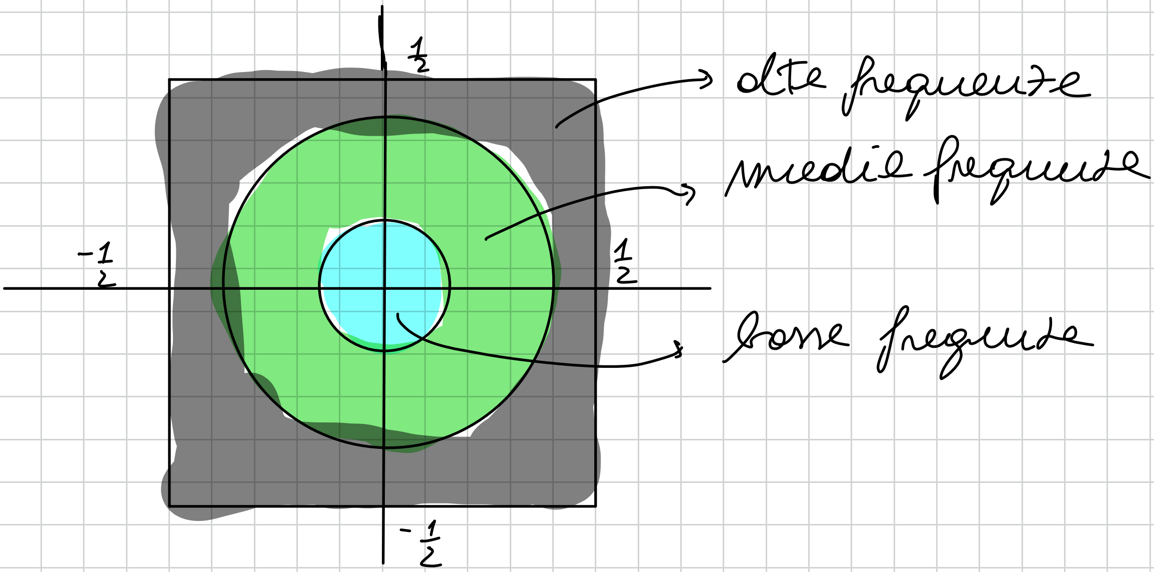


Figura 16: Immagini ricostruite con l'informazione dell'ampiezza e della fase scambiate.

Spettro di Ampiezze

Lo guarderemo tra $-\frac{1}{2}$ e $\frac{1}{2}$ per avere uno schema così:

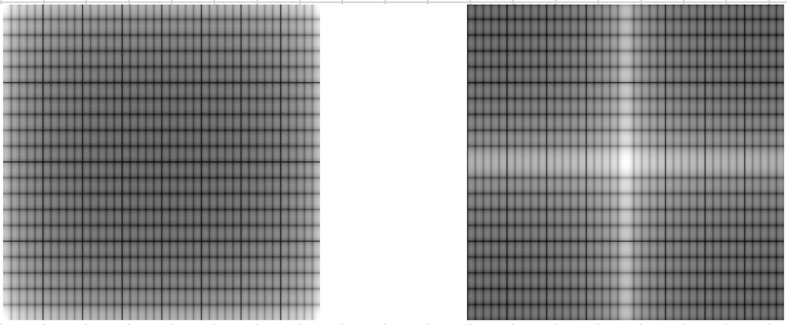


Esempio: finestra rettangolare 2D

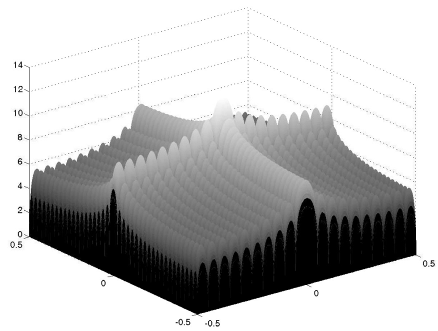
Prendendo una finestra rettangolare bidimensionale si ottiene una trasformata di Fourier scura

Figura 9: Immagine e modulo della trasformata di Fourier.

Questo è dovuto dal fatto che Fourier schiaccia i dettagli, quindi la brusca variazione (dettaglio) è poco visibile anche se alle basse frequenze (ai lati in questo caso) è visibile qualcosa. Se si effettua un enhancement non lineare con un logaritmo il risultato ottenuto è questo:



Si vede che è una sinc bidimensionale dove la griglia nera (nero = zero) sono gli zeri della sinc.



Esempio: pattern in una immagine

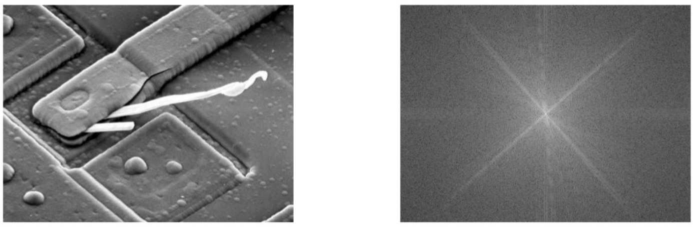


Figura 12: Immagine e trasformata di Fourier.

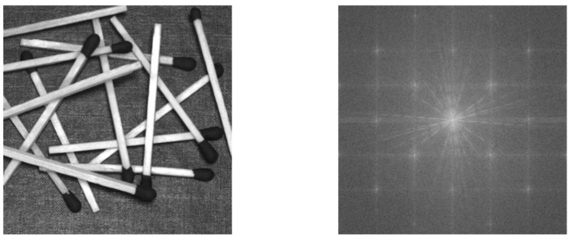


Figura 13: Immagine e trasformata di Fourier.

Esempio: *Immagini con contenuto direzionale.* In figura 12 e 13 si mostrano due immagini in cui sono evidenti strutture fortemente direzionali. In particolare nell'immagine di figura 12 che mostra un circuito integrato danneggiato sono presenti forti discontinuità intorno alla direzione $\pm 45°$, che ritroviamo nelle stesse direzioni delle componenti frequenziali.

Invece, in figura 13 sono presenti molte strutture orientate in direzioni diverse, per cui nella trasformata troviamo delle linee tratteggiate che attraversano il centro in molteplici direzioni. Inoltre, si può osservare una griglia regolare nella trasformata di Fourier. Questa è dovuta alla presenza della texture nell'immagine (lo sfondo su cui si trovano i fiammiferi) che costituisce un pattern regolare ripetitivo e che quindi causa nella trasformata la presenza di impulsi equidistanziati lungo le due direzioni.

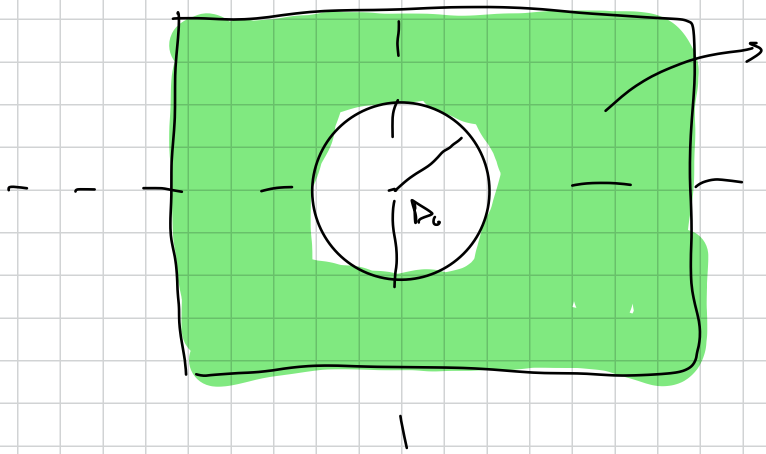
Filtraggio nel dominio della frequenza

$\leq N \to y(n)$ avrà dim $= L+N-1$

Per segnali lineari conviene lavorare in frequenza perché y(n)=x(n)*h(n) e quindi in frequenza ho Y(v) = X(v)H(v), quindi posso progettare H(v) per decidere di far passare o meno certe frequenze, per questo si parla di filtraggio.
Lavorare con segnali lineari conviene (è più facile) ma le CNN sfruttano anche non linearità per questo sono molto potenti.

Filtro passa-basso IDEALE

$$H(\mu,\nu) = \begin{cases} 1 & \sqrt{\mu^2+\nu^2} \leq D_0 \\ 0 & \sim \end{cases}$$



Maschera di 1 per le f. che voglio far passare.

È possibile anche avere una forma diversa, tipo:

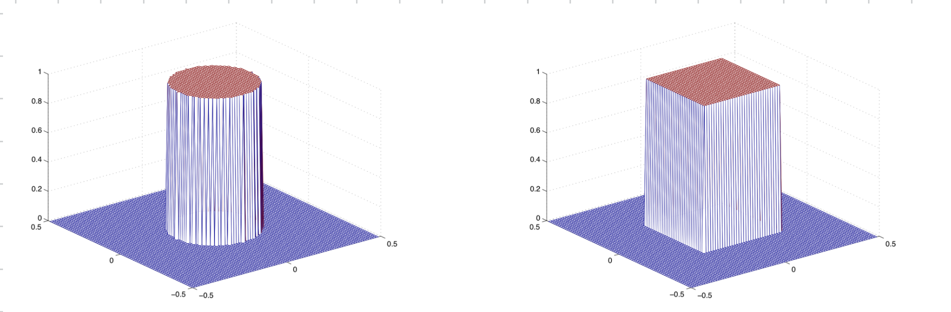


Figura 17: Filtri passa-basso ideali.

$$H_{LP}(\mu,\nu) = \begin{cases} 1 & |\mu| \leq \mu_c \quad |\nu| \leq \nu_c \\ 0 & \sim \end{cases}$$

Filtro passa-alto IDEALE

$$H(\mu,\nu) = \begin{cases} 1 & \sqrt{\mu^2+\nu^2} \geq D_0 \\ 0 & \sim \end{cases}$$

Filtro IDEALE vs Filtro Gaussiano

I filtri ideali lo sono solo dal punto di vista della frequenza, ciò comporta la moltiplicazione per una rect in frequenza, però ricordando che poi vogliamo ritornare nel dominio dello spazio bisogna che ci chiediamo cosa accade lì. Ciò che accade è che le rect moltiplicate diventano delle convoluzioni per delle sinc nello spazio, dando vita all'effetto **ringing,** ovvero si creano oscillazioni intorno agli oggetti dell'immagine

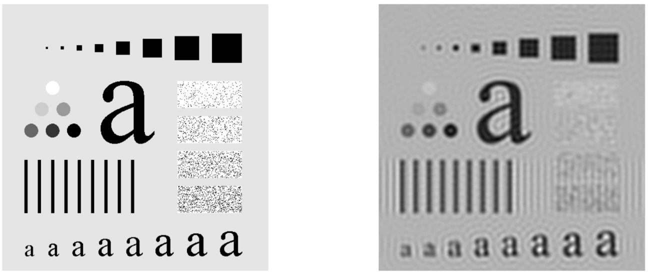


Figura 18: Immagine originale e filtrata con $D_0 = 0.08$.

Mentre se usiamo un filtro gaussiano non abbiamo questo problema perché abbiamo una transizione più dolce verso lo zero:

$$H(\mu,\nu) = e^{-2\pi\sigma^2(\mu^2+\nu^2)}$$

$$h(m,n) = \frac{1}{2\pi\sigma^2} e^{-\frac{(m^2+n^2)}{2\sigma^2}}$$

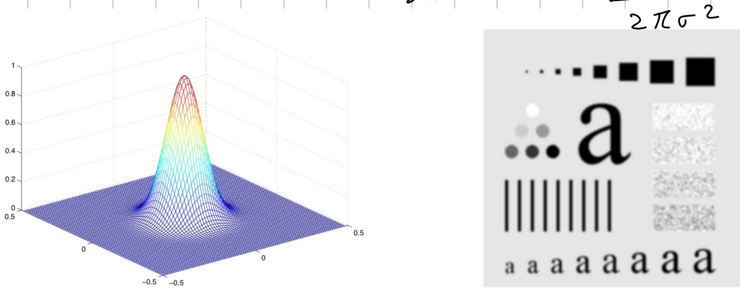
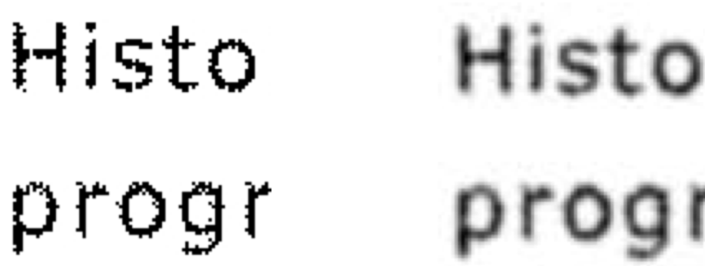


Figura 20: Filtro gaussiano con $\sigma = 0.08$ e immagine filtrata.



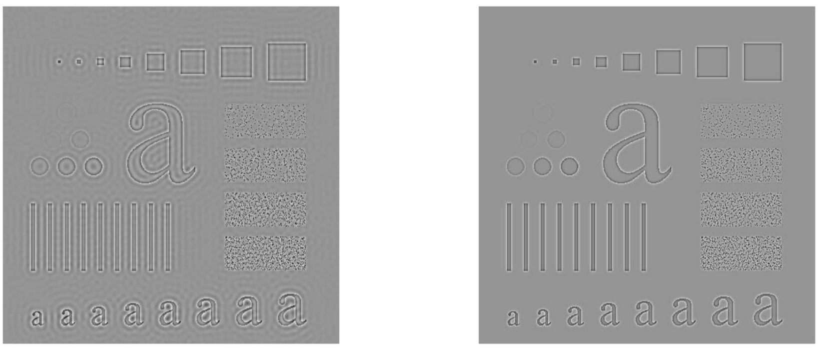


Figura 24: Filtraggio passa-alto ideale con $D_0 = 0.10$ e gaussiano con $\sigma = 0.10$.

Filtro passa-banda: diff. tra due LP filter