

# Relazione progetto calcolo numerico

## Introduzione

Errore. Il segnalibro non è definito.

<b>Il dataset</b>	3
Immagini geometriche e scientifiche	3
Kernel 5x5 = 0.5	3
Kernel 7x7 = 1	3
Kernel 9x9 = 1.3	4
<b>Analisi risultati</b>	4
Raccolta dati	4
Variazione kernel e sigma	5
Rumore	7
Commento	9
Soluzione naive	9
Soluzione con termine di regolarizzazione	9
Valore parametro di regolarizzazione	10
<b>Analisi medie e deviazioni standard di PSNR ed MSE</b>	12
PSNR	13
MSE	13
<b>Proprietà dei metodi del gradiente</b>	14
Prima esecuzione	14
Seconda esecuzione	16
<b>Conclusione</b>	18

# Introduzione

Il problema di deblur consiste nella ricostruzione di un'immagine a partire da un dato acquisito mediante il seguente modello:

$$b = Ax + \eta$$

Dove  $b$  rappresenta l'immagine corrotta,  $x$  l'immagine originale che vogliamo ricostruire,  $A$  l'operatore che applica il blur Gaussiano ed  $\eta$  il rumore additivo con distribuzione Gaussiana di media 0 e deviazione standard  $\sigma$ .

Il dataset utilizzato consiste di 8 immagini in scala di grigi composte da semplici figure geometriche e altre due immagini di ambito astronomico e biologico. Le immagini sono state degradate applicando 3 differenti combinazioni di kernel e sigma per l'impostazione dell'operatore di blur e 3 variazioni della deviazione standard per l'aggiunta di rumore Gaussiano.

Il primo algoritmo di risoluzione impiegato minimizza la funzione  $f(x)$  utilizzando il metodo dei gradienti coniugati implementato dalla funzione `scipy.optimize.minimize`, Funzione obiettivo e gradiente sono dunque:

$$f(x) = \frac{1}{2} \|Ax - b\|_2^2$$

$$\nabla f(x) = ATAx - ATb$$

A causa del mal condizionamento del problema, amplificato dalla perturbazione aggiunta dall'introduzione del rumore, il termine della regolarizzazione di Tikhonov viene incluso nel sistema con parametro di Tikhonov  $\lambda$ .

Le funzioni divengono:

$$f(x) = \frac{1}{2} \|Ax - b\|_2^2 + \frac{\lambda}{2} \|x\|_2^2$$

$$\nabla f(x) = ATAx - ATb + \lambda x$$

Il problema regolarizzato viene minimizzato utilizzando due differenti algoritmi: gradienti coniugati e un'implementazione personalizzata del metodo del gradiente.

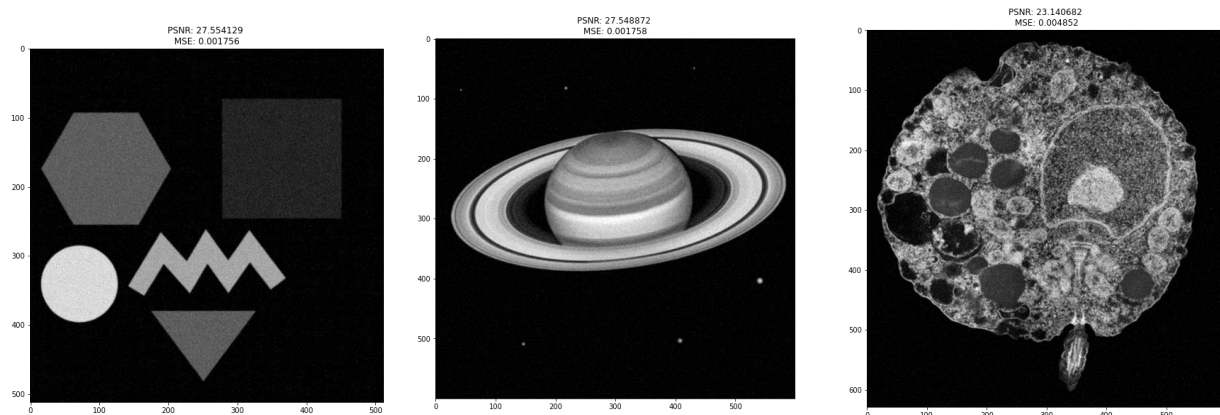
*Distinti d'ora in poi come Tikhonov (CG) e Tikhonov (G)*

# Il dataset

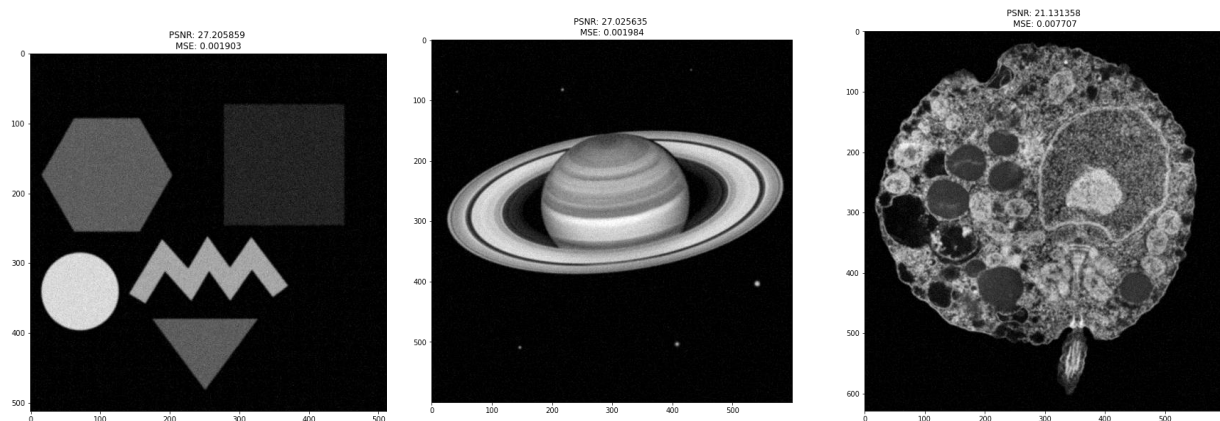
## Immagini geometriche e scientifiche

Queste 10 immagini sono state degradate e corrette proceduralmente dai loro valori di PSNR e MSE rispetto all'originale, secondo le 6 differenti configurazioni di kernel, sigma e deviazione standard. Le immagini sono allegate alla consegna, inseriamo alcuni esempi.

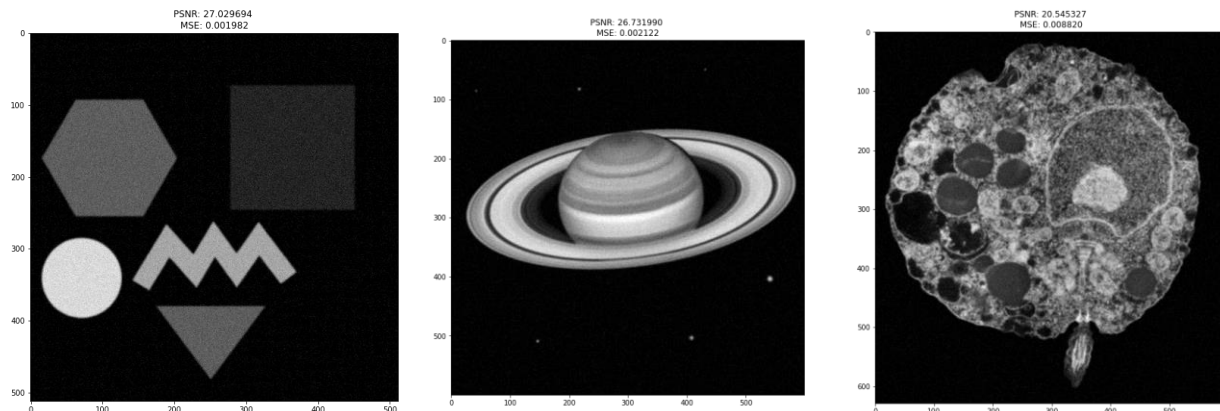
### Kernel 5x5 $\sigma = 0.5$



### Kernel 7x7 $\sigma = 1$



## Kernel 9x9 $\sigma = 1.3$



*Le 162 immagini generate presenti nella cartella “dataset” seguono le variazioni del parametro di blur sopra indicate combinate con tre differenti configurazioni della deviazione standard per generare rumore gaussiano:*

*$\sigma = 0.001$ ,  $\sigma = 0.025$  e  $\sigma = 0.05$ ; i file sono nominati secondo la struttura kernel-decimali  $\sigma + d$  se appartengono alle immagini del dataset geometrico. Nelle analisi dei risultati che seguono le immagini considerate sono quelle esposte negli esempi e riferite in ordine come: Immagine dataset(1), immagine astronomia(2), immagine biologia(3) 600x600*

## Analisi risultati

### Raccolta dati

Le seguenti analisi dei tre differenti approcci si basa su dati raccolti in diverse esecuzioni effettuate variando le immagini, i parametri di corruzione e il parametro lambda; in particolare l'utilizzo di dati in ingresso vari ha permesso di evidenziare le performance degli algoritmi impiegati per effettuare la ricostruzione. Le immagini processate dagli algoritmi sono riproducibili nel software fornito, in questa relazione utilizzeremo il peak signal to noise ratio e il mean squared error per valutare la qualità dell'immagine ricostruita a partire dalla corrotta rispetto all'originale; i risultati sono commentati dopo l'esposizione dei grafici.

I dati sono stati raccolti mediante una versione parametrizzata del codice in grado di effettuare autonomamente i test e di esportare i risultati in tabelle csv mediante la libreria pandas e la sua struttura dati 'DataFrame', le tabelle sono state utilizzate fedelmente per realizzare i seguenti grafici riassuntivi.

*Nota: nei test effettuati e riportati gli algoritmi sono stati impostati per ottimizzare il risultato:*

*Le iterazioni della funzione minimizzare sono state limitate a 6-10 e il parametro di regolarizzazione per entrambe le implementazioni fissato a  $10e-4$ .*

# Variazione kernel e sigma

Immagine dataset (1)

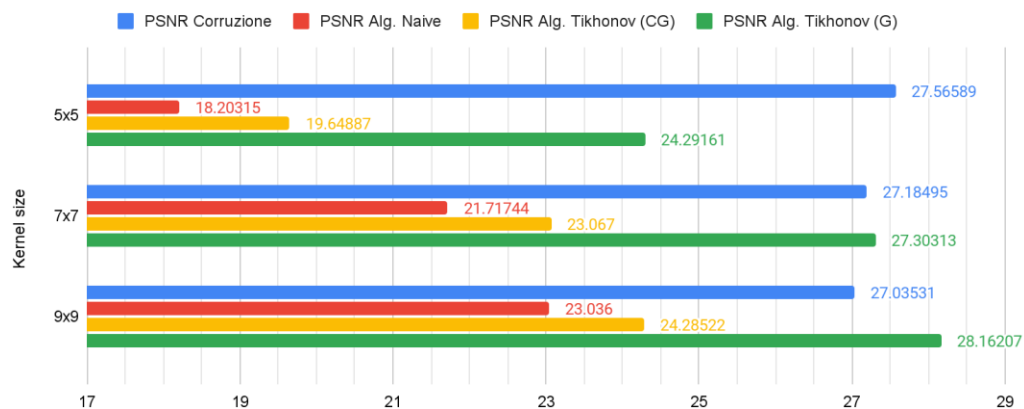


Immagine astronomia (2)

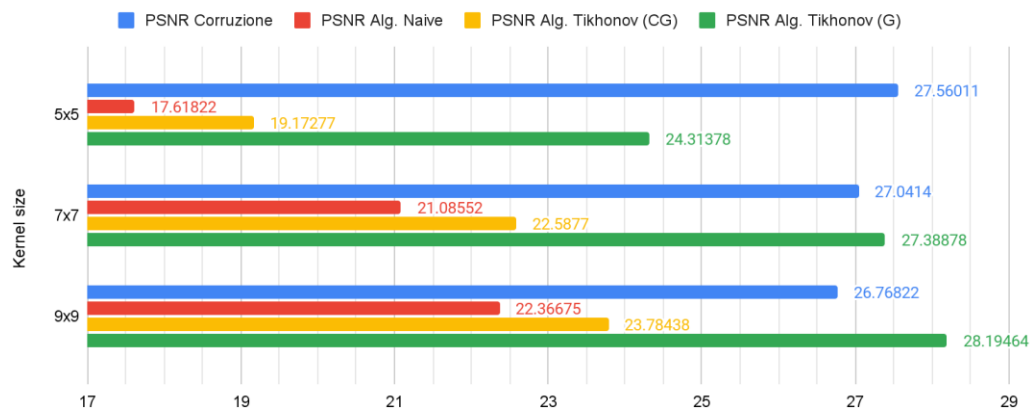
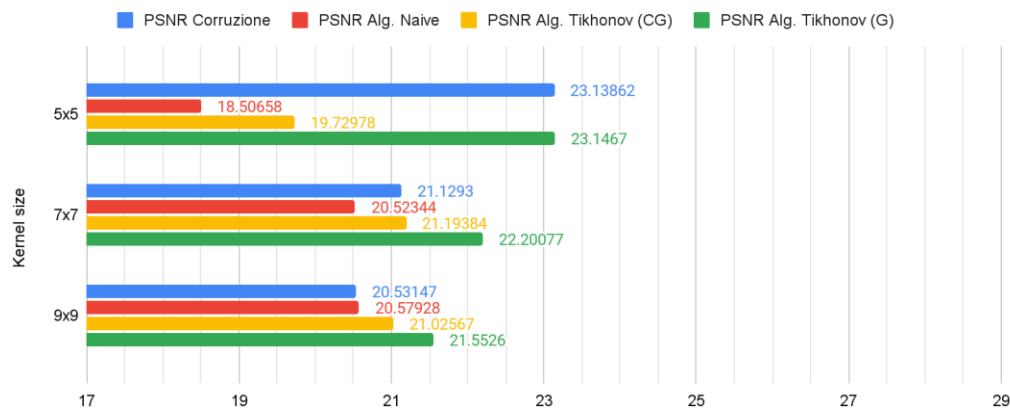
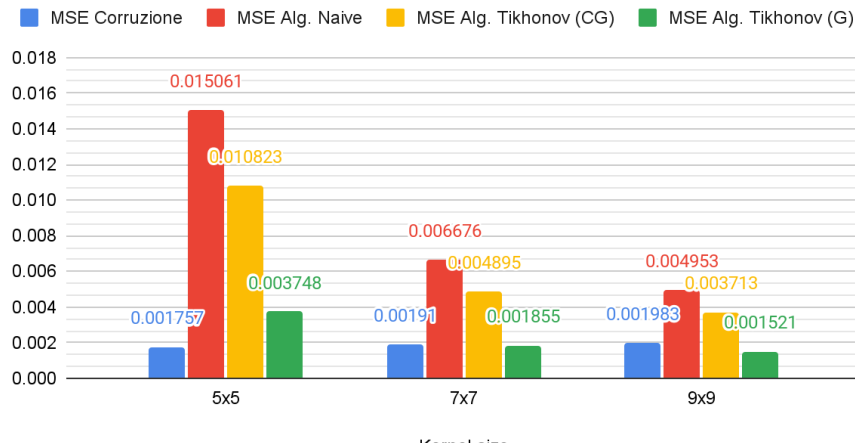


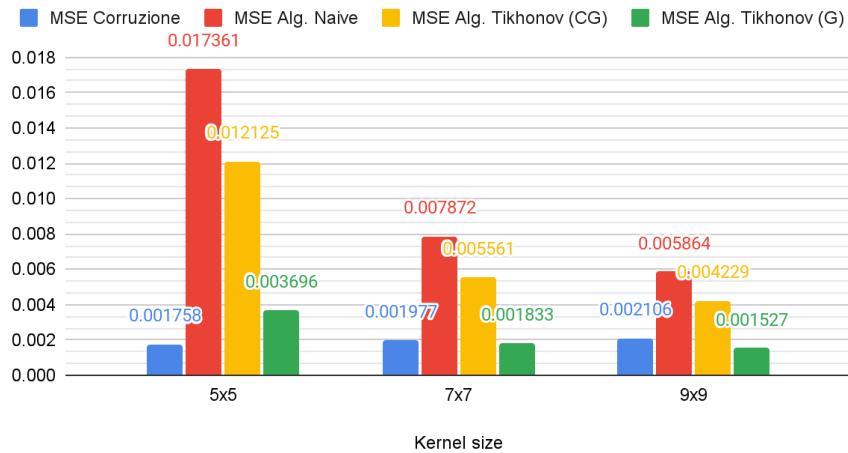
Immagine biologia (3)



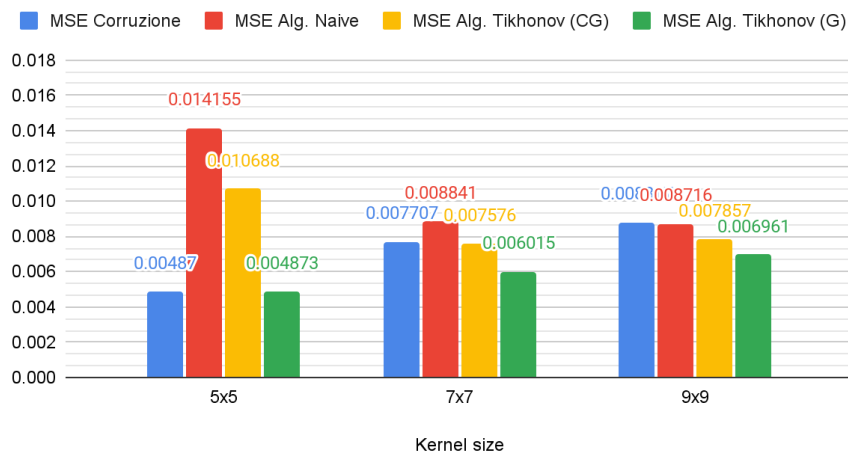
### Immagine dataset (1)



### Immagine astronomia (2)

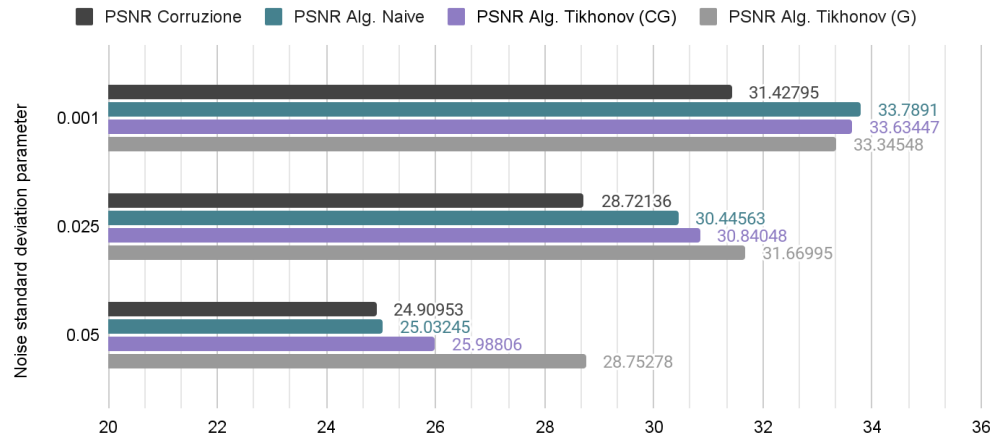


### Immagine biologia (3)

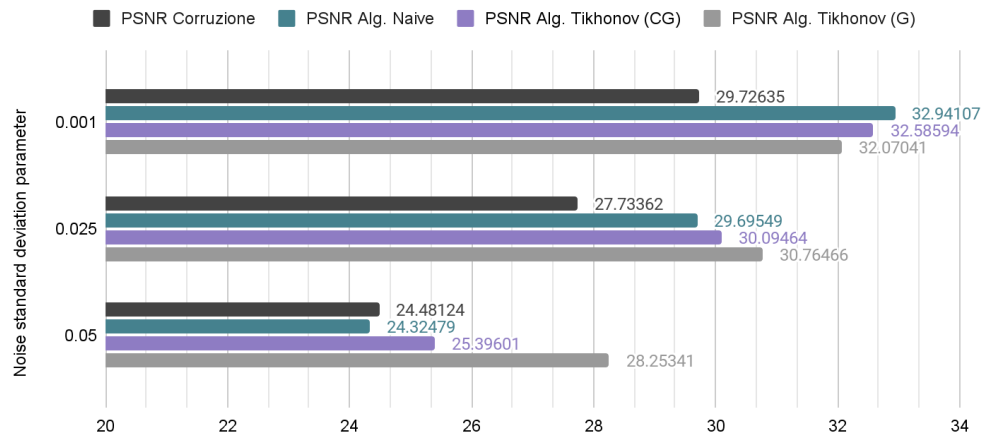


# Rumore

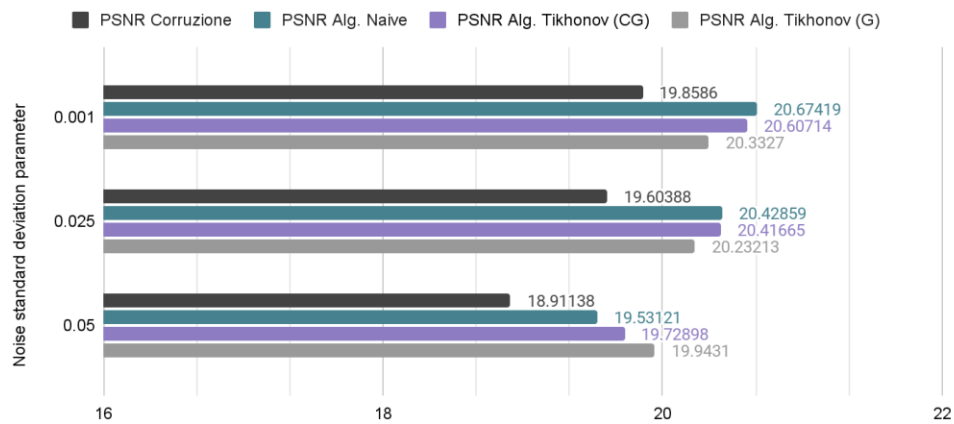
## Immagine dataset (1)



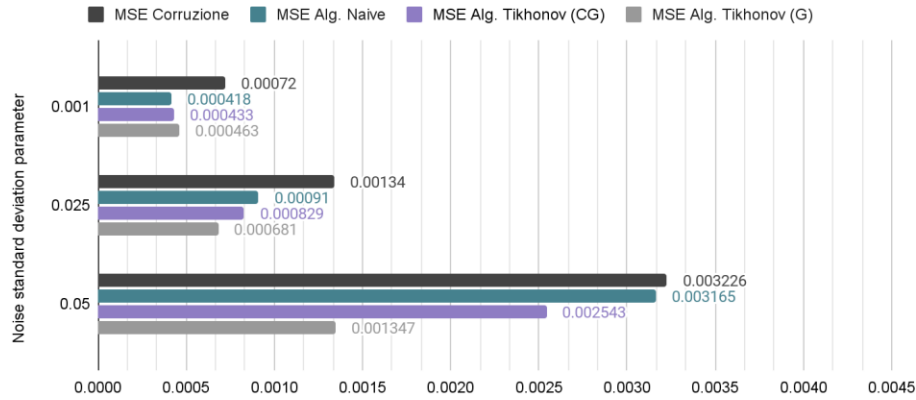
## Immagine astronomia (2)



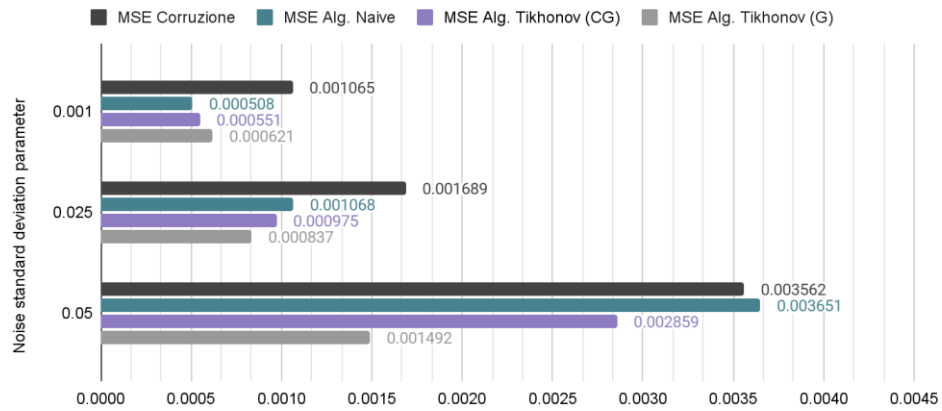
## Immagine biologia (3)



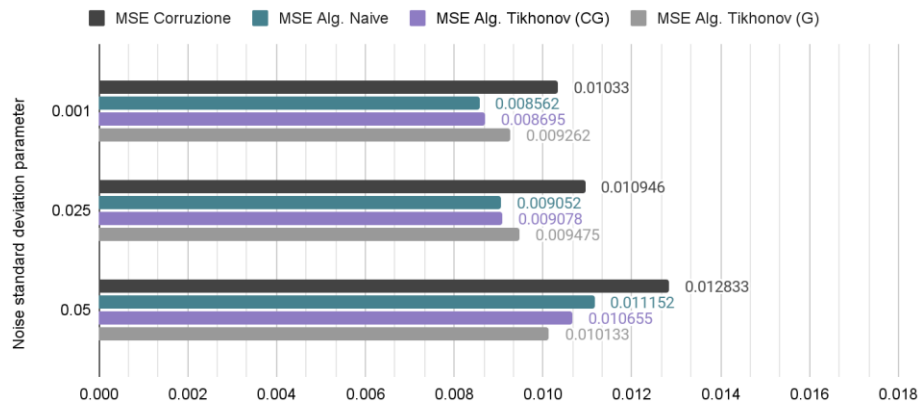
### Immagine dataset (1)



### Immagine astronomia (2)



### Immagine biologia (3)





## Commento

### Soluzione naive

Osservando i risultati dell'algoritmo naive appare evidente come il mal condizionamento del problema influenzi la soluzione, questo è reso evidente da quanto ottenuto con valori bassi della deviazione standard della funzione di blur gaussiano; la matrice di convoluzione ottenuta diviene composta da valori più alti che conseguono in una immagine corrotta più complessa da ricostruire in particolare per la prima soluzione; gli effetti del rumore (impostato con  $\sigma = 0.04$  per tutti gli esperimenti) delineano chiaramente come il rumore condizioni negativamente il sistema e porti il problema di ottimizzazione non vincolata a un'alta instabilità su tutti i risultati, per valori di rumore generati con deviazione standard  $\sigma \leq 0.025$  l'algoritmo risulta tanto efficace quanto i metodi con regolarizzazione.

### Soluzione con termine di regolarizzazione

La ricostruzione dell'immagine con rumore aggiunto beneficia notevolmente dall'introduzione del termine di regolarizzazione, portandoci a risolvere un problema di Tikhonov in cui l'opportuna scelta del parametro  $\lambda \in (0, 1)$  permette di compensare il mal condizionamento del sistema. L'efficacia di questo metodo è evidente per tutte le variazioni dei parametri di blur e in particolare per valori alti della deviazione standard utilizzata nella generazione del rumore. La minimizzazione di questo problema è stata effettuata utilizzando due differenti algoritmi: gradiente coniugato e gradiente; il primo, implementato nella libreria *scipy*, raggiunge l'approssimazione ottimale dell'immagine nella metà delle iterazioni rispetto all'algoritmo del gradiente, ma altrettanto velocemente se ne allontana peggiorando il risultato, essendo una funzione implementata esternamente non ci è stato possibile configurarla a piacimento; il metodo del gradiente, nonostante il maggior numero di iterazioni richieste, ha portato i risultati migliori: riuscendo sempre ad abbassare l'errore dell'immagine ricostruita rispetto alla corrotta e producendo immagini visivamente chiare a prescindere dai parametri utilizzati per il blur e l'aggiunta di rumore, in particolare nella maggior parte dei casi con rumore generato da un valore  $\sigma < 0.03$  l'algoritmo è stato in grado di convergere alla soluzione ottimale in un limitato numero di iterazioni inferiore alle massime specificate.

La scelta del parametro di regolarizzazione utilizzato nei test è stata effettuata dopo le analisi qui sotto riportate che hanno evidenziato come un valore di  $\lambda$   $10^{-5} < \lambda < 10^{-3}$  sia il giusto compromesso, migliorando il condizionamento del problema senza allontanare la soluzione da quella del problema originale.

# Valore parametro di regolarizzazione

Immagine dataset (1)

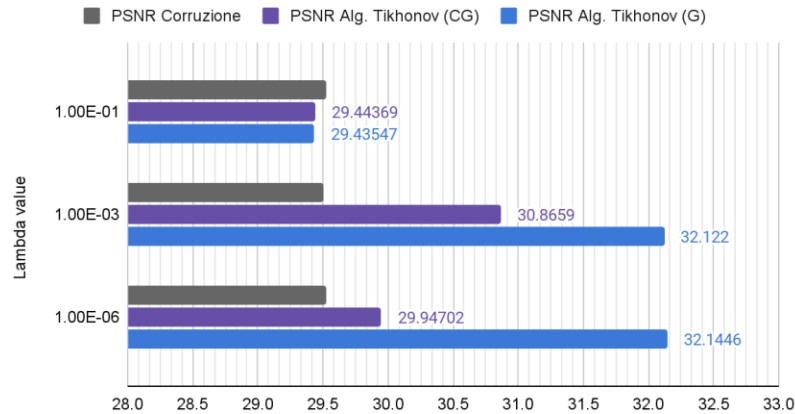


Immagine astronomia (2)

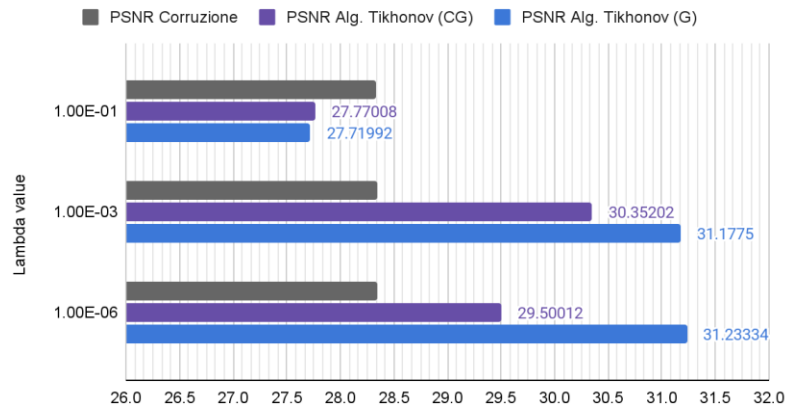


Immagine biologia (3)

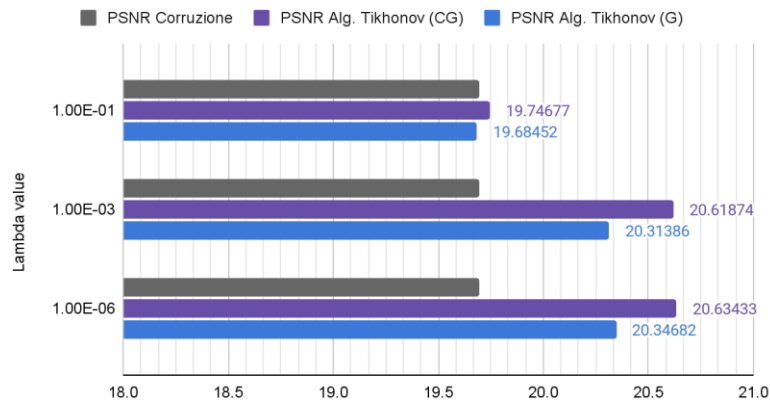


Immagine dataset (1)

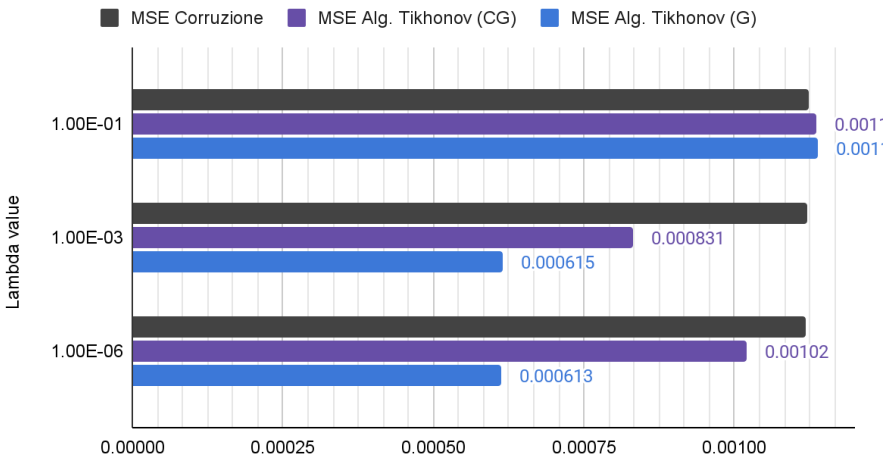


Immagine astronomia (2)

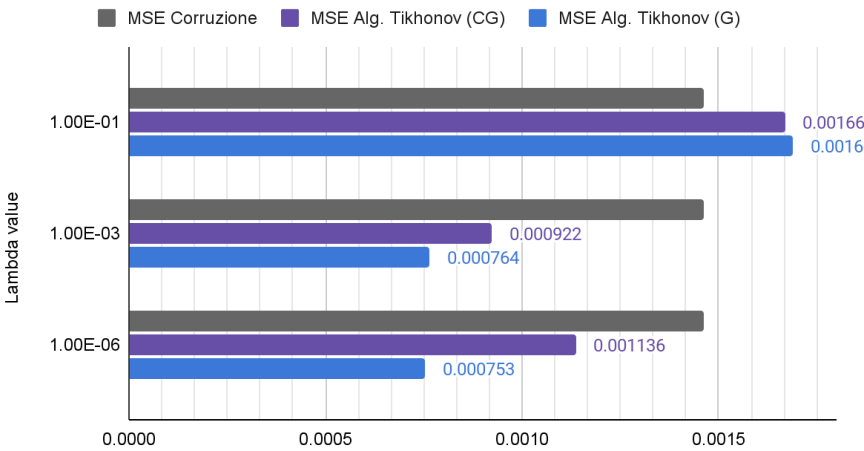
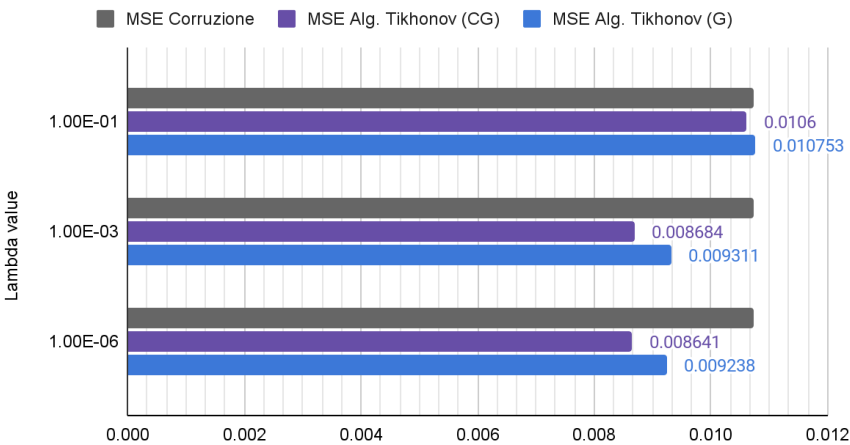


Immagine biologia (3)



# Analisi medie e deviazioni standard di PSNR ed MSE

10 ITER, KERNEL= 9, $\sigma= 1,3$ , DEVIAZIONE= 0,01, TIKHONOV=10e-7			
PSNR	ALGORITMO NAIVE	ALGORITMO CON R. DI TIKHONOV (CG)	ALGORITMO CON R. DI TIKHONOV (G)
MEDIA	26.894658	26.895006	35.01832
DEVIAZIONE STANDARD	0.114636	0.114625	0.396381
MSE	ALGORITMO NAIVE	ALGORITMO CON R. DI TIKHONOV (CG)	ALGORITMO CON R. DI TIKHONOV (G)
MEDIA	0.002045	0.002044	0.000316
DEVIAZIONE STANDARD	5.459490e-05	5.458539e-05	2.964536e-05

10 ITER, KERNEL= 9, $\sigma= 1,3$ , DEVIAZIONE= 0,04, TIKHONOV=10e-1			
PSNR	ALGORITMO NAIVE	ALGORITMO CON R. DI TIKHONOV (CG)	ALGORITMO CON R. DI TIKHONOV (G)
MEDIA	16.862782	15.354841	15.354841
DEVIAZIONE STANDARD	0.2951475	2.122798	2.122798
MSE	ALGORITMO NAIVE	ALGORITMO CON R. DI TIKHONOV (CG)	ALGORITMO CON R. DI TIKHONOV (G)
MEDIA	0.020639	0.032725	0.032725
DEVIAZIONE STANDARD	0.001352	0.015276	0.015277

Successivamente abbiamo calcolato sull'intero set di immagini da noi generate la media e la deviazione standard per le misurazioni del PSNR e dell'MSE.

Il PSNR è una misura che valuta il rapporto segnale/rumore, più questo valore è alto e meno l'immagine è rumorosa, ovvero c'è più segnale.

L' MSE invece è calcolato elevando al quadrato la differenza media tra i valori dell'immagine esatta e la ricostruzione di quest'ultima.

Per poter effettuare questi calcoli sono state utilizzate due funzioni di numpy, nello specifico *numpy.mean* per il calcolo della media e *numpy.std* per il calcolo della deviazione standard.

I dati sono stati raccolti utilizzando parametri differenti in modo tale da poter evidenziare maggiormente le diversità tra le medie e la deviazione standard.

## PSNR

Confrontando i dati ottenuti si nota che, con il valore di generazione del rumore molto basso, pari a 0,01, e il valore di  $\lambda$  a  $10^{-7}$ , otteniamo delle medie di PSNR vicine al valore originale dell'immagine senza corruzioni, mentre aumentando questi due parametri fino ad avere un rumore a 0,04 e  $\lambda$  a  $10^{-1}$ , le medie nella seconda tabella hanno valori simili ma differiscono molto dalle misurazioni precedenti, poiché aumentando il rumore abbiamo ottenuto delle immagini molto corrotte che, tramite il deblurring, non riescono ad avvicinarsi al valore PSNR dell'immagine corrotta inizialmente.

La deviazione standard indica la dispersione di una distribuzione di valori, più i punti sono lontani dalla media, maggiore sarà la deviazione e viceversa.

Dai dati riportati si può notare come l'aggiunta di rumore renda il tutto più instabile e infatti nei dati riguardanti il PSNR con una maggiore deviazione, abbiamo uno scarto quadratico medio molto maggiore rispetto a quando si utilizza una deviazione bassa.

Gli algoritmi dunque non vengono influenzati dalle immagini sulle quali lavorano ma sono resi più instabili all'aumento del rumore.

## MSE

Confrontando invece la media dei valori dell' MSE, notiamo che, come visto in precedenza, il rumore rende meno efficienti gli algoritmi, infatti mediamente l'errore aumenta quando aumentiamo l'intensità del rumore, ottenendo quindi delle immagini ricostruite meno precise rispetto a quando viene utilizzato un rumore nella media.

# Proprietà dei metodi del gradiente

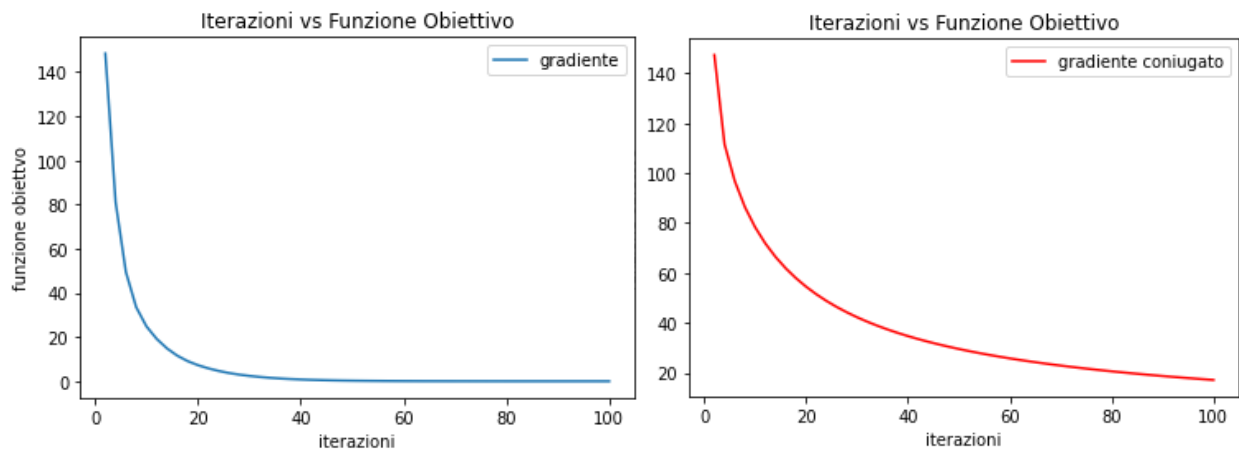
Nel quarto ed ultimo punto viene richiesto di analizzare le proprietà dei metodi del gradiente e del gradiente coniugato su 2 differenti esecuzioni.

## Prima esecuzione

Per la prima esecuzione abbiamo utilizzato i seguenti parametri:

- Dimensione kernel: 5x5
- Sigma: 0.5
- Deviazione : 0.04
- Lambda:  $10^{-3}$
- Iterazioni: 100

Generato il grafico che mette in relazione il numero di iterazioni dei due metodi con la Funzione Obiettivo, notiamo che quest'ultima diminuisce ad ogni iterazione (non per forza deve tendere a 0) per quanto riguarda il metodo del gradiente coniugato.

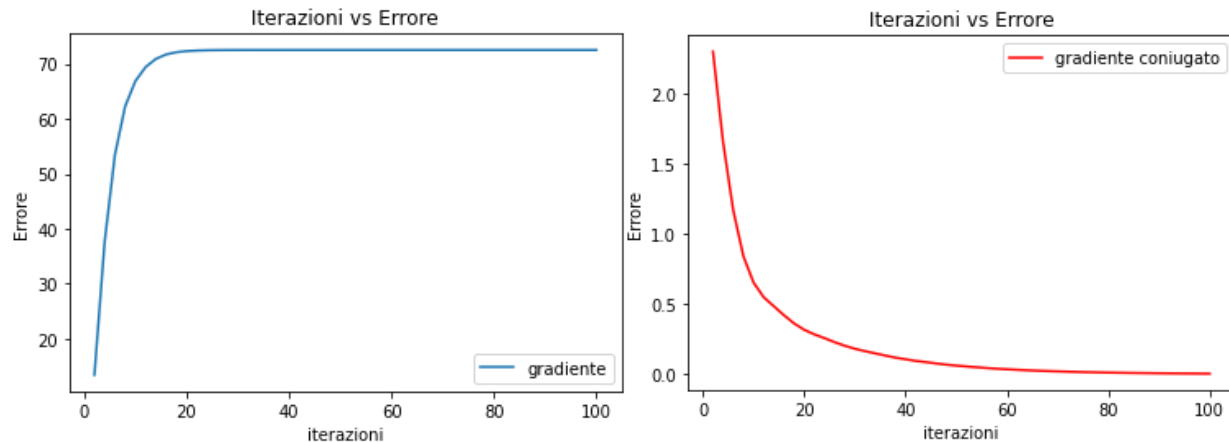


Il teorema seguente dà indicazioni su quanto velocemente decresce la funzione obiettivo, in particolare dice che il valore della decrescita della f.o. al passo  $k+1$  è minore uguale della decrescita della f.o. al passo  $k$ .

$$f(x_{k+1}) - f(x^*) \leq \left( \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} \right)^2 [f(x_k) - f(x^*)].$$

Questo teorema dice inoltre che la velocità con cui la f.o. decresce è proporzionale al quadrato della differenza del più grande meno il più piccolo fratto il più grande sommato al più piccolo degli autovalori della matrice Hessiana.

In seguito abbiamo realizzato il grafico che mette in relazione l'errore rispetto al numero di iterazioni.

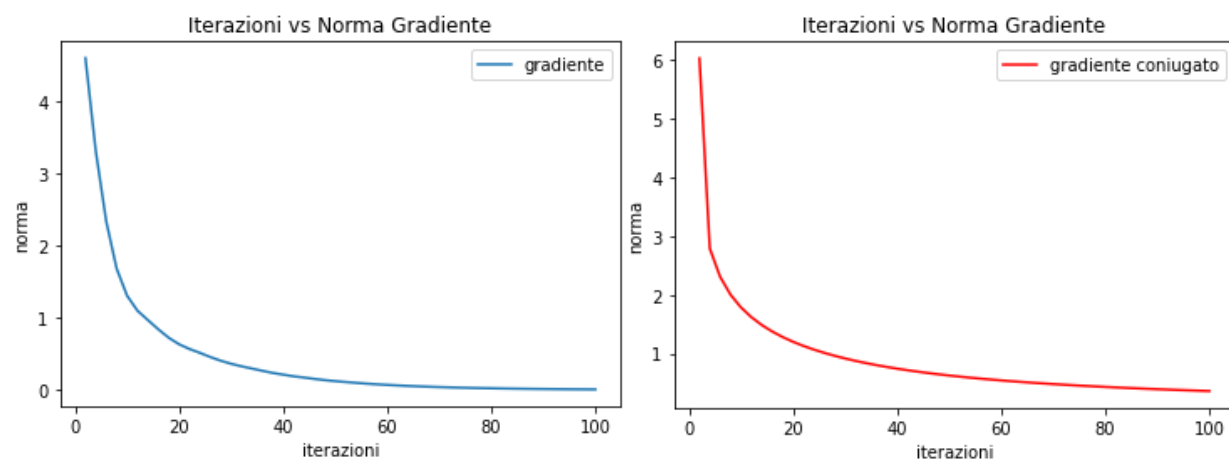


Conoscendo l'immagine di partenza che corrisponde alla soluzione esatta abbiamo potuto calcolare l'errore relativo tra questa e l'immagine ricostruita, questo grafico ci dà indicazioni sulla velocità di convergenza, nel senso di quanto velocemente decresce l'errore (per il metodo del gradiente coniugato) che deve comunque tendere a 0.

Bisogna anche osservare che per il gradiente coniugato l'andamento della curva dell'errore è simile a quello della f.o. vista in precedenza.

Un'altra osservazione che possiamo fare sta nel fatto che il metodo del gradiente essendo un metodo di discesa, ha una discesa monotona dell'errore come si vede anche dal grafico.

Per quanto riguarda il metodo del gradiente invece, l'errore tende a crescere all'aumentare delle iterazioni, infatti come già visto in precedenza all'aumentare delle iterazioni il PSNR tende a diminuire.



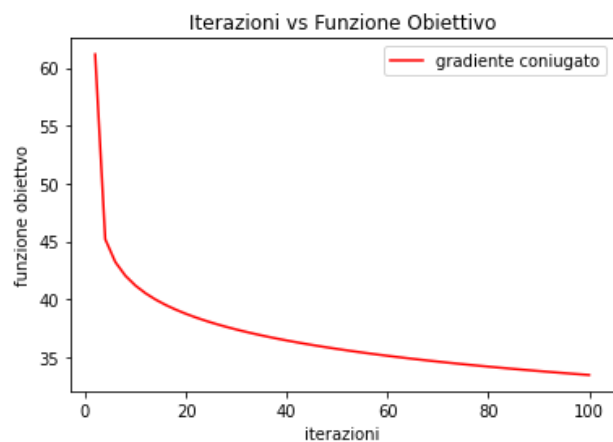
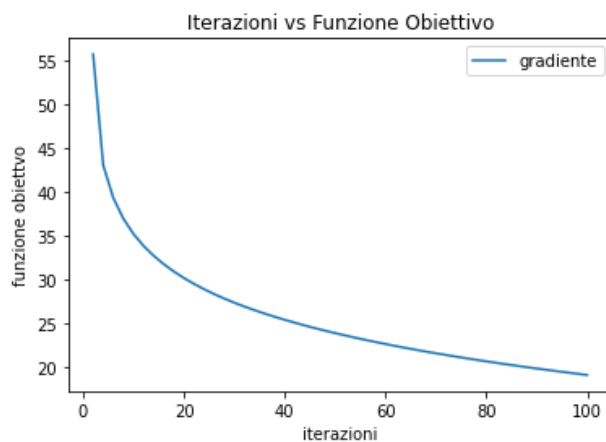
Alla base del metodo vi è stata l'idea di aggiungere un vincolo sulla norma della soluzione del problema, trasformandolo così in un problema di ottimizzazione vincolata.

Osservando il grafico della norma invece, notiamo che questa tende a diminuire ad ogni iterata, il metodo viene interpretato come un metodo iterativo che agisce sulla norma del residuo, questo si ferma nel momento in cui la norma è abbastanza piccola, ovvero quando nell'iterazione corrente ci troviamo sufficientemente vicini alla soluzione.

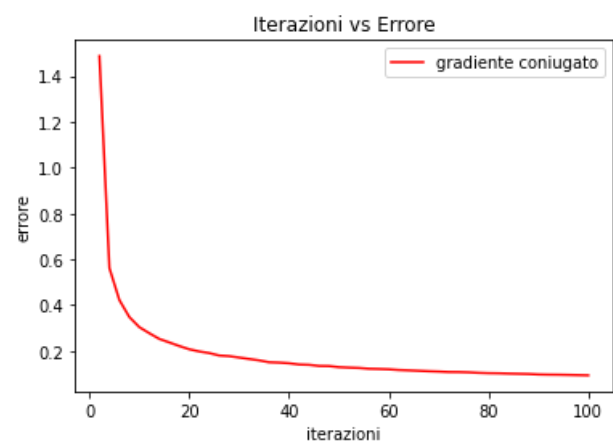
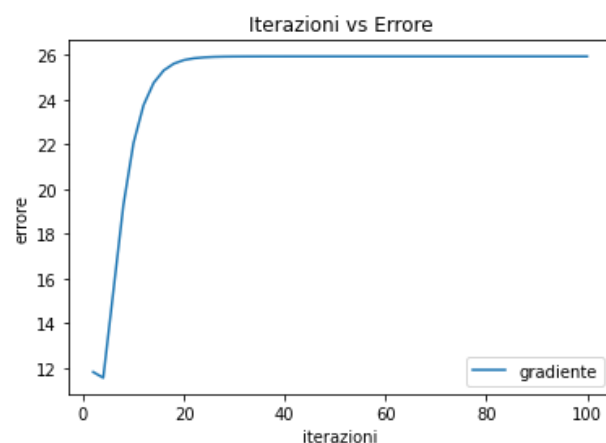
## Seconda esecuzione

Per la seconda esecuzione abbiamo utilizzato i seguenti parametri:

- Dimensione kernel: 9x9
- Sigma: 1.3
- Deviazione : 0.02
- Lambda:  $10^{-3}$
- Iterazioni: 100

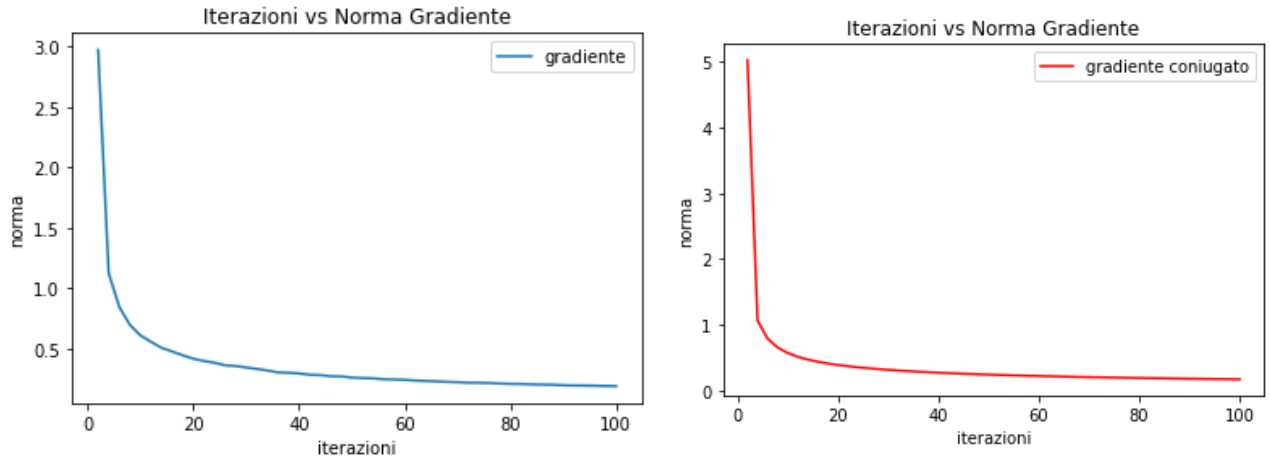


Come nell'esecuzione precedente entrambi i grafici hanno un andamento simile, e i valori delle f.o. tendono a decrescere ad ogni iterata.





Possiamo osservare che l'errore è decisamente inferiore rispetto all'esecuzione precedente, questo è dovuto alla riduzione della deviazione standard (da 0.04 a 0.02), quindi applicando meno rumore la ricostruzione dell'immagine ne risente in maniera positiva fornendo dei risultati migliori.



Anche l'andamento della norma è influenzato dalla deviazione standard, infatti si può vedere che i valori sono leggermente inferiori rispetto a prima, quindi significa che ci avvicineremo prima alla soluzione e quindi i metodi convergono più velocemente.

# Conclusione

Durante l'approfondimento dei diversi metodi utilizzati, ovvero il metodo "naive", il metodo del gradiente e il metodo del gradiente coniugato, abbiamo studiato le loro esecuzioni utilizzando parametri differenti per analizzare la loro applicazione nei problemi di minimizzazione non lineari.

Complessivamente i tre metodi si sono rivelati efficaci con valori bassi di rumore, infatti le immagini, poco affette da disturbi, vengono restituite con una certa qualità, mentre se questo valore aumenta, otteniamo delle immagini non ottime ma che comunque vengono deblurate nei limiti degli algoritmi, dato che anche con un rumore minimo, è praticamente impossibile dopo aver corrotto l'immagine, ottenere nuovamente l'immagine di partenza.

Nello specifico abbiamo notato che l'algoritmo di Tikhonov (G) si comporta meglio degli altri algoritmi per quanto riguarda i dati ottenuti e confrontando le due misure di PSNR ed MSE si può vedere come al termine del processo, si ottengano dei risultati migliori rispetto agli altri due metodi, soprattutto se si utilizzano valori alti della deviazione standard utilizzata nella generazione del rumore.