

# Metodi Iterativi in Tomografia

## Algoritmi algebrici di ricostruzione

Esiste un approccio completamente differente dalla FBP che si basa sull'assunzione che la sezione del corpo in esame sia un array di incognite e che si possa scrivere un **sistema di equazioni algebriche (lineari)** per le incognite in funzione dei dati misurati (proiezioni).

Questo approccio diventa essenziale quando:

- la propagazione non è rettilinea
- l'angolo totale di vista è limitato
- le proiezioni non sono equispaziate
- la risposta del sistema non è spazio invariante

Ovvero quando il modello utilizzato per la FBP non è più valido

## Rappresentazione dell'oggetto e delle proiezioni

La sezione del corpo sotto esame può essere schematizzata come una funzione di due variabili  $f(x,y)$ . Sovrapponiamo una griglia quadrata a questa funzione del piano, che assume i valori di densità per la Tac o concentrazione di radiofarmaco negli esami funzionali. Per ogni elemento della griglia di dimensione  $\delta \times \delta$  (pixel) possiamo supporre che la funzione sia costante e denotiamo tale valore con  $f_j$ . Come vedremo in seguito ci farà comodo esprimere tali valori con un solo indice, ordiniamo, quindi i pixel in modo lessicografico da 1 a  $n^2$  se la griglia ha dimensione  $n \times n$

## Rappresentazione dell'oggetto e delle proiezioni

Analogamente, anche le proiezioni assumendo valori discreti, possono essere rappresentate con un vettore con un numero totale di elementi pari al numero di punti per proiezione, solitamente uguale a  $n$ , per il numero di proiezioni. Chiamiamo il vettore dei dati con  $g_i$  con  $i=1, \dots, M$  con  $M= n \times N_{\text{angl}}$ .

Quindi una proiezione non è identificata da una linea senza "spessore", ma piuttosto da una piccola striscia di larghezza  $\tau$  (solitamente  $\tau = \delta$ )

## Rappresentazione dell'oggetto e delle proiezioni

I valori del vettore proiezione possono essere associati all'oggetto con la seguente equazione:

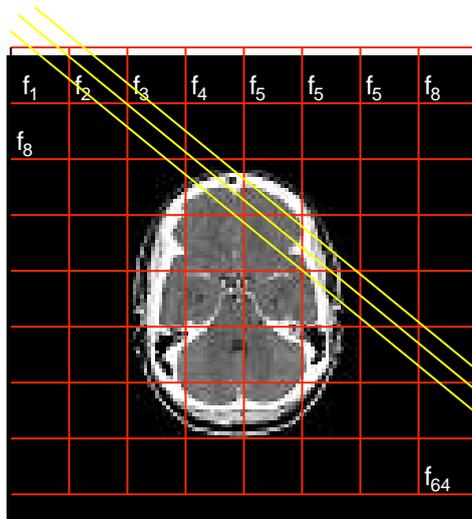
$$\sum_{j=1}^N a_{ij} f_j = g_i \quad \text{con } i = 1, \dots, M$$

dove  $a_{ij}$  è un fattore peso che rappresenta il contributo della  $j$ -esima cella all'  $i$ -esimo raggio.

I fattori peso possono essere scritti come l'area comune tra il raggio (striscia) e la cella.

Chiamiamo  $A$  la matrice di elementi  $a_{ij}$

## Rappresentazione dell'oggetto e delle proiezioni



$$f_j \quad \text{con } j = 1, N$$

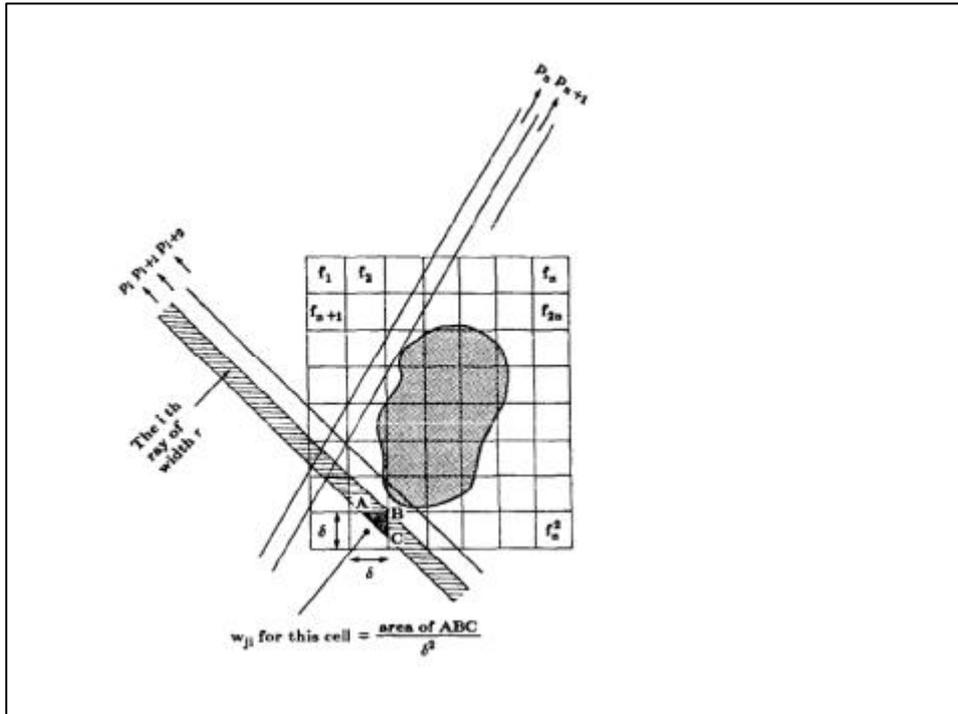
$$g_i \quad \text{con } i = 1, M$$

$$\sum_{j=1}^N a_{ij} f_j = g_i \quad i = 1, M$$

$$g_{i+2}$$

$$g_{i+1}$$

$$g_i$$



## Soluzione del sistema di equazioni

Analizziamo le caratteristiche di  $A$

- $A$  può non essere quadrata ( $M \times N$ ;  $M > N$ )
- $A$  è mal-condizionata
- $A$  è sparsa ma non ha, in generale, una struttura

Il sistema di equazioni appena scritto è in genere un sistema di equazioni sovradeterminato (più dati che incognite); se  $M$  e  $N$  fossero piccoli si potrebbe affrontare il problema con gli algoritmi standard, ma si possono avere facilmente 65.000 incognite (immagini  $256 \times 256$ ) con circa lo stesso numero di dati, ovvero **matrici 65.000 per 65.000**.

## Problema inverso lineare

$$\vec{g} = A\vec{f} + \vec{n}$$

Diagram illustrating the linear inverse problem equation  $\vec{g} = A\vec{f} + \vec{n}$  with annotations:

- $\vec{g}$ : dati affetti da rumore sinogramma (green)
- $A$ : Matrice che descrive il sistema (green)
- $\vec{f}$ : incognita "oggetto" (red)
- $\vec{n}$ : rumore (red)

Risolvere il **problema inverso** significa trovare una **stima dell'oggetto  $f$** , a partire dai dati  **$g$**  prodotti dall'apparato descritto dalla **matrice  $A$** .

## Soluzione del problema inverso

Un **problema inverso può essere "mal posto"** nel senso che

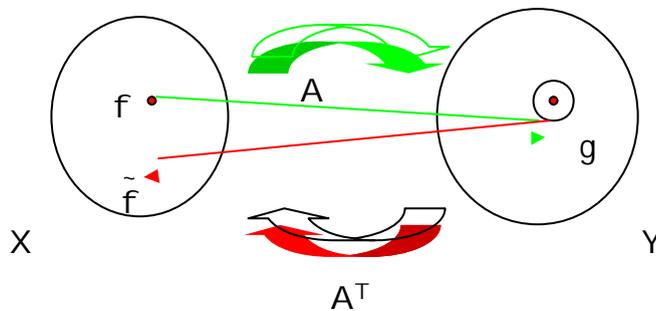
1. La soluzione può non esistere
2. La soluzione può non essere unica
3. Il rumore sui dati si può propagare amplificato alla soluzione, ovvero questa può essere numericamente instabile.

## Analisi del problema inverso

Prima di iniziare ad analizzare il problema inverso cerchiamo una "analogia" formale tra la matrice  $A$  e la trasformata di Radon, e definiamo due spazi vettoriali:  $X$  lo spazio della soluzioni e  $Y$  lo spazio dei dati.

$$\vec{g} = A\vec{f} \quad (Rf)(s, \mathbf{j}) = \int_{-\infty}^{+\infty} f(s\vec{J} + t\vec{J}^\perp) dt$$

$$\vec{\tilde{f}} = A^T \vec{g} \quad (R^\# g)(x, y) = \int_0^{2\pi} g(x \cos(\mathbf{j}) + y \sin(\mathbf{j}), \mathbf{j}) d\mathbf{j}$$



## Problema ai minimi quadrati

Un modo per ottenere l'esistenza della soluzione è quello di cercare di risolvere il seguente problema ai minimi quadrati:

Trovare la stima  $\tilde{f}$  che rende minima la quantità

$$\left\| A\tilde{f} - \bar{g} \right\|_Y^2 = \min$$

## Problema ai minimi quadrati

Per continuare l'analisi del problema dobbiamo verificare che la soluzione sia unica.

Se il **nucleo** (l'insieme degli oggetti che forniscono dati nulli) contiene solo l'elemento nullo, allora la soluzione è unica.

Se ciò accade, la matrice è detta a **rango pieno** (rango=dimensione del co-dominio)

## Unicità della soluzione

Per giustificare che la soluzione è unica ritorniamo al Fourier Slice Theorem.

Vogliamo dimostrare che l'equazione

$$R(\vec{f}) = \vec{0}$$

con  $R$  trasformata di Radon, ha solo la soluzione

$$\vec{f} = \vec{0}$$

## Unicità della soluzione

Il Fourier slice theorem dice che

$$\hat{g}_J(\mathbf{w}) = \hat{f}(\mathbf{w}\mathbf{J})$$

Se  $R(\vec{f}) = \vec{0}$



allora tutte le proiezioni sono nulle



tutte le trasformate di Fourier delle proiezioni sono nulle (per tutti gli angoli)



la trasformata di Fourier di  $f$  è nulla e quindi  $f$  è nulla

## Singular Value Decomposition

La matrice  $A$  può essere analizzata con la SVD (Singular Value Decomposition).

L'idea base è scomporre la matrice  $A(M \times N)$

$$A = U\Sigma V^T$$

dove  $\Sigma$  è una matrice diagonale ( $N \times N$ ) e  $U(M \times N)$  e  $V(N \times N)$  sono matrici isometriche (cioè  $U^T U = I$  e  $V^T V = I$ ).

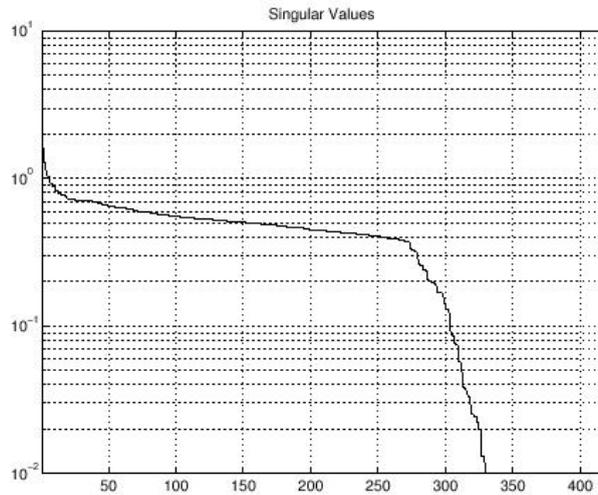
I valori sulla diagonale della matrice  $\Sigma$  sono detti valori singolari e sono normalmente ordinati in ordine non crescente:  $\mathbf{s}_1 \geq \mathbf{s}_2 \geq \dots \geq \mathbf{s}_N$

## Singular Value Decomposition

La SVD oltre a permetterci di trovare, come vedremo in seguito, una soluzione, ci permette di valutare il grado di mal-condizionamento del problema inverso. Infatti il numero

$$\mathit{cond}(A) = \frac{\mathbf{s}_1}{\mathbf{s}_N}$$

ci fornisce una valutazione della propagazione del rumore dai dati alla soluzione. In altre parole, il rumore dei dati viene **amplificato** al più di un fattore  **$\mathit{cond}(A)$** .



**Figure 9.5** The singular values of the system matrix  $A$ .

$$\text{cond}(A) = 200$$

## Singular Value Decomposition

Si dimostra che una soluzione del problema ai minimi quadrati è data da

$$\vec{f}^+ = V \Sigma^{-1} U^T \vec{g}$$

$$A^+ = V \Sigma^{-1} U^T \text{ e } \Sigma^{-1} = \begin{pmatrix} \frac{1}{s_1} & 0 & 0 & 0 \\ 0 & \frac{1}{s_2} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \frac{1}{s_N} \end{pmatrix}$$

inversa generalizzata

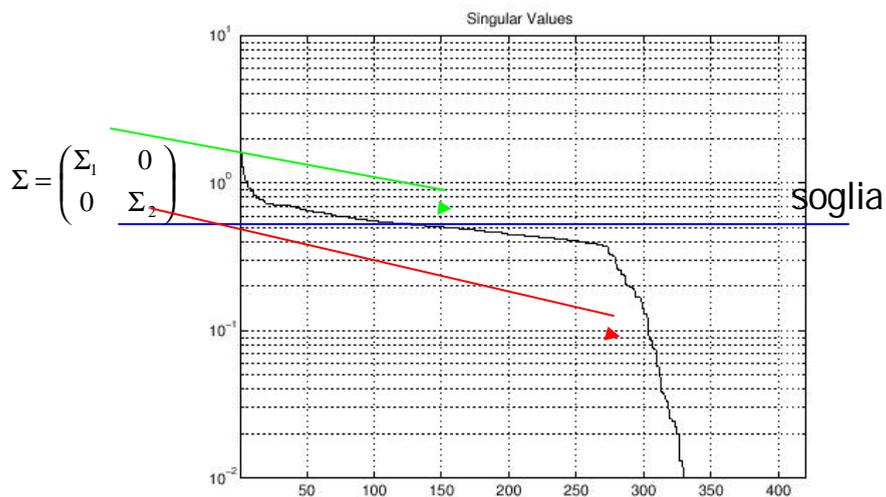
questa soluzione è chiamata **soluzione generalizzata** e, in genere, è numericamente instabile.

## Singular Value Decomposition

Un modo di limitare l'instabilità della soluzione è quello di mettere a zero il contributo dei valori singolari più piccoli, in un certo senso limitare i gradi di libertà della soluzione. Consideriamo la matrice dei valori singolari e la scomponiamo in due parti

$$\Sigma = \begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix}$$

In pratica si divide lo spazio della soluzioni in due: un sottospazio corrispondente a segnale ed uno corrispondente al rumore



**Figure 9.5** The singular values of the system matrix  $A$ .

$$\text{cond}(A) = 200$$

## Singular Value Decomposition

In funzione quindi del valore della soglia possiamo trovare delle soluzioni approssimate ma stabili sostituendo a  $\Sigma^{-1}$

$$\Sigma_{tr}^{-1} = \begin{pmatrix} \Sigma_1^{-1} & 0 \\ 0 & 0 \end{pmatrix}$$
$$\vec{f}_{tr} = V \Sigma_{tr}^+ U^T \vec{g}$$

## Singular Value Decomposition

Esistono anche altri metodi che implementano un filtro dei valori singolari, meno drastico del troncamento, per esempio il metodo di Tikhonov.

Tutti questi metodi sono detti "regolarizzanti" e forniscono una famiglia di soluzioni dipendente da un parametro: soglia oppure parametro di regolarizzazione che disegna il filtro sui valori singolari.

PROBLEMI DELLA SVD: complessità di calcolo  $O(N^3)$  matrici 7000x4000 35 ore calcolo su SUN

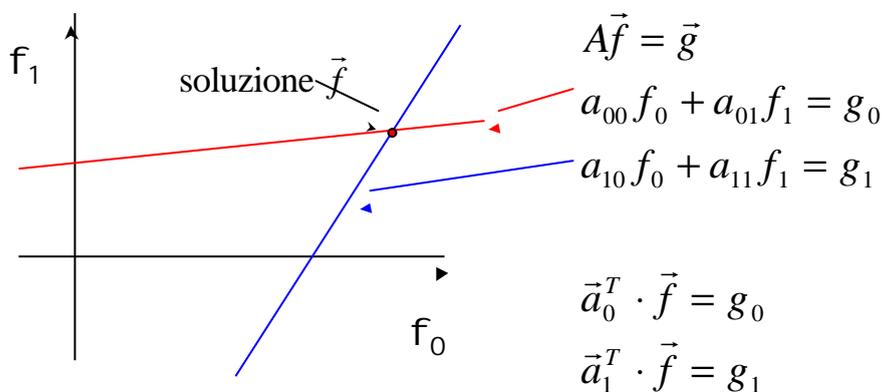
## Metodi Iterativi

L'alternativa alla SVD è un approccio iterativo. Infatti un modo per curare il mal-condizionamento del problema è scrivere algoritmi iterativi che, quando vengono fermati, hanno un effetto regolarizzante sulla soluzione.

Storicamente il primo algoritmo è stato ART utilizzato da Cormack e Hounsfield per le prime immagini tomografiche nel 1970. Più tardi fu scoperto che ART è una rivisitazione di un algoritmo scritto da Kaczmarz nel 1937.

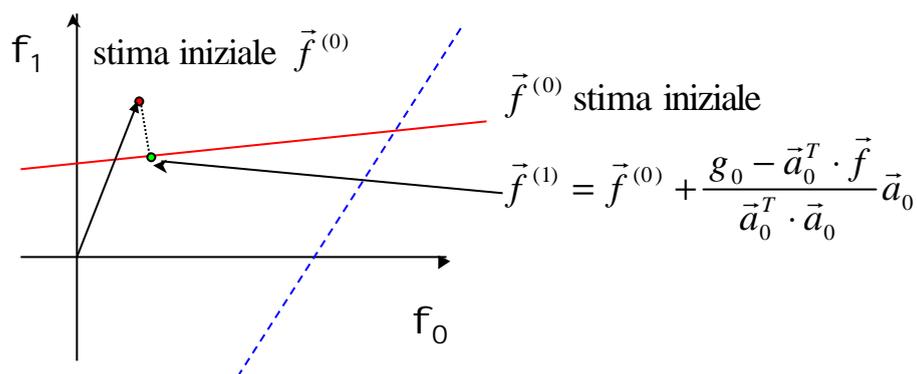
## ART

Per spiegare l'algoritmo ci limitiamo al caso 2 dati e 2 incognite in modo da poter visualizzare il processo in uno spazio a due dimensioni



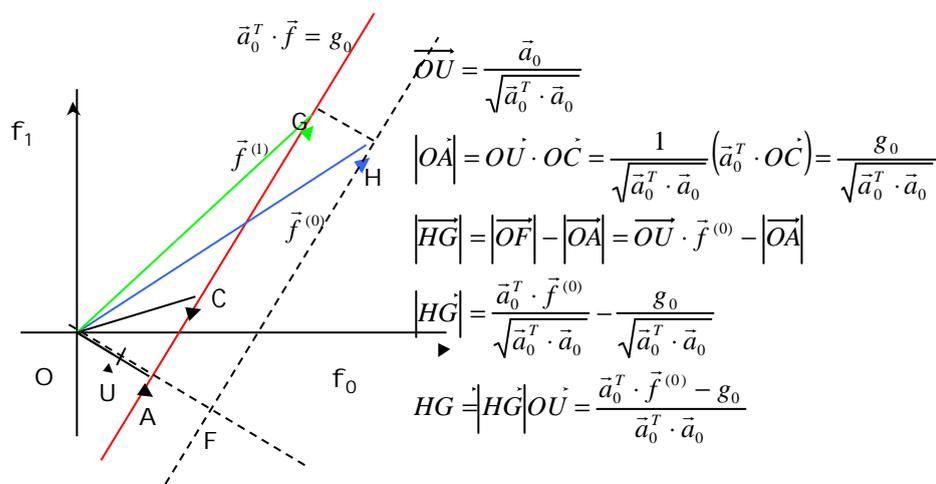
# ART

L'algoritmo di ricostruzione algebrica iterativa (Algebraic Reconstruction Iterative) utilizza una riga della matrice per volta



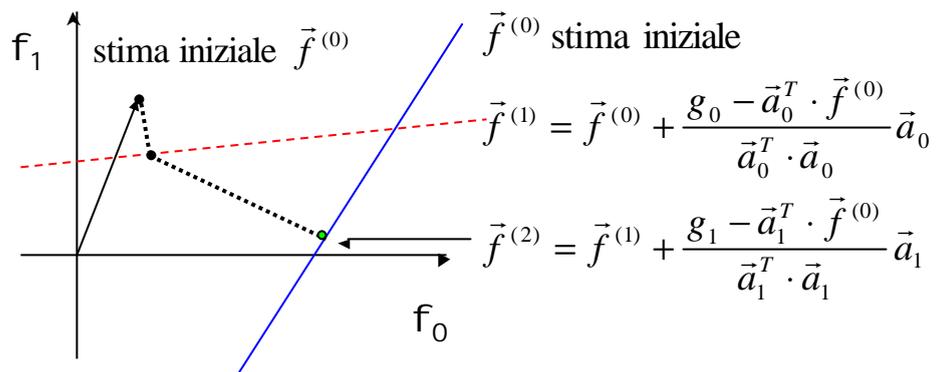
# ART

Interpretiamo il singolo passo di ART



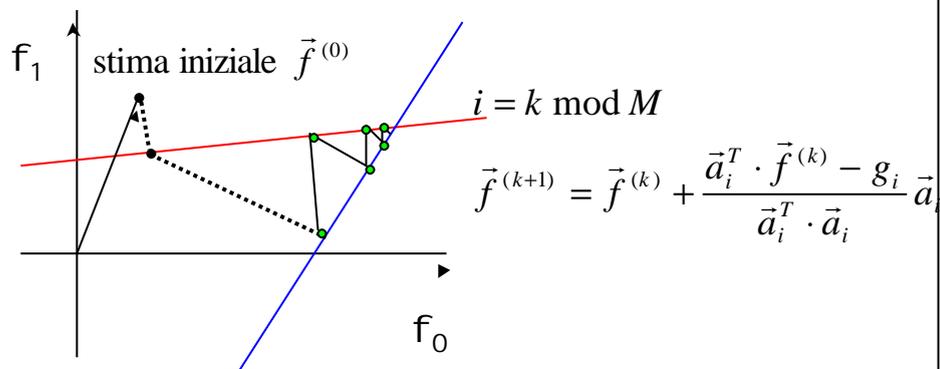
# ART

Ad ogni iterazione si proietta su una retta (iperpiano)



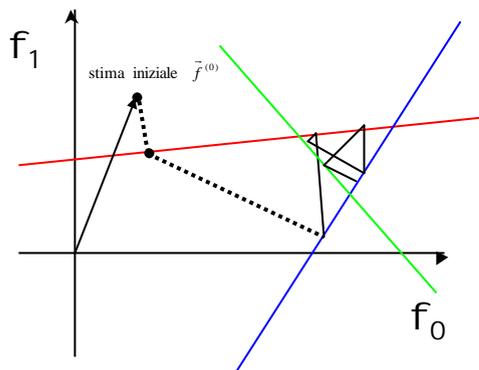
# ART

Fino a che non si raggiunge la convergenza



# ART

Se il problema è sovradeterminato (più dati che incognite) e i dati sono afflitti da noise si può creare la seguente situazione



$$i = k \bmod M$$

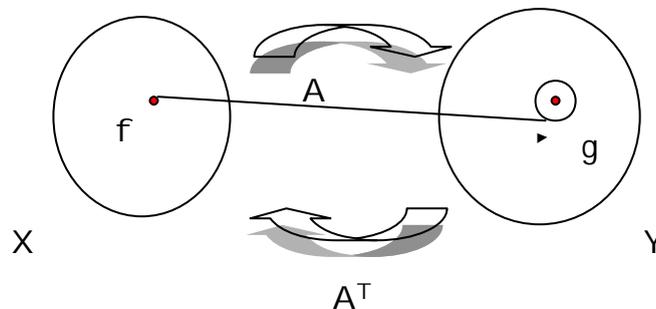
$$\vec{f}^{(k+1)} = \vec{f}^{(k)} + \frac{\vec{a}_i^T \cdot \vec{f}^{(k)} - g_i}{\vec{a}_i^T \cdot \vec{a}_i} \vec{a}_i$$

Riformuliamo il problema nel senso dei minimi quadrati

funzionale discrepanza  $\Rightarrow \mathbf{e}^2(f; g) = \|\mathbf{A}f - g\|_y^2 = \min$

gradiente  $\Rightarrow \nabla_f \mathbf{e}^2(f) = 2\mathbf{A}^T (\mathbf{A}f - g)$

soluzione iterativa  $\Rightarrow f^{(k+1)} = f^{(k)} + t\mathbf{A}^T (g - \mathbf{A}f^{(k)})$



### Proprietà di semiconvergenza

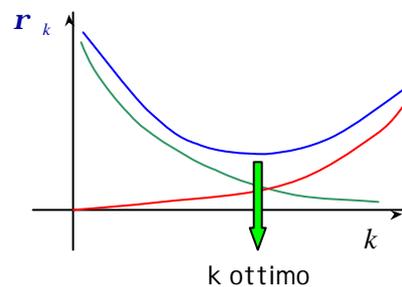
$$g = Af + n = g^{(0)} + n$$

In funzione delle iterazioni si può calcolare l'errore di ricostruzione

$$r_k = \frac{\|f^{(k)} - f\|}{\|f\|}$$

Errore di approssimazione

Errore propagazione del rumore



Esiste un valore ottimale del numero di iterazioni che minimizza l'errore di ricostruzione

Occorre valutare l'iterata ottimale nel caso di dati reali



Training dell'algoritmo con simulazioni realistiche per la valutazione dell'iterata ottimale

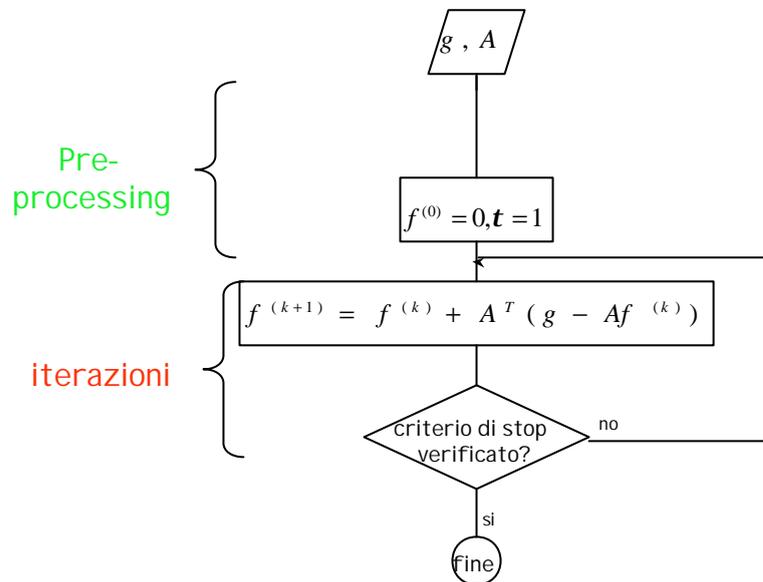
Se si posseggono informazioni "a priori" sul livello di rumore dei dati, si può fermare l'algoritmo quando la discrepanza raggiunge una soglia prestabilita



Dipendenza del numero di iterazioni ottimali dal tipo di oggetto (e dal noise)

Sottostima del numero di iterazioni, costo computazionale più alto

## Simultaneous Iterative Reconstruction Technique



Il metodo SIRT può essere facilmente modificato per risolvere problemi ai minimi quadrati **vincolati**.



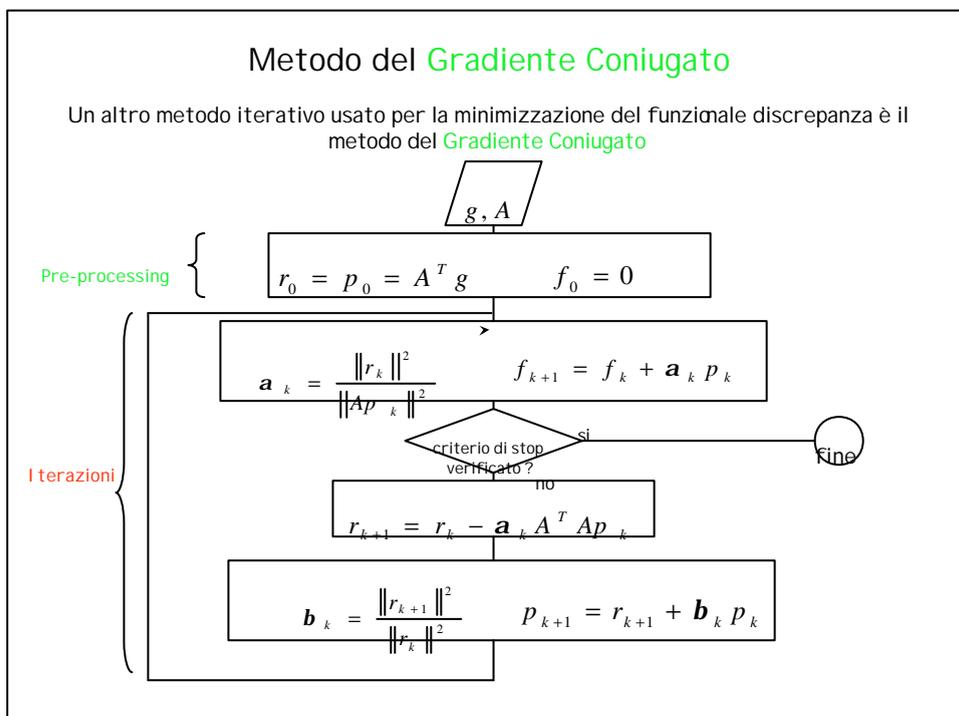
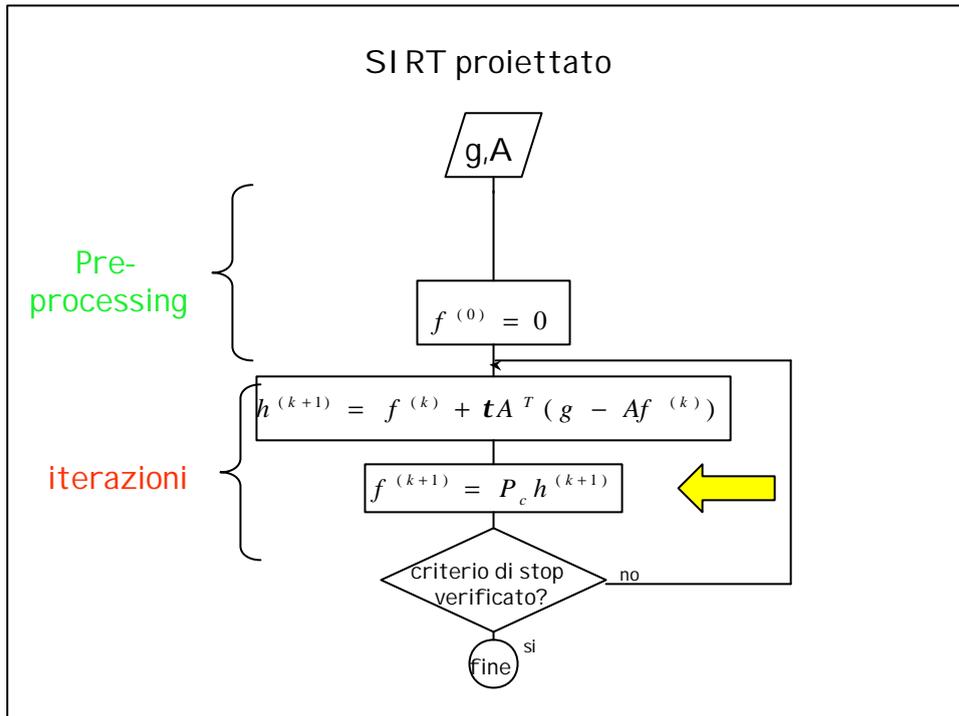
### SIRT con vincoli

Sia  $C \subset X$  un insieme chiuso e convesso, (soluzioni non negative), e sia  $P_c$  il proiettore su questo insieme, allora l'algoritmo è così modificato:

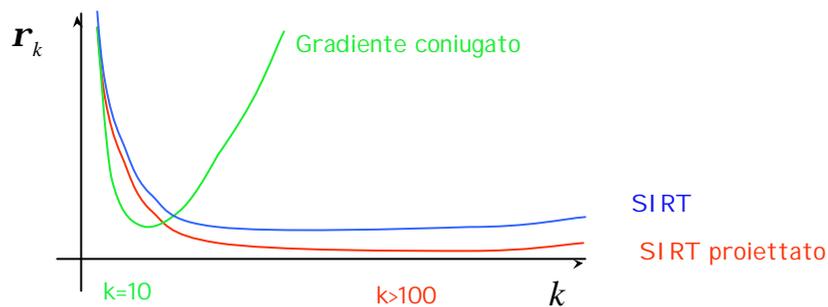
$$f_c^{(k)} = P_c \left\{ f_c^{(k)} + tA^T (g - Af_c^{(k)}) \right\}$$

convergenza molto lenta (dipendente dal "tipo" dell'oggetto)

minimo dell'errore di ricostruzione molto piatto



- Si dimostra che il **gradiente coniugato** presenta la proprietà di semiconvergenza; la convergenza al valore ottimale di iterazioni è molto **più rapida** che nel caso di SIRT ma la scelta è **più critica**.
- **Non** è possibile inserire facilmente **vincoli** sulla soluzione
- **E' possibile** usare precodizionamento per accelerare la convergenza



## Metodi di massima verosimiglianza

I metodi fino ora trattati possono essere considerati metodi **deterministici** perché non tengono in considerazione la natura **probabilistica del rumore** che perturba i dati.

I metodi di **massima verosimiglianza** considerano **l'oggetto** (ovvero l'incognita) **deterministico**, mentre le componenti dell'**immagine** e del **rumore** realizzazioni di **variabili aleatorie** e quindi il problema da risolvere si riscrive:

$$g = Af + n \quad \longrightarrow \quad \underset{\substack{\downarrow \\ \text{variabili aleatorie}}}{h} = Af + \underset{\substack{\downarrow \\ \text{variabili aleatorie}}}{n}$$

Se il **rumore** che affligge i dati è **Gaussiano bianco** il metodo della massima verosimiglianza porta al caso dei **minimi quadrati**. Più interessante è il caso in cui il rumore, e di conseguenza anche i dati, abbiano statistica **Poissoniana**. Questo si verifica in tutti quei casi in cui si ha una misura di **intensità luminosa** (astronomia) o **conteggi fotonici** (tomografia a emissione).

In questo caso **g, A e f** assumo valori **non-negativi**.

Sia  $g$  un valore osservato della variabile aleatoria  $h$  (una sua misura), allora si definisce **funzione di verosimiglianza** di un oggetto  $f$  la probabilità della variabile  $h$  di assumere il valore corrispondente a  $g$ , dato  $f$ .

$$L(f) = p_h(g | f)$$

L'oggetto  $\tilde{f}$  che **massimizza** la funzione  $L(f)$  è una stima dell'oggetto incognito  $f$ . Quindi il problema è quello di trovare il **massimo della funzione di verosimiglianza**.

Per determinare la posizione del massimo (o dei massimi) occorre calcolare gli zeri delle **derivate prime di  $L(f)$  rispetto alle componenti di  $f$** .

Le  $N$  componenti di  $h$  sono variabili aleatorie indipendenti e quindi la funzione densità di probabilità è il **prodotto** di  $N$  funzioni.

Utilizzando le proprietà del **logaritmo** si semplifica il calcolo delle derivate, e quindi si cerca il massimo della funzione **log-likelihood**

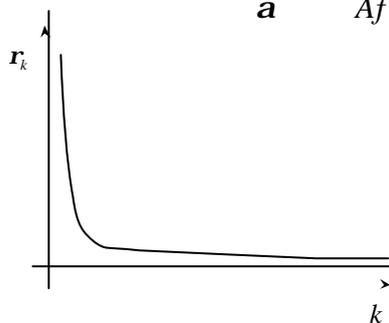
$$l(f) = \ln L(f) = \ln p_h(g | f)$$

Il problema della **ricerca del massimo** della funzione di log-likelihood è tuttavia un **problema mal-posto** e occorre **regolarizzare**.

Il metodo di Lucy-Richardson o **Expectation-Maximization** è un metodo iterativo che converge ( $k \rightarrow \infty$ ) al **massimo della funzione di log-likelihood**

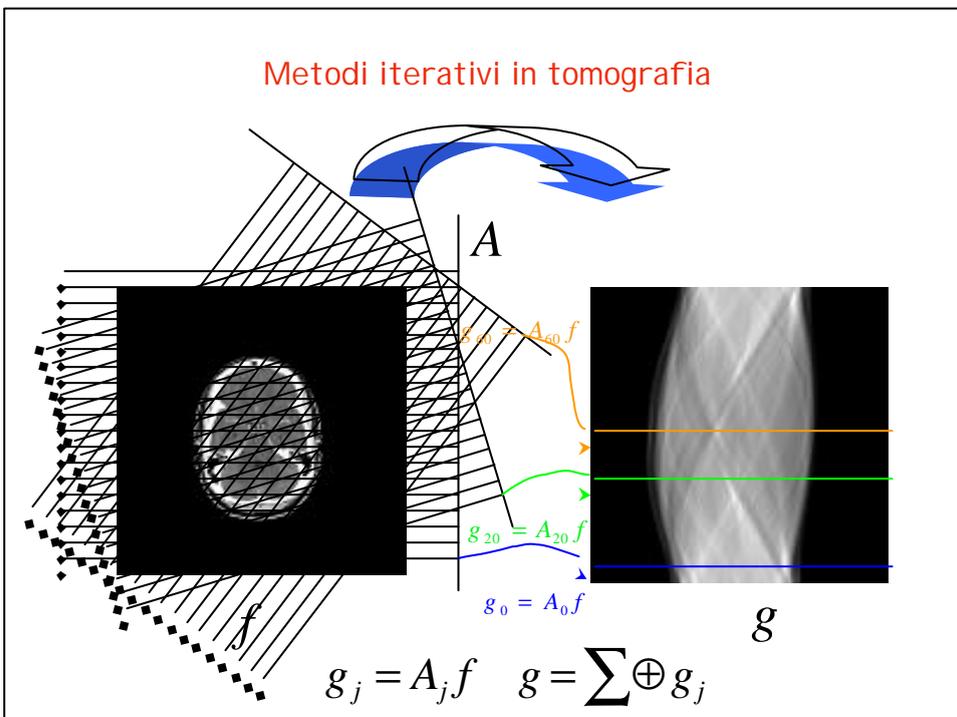
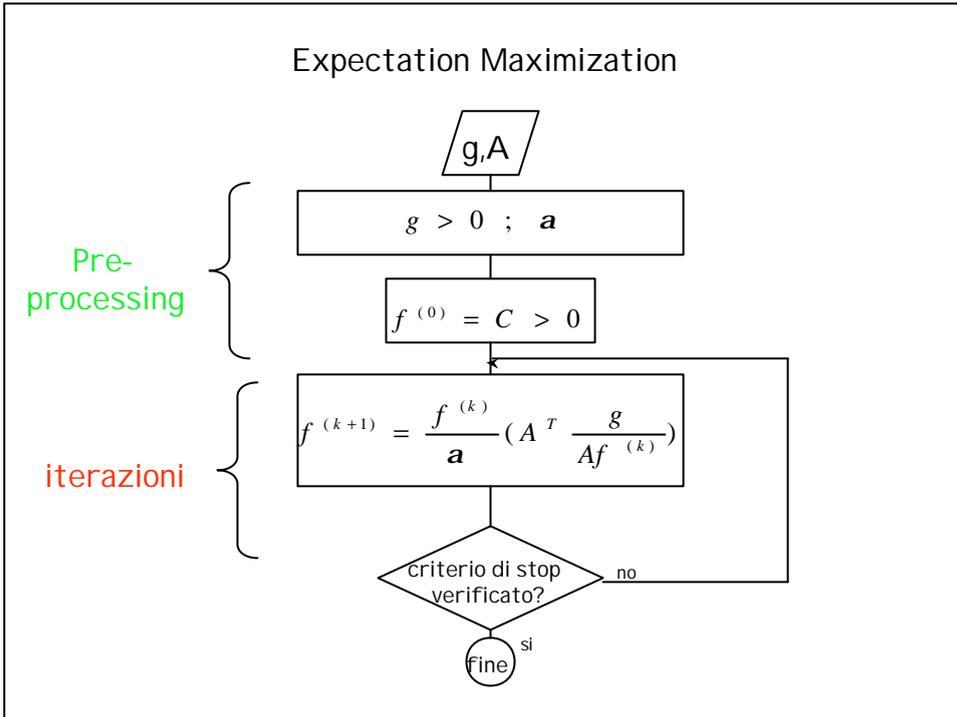
L'effetto **regolarizzante** si ottiene **fermando le iterazioni** come avviene per gli altri metodi iterativi.

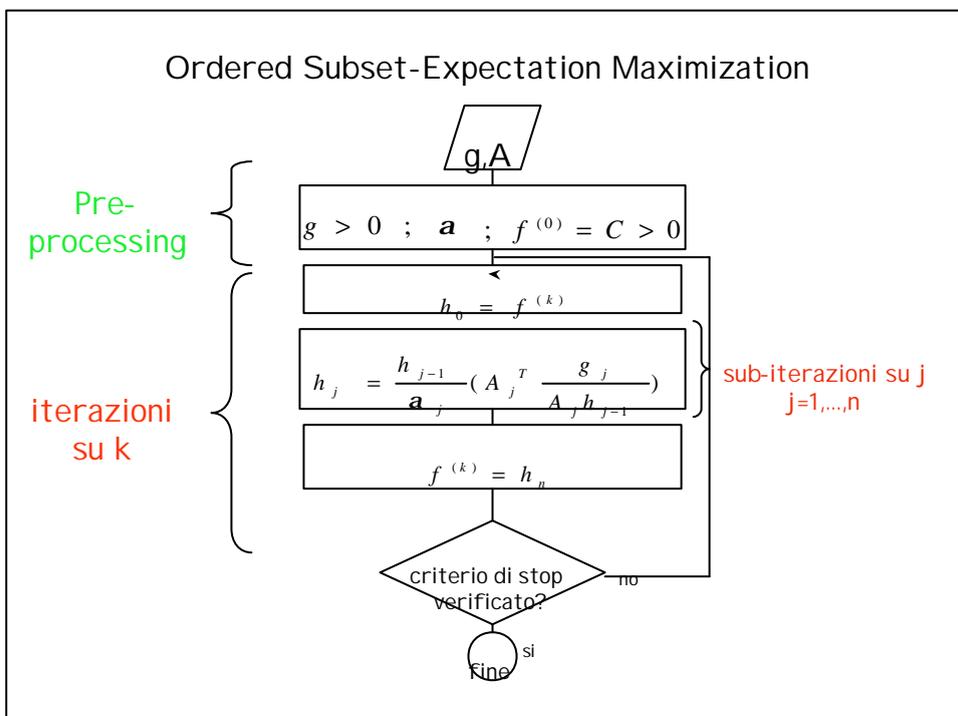
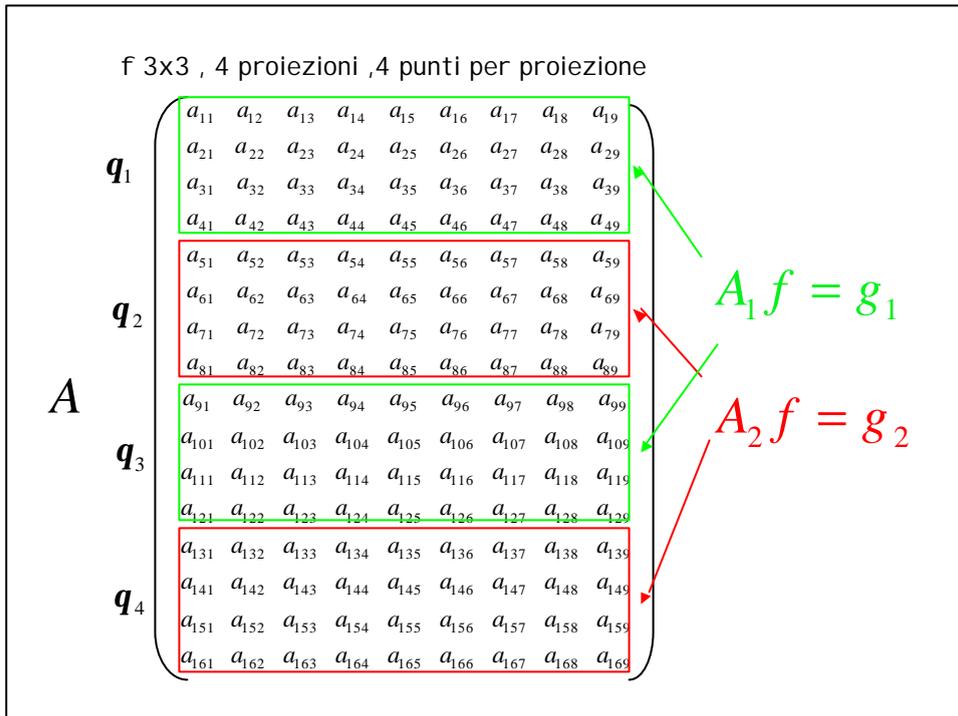
$$f^{(k+1)} = \frac{f^{(k)}}{\mathbf{a}} \left( A^T \frac{\mathbf{g}}{A f^{(k)}} \right) ; f^{(0)} = \mathbf{C} ; \mathbf{a}_n = \sum_m A_{mn}$$



- Ogni iterata è non negativa
- se  $\mathbf{a} = 1$   $\sum_n f_n^{(k)} = \sum_m g_m \quad \forall k$
- Se per  $k = \bar{k}$ ,  $f_n^{(\bar{k})} = 0$  allora  
 $f_n^{(k)} = 0$  per  $k > \bar{k}$

È possibile imporre il supporto alla soluzione





Sono stati presentati **solo alcuni esempi** di metodi iterativi utilizzati per la risoluzione di problemi inversi lineari

- I metodi iterativi sono utili per la risoluzione dei problemi inversi lineari perché hanno un effetto regolarizzante sulla soluzione.
- E' possibile imporre vincoli (supporto, non negatività) sulla soluzione.
- Occorre studiare per ogni tipo di problema il criterio di stop più opportuno.