

ALMA MATER STUDIORUM · UNIVERSITÀ DI
BOLOGNA

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI
Corso di Laurea Specialistica in Matematica

**METODI NUMERICI
PER LA RISOLUZIONE
DI PROBLEMI
DI BLIND DECONVOLUTION
IN IMMAGINI ASTRONOMICHE**

Tesi di Laurea in Analisi Numerica

Relatore:
Chiar.ma Prof.ssa
Loli Piccolomini Elena

Presentata da:
Benfenati Alessandro

III Sessione
Anno Accademico 2009/2010

Indice

Introduzione	iv
1 Formazione delle Immagini Digitali	1
1.1 Acquisizione delle Immagini	1
1.2 Modello Matematico Continuo	2
1.2.1 Blurring	2
1.2.2 Noise	5
1.3 Modello Matematico Discreto	7
1.3.1 Blurring	7
1.3.2 Noise	13
1.4 Image Deblurring	15
2 Metodi di Regolarizzazione	17
2.1 Pseudo Inversa di Moore-Penrose	17
2.2 Teoria spettrale	19
2.3 Operatori Compatti e Sistemi Singolari	23
2.4 Operatori di Regolarizzazione	25
2.5 Regolarizzazione nel Discreto	27
2.5.1 SVD di una Matrice: metodo di Tikhonov	28
2.5.2 Total Variation	30
2.6 Scelta del Parametro	33
3 Blind Deconvolution	35
3.1 Metodi di Ottimizzazione	35
3.1.1 Discesa Ripida	36
3.1.2 Gradiente Coniugato	37
3.1.3 Metodo di Newton	39
3.1.4 Metodo di Newton Proiettato	40
3.2 Problema di Blind Deconvolution	45
4 Risultati Numerici	47
4.1 Dati iniziali	47
4.2 FPQN	48
4.2.1 Rumore Gaussiano	48
4.2.2 Rumore Poissoniano	50
4.3 Blind Deconvolution	51
4.3.1 Rumore Gaussiano	52
4.3.2 Rumore poissoniano	55

4.4 Conclusioni	56
Bibliografia	59

Introduzione

Lo scopo di questo elaborato è lo studio di metodi per image deblurring: problemi di questo tipo sorgono in vari contesti applicativi, come astrofisica, astronomia, medical imaging, geofisica e inverse scattering.

Il modello discreto di formazione considerato è

$$\mathbf{b} = K(\mathbf{y})\mathbf{x} + \eta$$

dove $\mathbf{x} \in \mathbb{R}^n$ rappresenta l'immagine reale, $\eta \in \mathbb{R}^m$ è il rumore additivo sconosciuto, $K(\mathbf{y}) \in \mathcal{M}_{m \times n}(\mathbb{R})$ è una matrice che modella la perturbazione presente sull'immagine ed è dipendente da un vettore di parametri $y \in \mathbb{R}^p$ e \mathbf{b} è il dato a disposizione. Il fine degli algoritmi di deblurring è risalire ad un'approssimazione dell'immagine \mathbf{x} .

Nelle applicazioni è necessario avere informazioni sulla matrice $K(\mathbf{y})$ per applicare algoritmi di tale tipo. Normalmente si conosce solo la forma parametrica di K , senza conoscere il vettore \mathbf{y} ; si attuano allora algoritmi di blind deconvolution, si cerca cioè un'approssimazione dei veri parametri partendo da una stima iniziale per poter giungere ad una soluzione accettabile.

In questa tesi si svilupperà la teoria necessaria per spiegare gli algoritmi che stanno alla base dei procedimenti di image deblurring, per arrivare infine all'applicazione di metodi per la blind deconvolution. L'obiettivo finale sarà l'applicazione di metodi per image deblurring e blind deconvolution nel caso del rumore poissoniano, con l'aggiunta del vincolo di positività sulla soluzione. In letteratura è usuale trovare il problema esposto nel caso di rumore gaussiano e senza il vincolo di positività.

Nel primo capitolo verranno esposti i modelli di formazione di immagine, cominciando con lo spiegare la conversione da immagini analogiche a digitali per poi continuare con il modello matematico prima continuo poi discreto.

Nel secondo capitolo verranno poste le basi teoriche per poter affrontare gli algoritmi implementati. Si comincerà con la generalizzazione dell'inverso di un operatore tramite l'inversa di Moore-Penrose, per continuare con elementi di teoria spettrale; si continuerà l'esposizione con cenni di teoria di operatori compatti, operatori che emergono nei problemi applicativi. Si esporranno la teoria degli operatori di regolarizzazione, operatori che hanno il compito di tenere sotto controllo l'influenza del rumore sul risultato finale; a questo proposito verranno riportati gli esempi dei funzionali di Tikhonov e della Total Variation. Infine

verrà esposto l'esempio della GCV come funzione per la scelta del parametro di regolarizzazione.

Nel terzo capitolo si affronteranno i metodi di ottimizzazioni, necessari per risolvere i problemi di minimo presenti negli algoritmi per image deblurring. Verranno presentate successivamente le idee basilari per l'algoritmo di blind deconvolution.

Il quarto capitolo chiude l'elaborato presentato i risultati ottenuti dagli esperimenti. Verranno inizialmente mostrati i problemi test creati al calcolatore; verranno esposti i risultati di image deblurring avendo a disposizione i parametri esatti della matrice K . Infine si mostreranno i risultati ottenuti per l'algoritmo di blind deconvolution implementato.

Capitolo 1

Formazione delle Immagini Digitali

In questo capitolo introduttivo si espongono le basi per affrontare il problema di ricostruzione delle immagini digitali.

Nella prima sezione viene spiegato come l'immagine di un'oggetto viene acquisita. Per approfondire il discorso, si possono consultare [BP] e [BLZ].

Nella seconda sezione si introduce il modello matematico continuo, con l'equazione integrale di Fredholm di prima specie. Si considererà, in questo elaborato, il caso di operatori spazio invarianti, quindi il modello continuo sarà rappresentato da un prodotto di convoluzione.

Nella terza parte del capitolo viene spiegata la discretizzazione del modello continuo per poter creare problemi test al calcolatore. Nelle ultime due sezioni si è seguita l'esposizione in [CV] e [HNO]; per una discussione più approfondita sulla formazione del rumore poissoniano, si può consultare [BL].

Nella quarta e ultima sezione si mostra la malposizione del problema, tramite un esempio pratico.

1.1 Acquisizione delle Immagini

Un'immagine digitale viene acquisita con strumenti specifici per il tipo di dato che si sta trattando: ad esempio un telescopio per un'immagine di una galassia, un microscopio elettronico per l'immagine di un batterio. Indipendentemente dal tipo di immagine che si sta studiando, uno strumento per l'acquisizione è comunemente composto da due parti:

- ▷ la prima parte è in grado di captare le radiazioni emesse dall'oggetto di cui si vuole avere l'immagine. Ad esempio, nel caso di un telescopio il dispositivo in questione è il C.C.D. (Charge Copuled Device, dispositivo a scorrimento di carica): è composto da materiali semiconduttori, che convertono la luce incidente in cariche elettriche;
- ▷ la seconda parte è costituita dal dispositivo che trasforma il segnale fisico in un segnale digitale, cioè in dati trattabili dagli elaboratori; ad esempio, nel caso del telescopio sopracitato, è costituito dal dispositivo che tramuta le cariche elettriche in dati numerici gestibile da computer.

L'acquisizione di queste immagini non è esente da errori: sussistono alcuni effetti, dovuti a fattori fisici, che non permettono di ottenere un'immagine 'pura' dell'oggetto che si sta esaminando.

La degradazione dell'immagine è dovuta ai seguenti fattori: lo sfocamento (*blurring*) dovuto alla struttura dello strumento con cui si sta lavorando e causato, ad esempio, da diffrazioni, aberrazioni o altri fenomeni fisici analoghi; un'altra causa di degradazione sta nel fatto che si effettua un passaggio da uno spazio continuo (la realtà) ad uno spazio discreto (la memoria del computer), cioè si sta effettuando una transizione tra un'immagine analogica ad una digitale tramite un processo di discretizzazione. Questa discretizzazione dipende ancora una volta dalla struttura dello strumento, nella cui costruzione normalmente si tiene conto di teoremi di campionamento, e i dati sono (nella maggior parte dei casi) sovracampionati. Inoltre, si introducono degli errori statistici dovuti al processo di digitalizzazione; i dati raccolti, quindi, sono in realtà realizzazioni di variabili aleatorie.

Il modello generale della formazione di un'immagine digitale è dato da

$$g = \underbrace{K(f)}_{\substack{\text{degradazione dovuta} \\ \text{allo strumento}}} + \underbrace{\eta}_{\substack{\text{errore} \\ \text{statistico}}}$$

dove f è l'immagine reale dell'oggetto che si sta considerando, K è un operatore che modifica l'immagine originale, η è l'errore di tipo statistico descritto prima e g è l'immagine ottenuta alla fine del processo.

1.2 Modello Matematico Continuo

Un'immagine di un'oggetto viene considerata come una funzione $f : \Omega \rightarrow \mathbb{R}$, dove $\Omega \subseteq \mathbb{R}^2$, in cui (x, y) sono le variabili spaziali e dove $f(x_0, y_0)$ determina l'intensità dell'oggetto alle coordinate (x_0, y_0) .

1.2.1 Blurring

Il processo di formazione dell'immagine è definito tramite l'azione di un operatore \tilde{K} , supposto lineare e continuo.

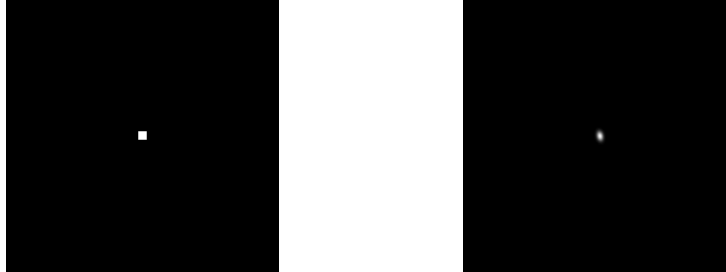
Il modello continuo è quindi definito dalla formula

$$g(x, y) = \iint_{(s,t) \in \Omega} \tilde{K}(x, s, y, t) f(s, t) ds dt \quad (1.1)$$

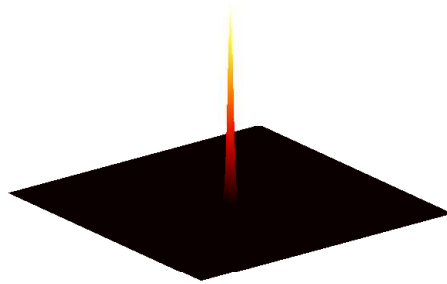
che ha la forma di un integrale di Fredholm di prima specie. L'operatore \tilde{K} prende il nome di *risposta in impulso del sistema*: considerando una sorgente puntiforme di intensità unitaria centrata nel punto (x_0, y_0) , modellata da una δ di Dirac centrata in quel punto, allora l'immagine restituita dal sistema è

$$g(x, y) = \iint_{(s,t) \in \Omega} \tilde{K}(x, s, y, t) \delta(s - x_0, t - y_0) ds dt = \tilde{K}(x, x_0, y, y_0) \quad (1.2)$$

cioè l'immagine di una sorgente puntiforme di intensità 1 è proprio la risposta in impulso del sistema (fig 1.1).



(a) Sorgente puntiforme di intensità unitaria. (b) Immagine della sorgente puntiforme.



(c) Rappresentazione tridimensionale della PSF.

Figura 1.1: In 1.1(a), l'immagine originale; in 1.1(b), l'immagine ottenuta; in 1.1(c), la point spread function vista nelle tre dimensioni.

In questo caso, la risposta in impulso di sistema viene detta anche *Point Spread Function* (PSF), ovvero la funzione di allargamento del punto. L'azione della PSF sul punto (e quindi su tutta l'immagine) viene detto *blurring* (sfocamento), in quanto rende meno nitidi i dettagli.

Nel caso in cui la formazione dell'immagine di una sorgente puntiforme (e quindi anche di un'immagine più complessa) non dipenda dalla posizione nello spazio del punto stesso, si dice che il sistema è *spazio invariante* e l'operatore \tilde{K} prende la forma

$$\tilde{K}(x, s, y, t) = K(x - s, y - t) \quad (1.3)$$

Le ipotesi assunte sull'operatore K sono:

$$K(x, y) \geq 0 \quad \forall x, y \in \mathbb{R}, \quad \int_{\mathbb{R}} K(x, y) dx dy = 1 \quad (1.4)$$

Sostituendo quindi (1.3) in (1.1) si ottiene

$$g(x, y) = \iint_{(s,t) \in \Omega} K(x-s, y-t) f(s, t) ds dt \quad (1.5)$$

cioè l'immagine g ottenuta alla fine del processo di acquisizione è il prodotto di convoluzione $K * f$.

In generale, la PSF è conosciuta tramite stime oppure tramite una funzione vera e propria. Le stime si possono ottenere grazie ad esperimenti: considerando una sorgente puntiforme come descritto prima, si acquisiscono varie immagini di questa sorgente e in questo modo, per ciò che si è osservato, si può avere una buona stima della risposta in impulso del sistema. Nel caso migliore, invece, si dispone della funzione matematica che descrive il comportamento della PSF, funzione che viene fornita dai costruttori del sistema di acquisizione di immagini.

Ad esempio, la PSF mostrata in figura (1.1) è descritta dalla funzione

$$p(x, y) = C_0 \exp \left(-\frac{1}{2} \begin{bmatrix} x-k \\ y-l \end{bmatrix}^T \begin{bmatrix} s_1^2 & \rho^2 \\ \rho^2 & s_2^2 \end{bmatrix}^{-1} \begin{bmatrix} x-k \\ y-l \end{bmatrix} \right) \quad (1.6)$$

dove (k, l) è il centro della PSF, mentre s_1, s_2 sono i parametri regolanti la larghezza e l'altezza e ρ rappresenta l'orientamento.

Altri esempi di point spread functions sono:

- Out-of-focus Blur

$$p(x, y) = \begin{cases} C_1 \frac{1}{\pi r^2} & (x-k)^2 + (y-l)^2 \leq r^2 \\ 0 & \text{altrimenti} \end{cases} \quad (1.7)$$

dove (k, l) è ancora il centro della PSF e $r \in \mathbb{R}$, $r > 0$, è il raggio dello sfocamento; in questo caso, il sistema di acquisizione non è 'a fuoco'.

- Moffat Blur:

$$p(x, y) = C_2 \left(1 + \begin{bmatrix} x-k \\ y-l \end{bmatrix}^T \begin{bmatrix} s_1^2 & \rho^2 \\ \rho^2 & s_2^2 \end{bmatrix}^{-1} \begin{bmatrix} x-k \\ y-l \end{bmatrix} \right)^{-\beta} \quad (1.8)$$

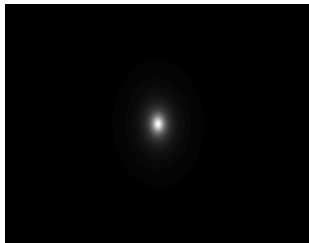
dove s_1, s_2, ρ sono parametri come nel caso precedente, (k, l) il centro, mentre $\beta > 0$ controlla l'andamento della funzione, che in generale è asintoticamente più lento di una PSF Gaussiana. Questo tipo di funzione modella la PSF di un telescopio astronomico.

- Horizontal motion blur

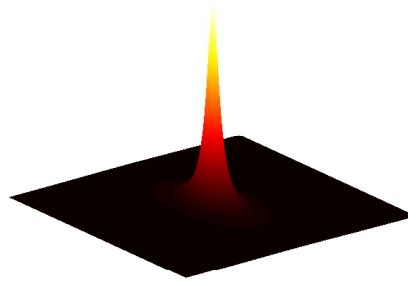
$$p(x, y) = \begin{cases} C_3 & a \leq x \leq b \\ 0 & \text{altrimenti} \end{cases} \quad (1.9)$$

dove $a, b > 0$; questa PSF modella l'acquisizione di un'immagine in movimento. Nel caso esposto, si considera uno spostamento (dell'oggetto o del sistema) orizzontale, ma è possibile avere un movimento anche verticale, diagonale o circolare.

Le costanti C_i che appaiono nelle formule hanno il compito di normalizzare la funzione, in modo che l'integrale su tutto lo spazio abbia valore 1.



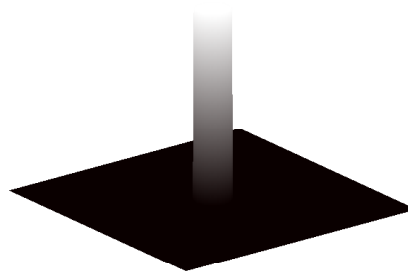
(a) Moffat Blur.



(b) Moffat Blur tridimensionale.



(c) Out-of-focus Blur.



(d) Out-of-focus Blur tridimensionale.

Figura 1.2: Esempi di PSF.

1.2.2 Noise

In aggiunta agli effetti di blurring dati dal sistema di acquisizione, l'altra componente di degradazione dell'immagine è data dal rumore (*noise*). Può nascere

da diverse cause, può essere lineare, non lineare, moltiplicativo o additivo. Il modello più comune è il noise lineare, dovuto principalmente a tre cause.

- ▷ Il dispositivo a scorrimento di carica ha il compito di tramutare la luce incidente in cariche elettriche, cioè ‘conta’ i fotoni incidenti sul chip di materiale semiconduttore e converge questo conteggio in cariche elettriche. Questo conteggio non è esente da imperfezioni e normalmente gli errori che affliggono un processo di questo tipo seguono una distribuzione di Poisson:

$$p_\lambda(k) = e^{-\lambda} \frac{\lambda^k}{k!} \quad (1.10)$$

Questo tipo di errore dipende direttamente dall’immagine considerata, come vedremo nel capitolo (1.3.2), in quanto il parametro λ è dipendente dal prodotto di convoluzione Af . Inoltre, si devono considerare le emissioni di fotoni dell’ambiente circostante: il numero di questi fotoni segue anch’esso una distribuzione di Poisson di parametro γ fissato.

- ▷ Il processo di discretizzazione e la conversione di dati da analogici a digitali hanno come inconveniente il cosiddetto *rumore di lettura*: si ha sia quando si tramuta il conteggio dei fotoni in cariche elettriche, sia quando queste cariche elettriche vengono ‘lette’ e ‘convertite’ in dati numerici. È costituito da valori casuali identicamente e indipendentemente distribuiti ed è chiamato anche *whitenoise*¹; il suo modello matematico è una curva gaussiana:

$$p_{\mu,\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp - \frac{(x - \mu)^2}{2\sigma^2}$$

dove la media è zero e la deviazione standard (fissata) è proporzionale all’ampiezza del rumore.

- ▷ La conversione da dato analogico a dato digitale restituisce un altro tipo di errore, detto *errore di quantizzazione*, quando il numero di bit (quindi le informazioni che abbiamo sull’oggetto) utilizzato è relativamente basso. Anch’esso è modellizzato da rumore bianco uniformemente distribuito la cui deviazione standard è inversamente proporzionale ai numeri di bit usati.

Quindi, l’immagine affetta da errori è ottenuta nella seguente forma:

$$g = \eta_{Pois(\lambda)} + \eta_{bg} + \eta_{wn} + \eta_{quant}$$

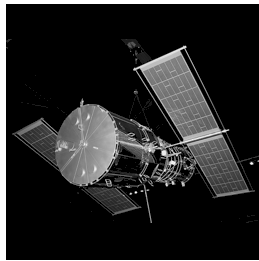
Dove $\eta_{Pois(\lambda)}$ è il rumore poissoniano, η_{bg} è il rumore di background, η_{wn} è il rumore bianco e infine η_{quant} rappresenta l’errore di quantizzazione. Trattando alcuni tipi di immagini, ad esempio PET od immagini astronomiche, è possibile considerare l’errore di tipo esclusivamente poissoniano e supporre che l’errore di quantizzazione sia trascurabile.

¹Prende il nome di rumore bianco in quanto ha delle proprietà spettrali simile alla luce bianca.

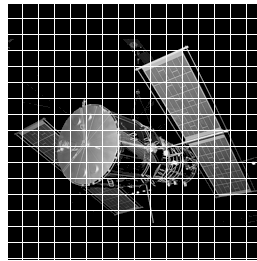
1.3 Modello Matematico Discreto

Per poter affrontare il problema al computer, è necessario discretizzare il modello. Un'immagine non sarà più una funzione continua $f : \Omega \rightarrow \mathbb{R}$, ma sarà una matrice $f \in \mathcal{M}_{m \times n}(\mathbb{R})$, dove $f(i, j)$ sarà l'elemento di riga i e colonna j ed indicherà come nel caso precedente l'intensità dell'immagine. Si lavorerà con livelli di intensità nella scala di grigio, quindi i valori che f può assumere possono variare negli intervalli² $[0, 255]$, $[0, 65535] \subset \mathbb{N}$, oppure nell'intervallo $[0, 1] \subset \mathbb{R}$.

1.3.1 Blurring



(a) Immagine continua.



(b) Immagine discreta; ogni quadrato rappresenta un elemento della matrice.

Figura 1.3: Discretizzazione di un'immagine continua.

Considerando il fatto che l'operatore K può essere visto come l'immagine acquisita di una sorgente puntiforme, allora anch'esso si può considerare come una matrice. Supponendo come prima di lavorare con operatori lineari e continui, allora l'acquisizione dell'immagine può essere scritta sotto la forma

$$\mathbf{g} = A\mathbf{f} \quad (1.11)$$

dove $\mathbf{f} = \text{vec}(f)$, cioè la matrice f viene riscritta sotto forma di vettore, ponendo le colonne in ordine lessicografico una sotto l'altra. Le dimensioni della matrice A , costruita a partire da K , saranno $m \cdot n \times m \cdot n$, dove $n \times m$ sono le dimensioni di f e g . Il modello discreto di acquisizione di un'immagine prende quindi la forma di un *sistema lineare*.

Assumendo ancora di lavorare con operatori spazio invarianti, si avrà nuovamente un prodotto di convoluzione, che prenderà la forma

$$g(i, j) = \sum_{i'=1}^m \sum_{j'=1}^n f(i', j') P(i' - i, j' - j) \quad (1.12)$$

²In un intervallo del tipo $[a, b]$, a rappresenta il colore nero, b il bianco

dove P è la matrice ottenuta mediante lo *shift* e il prolungamento periodico dell'immagine K della PSF.

Effettuare lo shift (rispetto al centro) di una matrice significa invertire le righe e le colonne della matrice rispetto al centro della matrice stessa. Ad esempio, se la matrice P è scritta

$$\begin{pmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{pmatrix}$$

allora il suo shift rispetto al centro è

$$\begin{pmatrix} p_{33} & p_{32} & p_{31} \\ p_{23} & p_{22} & p_{21} \\ p_{13} & p_{12} & p_{11} \end{pmatrix}$$

Per capire la funzione del prolungamento periodico enunciato prima, si osservi il prodotto di convoluzione: effettuare questo prodotto significa che ogni elemento³ della matrice g è determinato dalla somma pesata di tutti gli elementi dell'immagine f , dove i pesi sono gli elementi della P . Effettuare una convoluzione, significa far 'scorrere' la matrice P su ogni elemento dell'immagine f , moltiplicare gli elementi corrispondenti e sommarli. Ad esempio, se la matrice P è quella ottenuta nel precedente esempio, per determinare il pixel g_{22} si effettua il passaggio della P in questo modo:

$$\left[\begin{pmatrix} p_{33} \cdot f_{11} & p_{32} \cdot f_{12} & p_{31} \cdot f_{13} \\ p_{23} \cdot f_{21} & p_{22} \cdot f_{22} & p_{21} \cdot f_{23} \\ p_{13} \cdot f_{31} & p_{12} \cdot f_{32} & p_{11} \cdot f_{33} \end{pmatrix} \right]$$

si sommano tutti gli elementi e si ottiene

$$g_{22} = f_{11}p_{33} + f_{12}p_{32} + f_{13}p_{31} + f_{21}p_{23} + f_{22}p_{22} + f_{23}p_{21} + f_{31}p_{13} + f_{32}p_{12} + f_{33}p_{11}$$

Ma effettuando lo stesso passaggio su un elemento del 'bordo' dell'immagine, ad esempio per ottenere il pixel g_{21} , si nota il sorgere di un problema:

$$\left[\begin{matrix} p_{33} \bullet \\ p_{23} \bullet \\ p_{13} \bullet \end{matrix} \begin{pmatrix} p_{32} \cdot f_{12} & p_{31} \cdot f_{13} \\ p_{23} \cdot f_{21} & p_{22} \cdot f_{22} \\ p_{13} \cdot f_{31} & p_{12} \cdot f_{32} \end{pmatrix} \begin{matrix} f_{13} \\ f_{23} \\ f_{33} \end{matrix} \right]$$

non si capisce cioè per cosa vadano moltiplicati gli elementi della prima colonna di P . Per fare ciò, si devono assegnare le condizioni al contorno, cioè decidere il comportamento dell'immagine al di fuori dei confini che si stanno considerando, in quanto in realtà si sta considerando solo una porzione finita di spazio.

Le condizioni al contorno usate possono essere:

³D'ora in poi un elemento di una matrice bidimensionale che rappresenta un'immagine verrà chiamato *pixel*, abbreviazione di *picture element*. Nel caso di matrici 3D, verrà chiamato *voxel*, abbreviazione di *volume element*.

- ▷ *Zero Boundary conditions*: si suppone che l'immagine sia immersa in uno sfondo totalmente nero. Si effettua allora il cosiddetto *zero-padding*, si aggiungono cioè degli zeri attorno all'immagine considerata.

Ad esempio, se F è scritta per blocchi nel seguente modo

$$\begin{pmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{pmatrix}$$

allora il suo zero-padding è

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & F_{11} & F_{12} & 0 \\ 0 & F_{21} & F_{22} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

dove 0 rappresenta una matrice di dimensioni opportune.

Se applichiamo il procedimento all'immagine in figura (1.3(a)) otteniamo il risultato in figura (1.4).

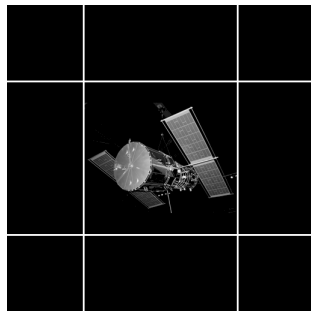


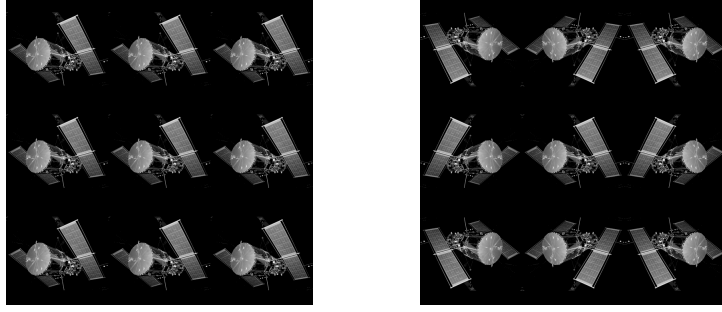
Figura 1.4: Zero-padding di un'immagine

- ▷ *Periodic Boundary conditions*: si suppone che l'immagine si ripeta identicamente in tutte le direzioni, infinitamente. Per ottenere ciò, si circonda l'immagine con tante copie di se stessa. Ad esempio, se si ha un'immagine rappresentata dalla matrice X , per utilizzare queste condizioni al contorno si considererà l'immagine rappresentata dalla matrice seguente:

$$\begin{pmatrix} X & X & X \\ X & X & X \\ X & X & X \end{pmatrix}$$

Utilizzando l'immagine degli esempi precedenti, avremo quindi il risultato in figura (1.5(a)).

- ▷ *Reflexive Boundary conditions*: in questo caso, si suppone che l'immagine sia circondata da riflessioni speculari di se stessa. Ad esempio, se



(a) Periodic boundary conditions.

(b) Reflexive boundary conditions.

Figura 1.5: Condizioni al contorno.

l'immagine F è rappresentata dalla matrice

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

allora, imponendo queste condizioni al contorno si ottiene

$$\left(\begin{array}{ccc|ccc|ccc} 9 & 8 & 7 & 7 & 8 & 9 & 9 & 8 & 7 \\ 6 & 5 & 4 & 4 & 5 & 6 & 6 & 5 & 4 \\ 3 & 2 & 1 & 1 & 2 & 3 & 3 & 2 & 1 \\ \hline 3 & 2 & 1 & 1 & 2 & 3 & 3 & 2 & 1 \\ 6 & 5 & 4 & 4 & 5 & 6 & 6 & 5 & 4 \\ 9 & 8 & 7 & 7 & 8 & 9 & 9 & 8 & 7 \\ \hline 9 & 8 & 7 & 7 & 8 & 9 & 9 & 8 & 7 \\ 6 & 5 & 4 & 4 & 5 & 6 & 6 & 5 & 4 \\ 3 & 2 & 1 & 1 & 2 & 3 & 3 & 2 & 1 \end{array} \right)$$

Applicando ancora tutto ciò alla figura (1.3(a)) si ottiene (1.5(b)).

La scelta delle condizioni al contorno è dettata dal tipo di problema e dal tipo di immagine che si sta considerando: ad esempio lo zero padding si utilizza nel caso di immagini astronomiche, in quanto gli oggetti in considerazione sono immersi nello spazio profondo, cioè in un background totalmente (o quasi) nero.

Una volta effettuata questa scelta, si può effettuare il prodotto di convoluzione: tornando all'esempio precedente, imponendo le condizioni al contorno degli zeri e quindi effettuando lo zero padding di F , per ottenere il pixel g_{21} si avrà

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ p_{33} \cdot 0 & p_{32} \cdot f_{12} & p_{31} \cdot f_{13} & f_{13} & 0 \\ p_{23} \cdot 0 & p_{23} \cdot f_{21} & p_{22} \cdot f_{22} & f_{23} & 0 \\ p_{13} \cdot 0 & p_{13} \cdot f_{31} & p_{12} \cdot f_{32} & f_{33} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

cioè

$$g_{21} = f_{12}p_{32} + f_{13}p_{31} + f_{21}p_{23} + f_{22}p_{22} + f_{31}p_{13} + f_{32}p_{12}$$

Effettuando lo stesso calcolo per tutti gli altri elementi g_{ij} , ed utilizzando la notazione $\mathbf{g} = \text{vec}(g)$, si ottiene

$$\begin{pmatrix} g_{11} \\ g_{21} \\ g_{31} \\ g_{12} \\ g_{22} \\ g_{32} \\ g_{13} \\ g_{23} \\ g_{33} \end{pmatrix} = \begin{pmatrix} p_{22} & p_{12} & 0 & p_{21} & p_{11} & 0 & 0 & 0 & 0 \\ p_{32} & p_{22} & p_{12} & p_{31} & p_{21} & p_{11} & 0 & 0 & 0 \\ 0 & p_{23} & p_{22} & 0 & p_{31} & p_{21} & 0 & 0 & 0 \\ \hline p_{23} & p_{13} & 0 & p_{22} & p_{12} & 0 & p_{21} & p_{11} & 0 \\ p_{33} & p_{23} & p_{13} & p_{32} & p_{22} & p_{12} & p_{31} & p_{21} & p_{11} \\ 0 & p_{33} & p_{23} & 0 & p_{32} & p_{22} & 0 & p_{31} & p_{21} \\ \hline 0 & 0 & 0 & p_{23} & p_{13} & 0 & p_{22} & p_{12} & 0 \\ 0 & 0 & 0 & p_{33} & p_{23} & p_{13} & p_{32} & p_{22} & p_{12} \\ 0 & 0 & 0 & 0 & p_{33} & p_{23} & 0 & p_{32} & p_{22} \end{pmatrix} \begin{pmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ f_{22} \\ f_{32} \\ f_{13} \\ f_{23} \\ f_{33} \end{pmatrix}$$

cioè il sistema iniziale

$$\mathbf{g} = A\mathbf{f}$$

Conoscendo le condizioni al bordo e il tipo di PSF si può quindi costruire la matrice A , che può avere particolari strutture, dettate appunto dalle condizioni al bordo. Le strutture possibili sono:

- ▷ *Matrice di Toeplitz a blocchi di Toeplitz (BTTB)*: una matrice si dice di Toeplitz se i suoi elementi sono costanti su ogni diagonale; una matrice a blocchi si dice di Toeplitz se i blocchi sono costanti su ogni diagonale. Una matrice si dice di Toeplitz a blocchi di Toeplitz se è una matrice a blocchi di Toeplitz e ogni blocco è di Toeplitz (come nell'esempio precedente).
- ▷ *Matrice circolante a blocchi circolanti (BCCB)*: una matrice si dice circolante se ogni colonna (e ogni riga) è uno shift periodico della precedente colonna (riga); si dice circolante a blocchi se ogni riga (colonna) formata dai blocchi è uno shift periodico della precedente riga (colonna). Una matrice si dice quindi circolante a blocchi circolanti se è circolante a blocchi e ogni blocco è circolante.
- ▷ *Matrice di Hankel*: una matrice si dice di Hankel se i suoi elementi sono costanti su ogni antidiagonale. Si può avere quindi una matrice a blocchi di Toeplitz, con blocchi di Hankel (BTHB), una matrice a blocchi di Hankel con blocchi di Toeplitz (BHTB) oppure una matrice a blocchi di Hankel con blocchi di Hankel (BHHB).

Nel caso condizioni al contorno nulle, A è una BTTB (visto nell'esempio precedente), mentre nel caso periodico A è una BCCB; infine, nel caso riflessivo la matrice A è la somma di quattro matrici BTTB, BTHB, BHTB e BHHB.

Il modello discreto per la formazione di un'immagine digitale può essere quindi visto come la convoluzione tra la matrice f e l'operatore P , oppure come il prodotto matrice-vettore Af ; in entrambi i casi, possono sorgere dei problemi: il costo computazionale della convoluzione discreta per matrici di

$$\begin{pmatrix} p_3 & p_2 & p_1 & 0 \\ p_4 & p_3 & p_2 & p_1 \\ 0 & p_4 & p_3 & p_2 \\ 0 & 0 & p_4 & p_3 \end{pmatrix} \begin{pmatrix} p_1 & p_2 & p_3 & p_4 \\ p_4 & p_1 & p_2 & p_3 \\ p_3 & p_4 & p_1 & p_2 \\ p_2 & p_3 & p_4 & p_1 \end{pmatrix} \begin{pmatrix} p_1 & p_2 & p_3 & p_4 \\ p_2 & p_3 & p_4 & p_5 \\ p_3 & p_4 & p_5 & p_6 \\ p_4 & p_5 & p_6 & p_7 \end{pmatrix}$$

Figura 1.6: Tipi di matrici: Toeplitz, circolante, Hankel.

grandi dimensioni, infatti, è molto alto⁴, mentre il prodotto Af può coinvolgere matrici di dimensioni troppo elevate⁵. Una soluzione a questi problemi può essere l'utilizzo della trasformata di Fourier e del seguente teorema:

Teorema 1.1. (di convoluzione) Siano $f, g \in \mathcal{L}^1(\mathbb{R})$, sia

$$\mathcal{F}\{f\}(\xi) = \hat{f}(\xi) = \frac{1}{2\pi} \int_{\mathbb{R}} e^{-i\langle x, \xi \rangle} f(x) dx \quad (1.13)$$

la trasformata di Fourier della funzione f . Allora

$$\mathcal{F}\{f * g\} = \hat{f} \cdot \hat{g} \quad (1.14)$$

e

$$\mathcal{F}\{f \cdot g\} = \hat{f} * \hat{g} \quad (1.15)$$

ovvero la trasformata della convoluzione di due funzioni f e g è il prodotto delle trasformate, e analogamente la trasformata del prodotto di f e g è la convoluzione delle trasformate \hat{f} e \hat{g} .

Considerando la versione discreta⁶ della (1.13)

$$\mathbf{DFT}(f_k) = \sum_{n=0}^{N-1} f_k e^{\frac{2\pi i}{N} kn} \quad (1.16)$$

(dove f_k sono una sequenza di numeri complessi, $k = 0, \dots, N-1$) e applicandola alla (1.12) si ha

$$\mathbf{DFT}(g) = \mathbf{DFT}(f) \cdot \mathbf{DFT}(P) \quad (1.17)$$

quindi per ottenere g si applica la trasformata inversa

$$g = \mathbf{IDFT}[\mathbf{DFT}(f) \cdot \mathbf{DFT}(P)] \quad (1.18)$$

in questo modo i tempi di calcolo si riducono. L'algoritmo più utilizzato per applicare la trasformata di Fourier discreta è l'algoritmo **FFT**, Fast Fourier Transform, il cui costo computazionale si riduce quando si lavora con immagini che hanno dimensioni che sono potenze di 2.

⁴Ad esempio, utilizzando la funzione `conv2` implementata in MatLab, è necessario aspettare un lasso di tempo inaccettabile nelle applicazioni.

⁵Ad esempio se l'immagine f ha dimensioni 256×256 , il prodotto matrice-vettore implica la moltiplicazione di una matrice A di dimensioni $256^2 \times 256^2 = 65536 \times 65536$ per un vettore f di dimensioni $256^2 \times 1 = 65536 \times 1$.

⁶Discrete Fourier Transform.

1.3.2 Noise

Come si è detto nel capitolo (1.2.2), il rumore totale che interviene sull'immagine è dato da rumore gaussiano, da rumore poissoniano oppure da una combinazione dei due.

Il rumore gaussiano bianco interviene sull'immagine in misura percentuale: ad esempio, se il livello di rumore è dell'1%, questo significa che il rapporto fra il rumore additivo e l'immagine perturbata con la PSF vale 0.01 (in norma⁷):

$$\frac{\|\eta\|}{\|K(f)\|} = 0.01$$

Il rumore aggiunto ha normalmente media pari a 0 e deviazione standard pari al livello di rumore:

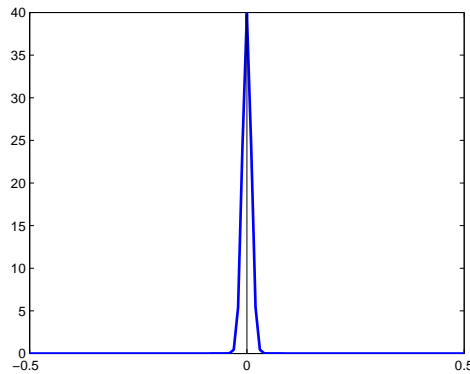


Figura 1.7: Rumore gaussiano bianco di media 0 e deviazione standard 0,01.

Dalla figura (1.7) si può vedere che c'è poca variazione rispetto al valore medio, ma anche questa piccola variazione può creare effetti altamente indesiderati.

Per simulare al computer la presenza di rumore sull'immagine, si utilizzano le funzioni pseudorandom implementate per generare realizzazioni di variabili aleatorie. Per aggiungere rumore gaussiano, si può utilizzare⁸ la funzione `randn()`, che genera numeri pseudocasuali da una distribuzione gaussiana di media 0 e varianza 1; il comando `randn(nx,ny)` genera una matrice di dimensioni $nx \times ny$ in cui ogni elemento è un numero casuale di media 0 e varianza 1. Il passo successivo è normalizzare la matrice; infine, si moltiplica la matrice così ottenuta per il livello di rumore $Nlevel$ e per la norma dell'immagine perturbata.

Il procedimento è il seguente:

1. `E = randn(nx,ny)`

⁷Nelle applicazioni, le norme considerate sono quella euclidea e la norma di Frobenius.

⁸D'ora in poi, si farà riferimento alle funzioni implementate in MatLab.

$$2. E = E/\text{norm}(E)$$

$$3. G = Kf + Nlevel * \text{norm}(Kf) * E$$

dove Kf indica l'immagine ottenuta dopo la convoluzione con la PSF, $Nlevel$ è il livello del rumore e $\text{norm}()$ indica una qualsiasi norma, specificata a seconda delle applicazioni.

Si nota che in realtà effettuare il prodotto $Nlevel * E$, dove E è la matrice casuale normalizzata, significa generare numeri casuali da una distribuzione di media zero e varianza $Nlevel^2$.

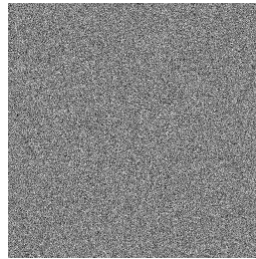


Figura 1.8: Errore gaussiano additivo, con livello di rumore pari a 1%.

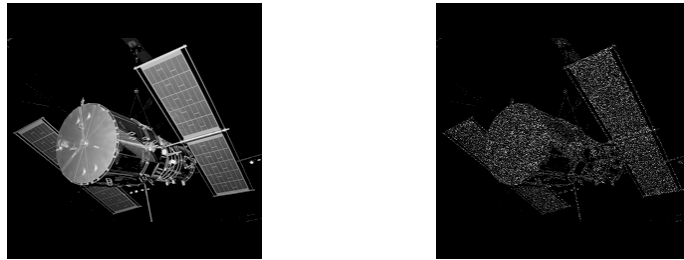
Il procedimento per l'aggiunta di rumore poissoniano è diverso rispetto al caso precedente: dato che questo tipo di rumore interviene nel conteggio dei fotoni che incidono sul CCD, esso è dipendente dal segnale incidente stesso, cioè dall'immagine che si sta acquisendo. Per simulare tutto ciò al computer, si utilizza la funzione `imnoise()`, che prende in input l'immagine a cui aggiungere il rumore e il tipo di noise voluto; per aggiungere rumore poissoniano all'immagine g , si scrive quindi `imnoise(g, 'poisson')`, ottenendo il risultato in figura (1.9(a)).

Dalla (1.9(a)) sembra non esserci alcuna differenza dall'immagine originale, ma osservando la (1.9(b)), che rappresenta la differenza fra l'immagine con rumore e quella pulita, si nota che in realtà sono presenti delle perturbazioni.

Il fatto interessante è che il rumore più alto si ha in corrispondenza dell'oggetto, mentre il background rimane (quasi) invariato: questa è la conseguenza del fatto che il rumore dipende proprio dal segnale incidente.

Un ulteriore tipo di rumore sull'immagine può provenire da un malfunzionamento del dispositivo CCD: se uno dei suoi componenti è guasto, oppure spento, può accadere che non conti nessun fotone incidente. Un altro guasto potrebbe implicare la restituzione del valore massimo ottenibile nel conteggio⁹. Questo

⁹Il caso in cui il conteggio venga eseguito male è contemplato nel rumore poissoniano.



(a) Immagine con rumore poissoniano. (b) Differenza fra l'immagine originale e quella rumorosa.

Figura 1.9: Rumore poissoniano.

tipo di malfunzionamento produce il rumore detto sale e pepe, in quanto appaiono sull'immagine pixel bianchi e neri, come se fossere granelli di sale e pepe.

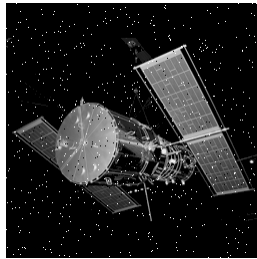


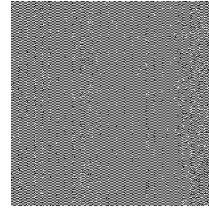
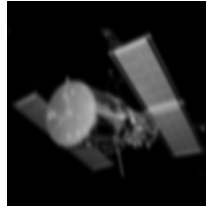
Figura 1.10: Salt&Pepper Noise.

1.4 Image Deblurring

Avendo modellizzato la formazione dell'immagine con l'equazione

$$g = K(f) + \eta$$

intesa sia come sistema lineare che come prodotto di convoluzione, lo scopo è riottenere l'immagine f . Il problema però è mal posto: l'aggiunta del rumore η produce grandi variazioni nel calcolo della soluzione f , ed assieme al blurring, inoltre, non permette di raggiungere il risultato esatto. Sono quindi necessari metodi di regolarizzazione che tengano sotto controllo l'influenza del rumore sulla soluzione esatta.



(a) Immagine perturbata con una PSF di tipo gaussiano di parametri (2, 3, 1) e con rumore gaussiano al 1%.

(b) Immagine ricostruita con anti-trasformata di Fourier.

Figura 1.11: Ricostruzione naïve.

Si può osservare, ad esempio, la figura (1.11(b)) che è una ricostruzione di (1.11(a)) ottenuta tramite le trasformate di Fourier: dalla 1.17 si ottiene

$$f = \text{IDFT}\{\text{DFT}(P) ./ \text{DFT}(g)\} \quad (1.19)$$

dove la notazione ./ indica la divisione elemento per elemento.

Capitolo 2

Metodi di Regolarizzazione

In questo capitolo verranno esposte le basi teoriche per affrontare problemi di blind deconvolution nel caso delle immagini astronomiche.

Nella prima sezione ci si occuperà di una definizione di pseudo-inversa per operatori non invertibili.

Nella seconda sezione si porranno le basi per la teoria spettrale necessaria all'analisi dei metodi.

Nella terza sezione si affronterà il problema degli operatori compatti derivante dalle applicazioni.

Nella quarta sezione ci si occuperà di esporre definizioni e risultati sui metodi di regolarizzazione, metodi necessari a tenere sotto controllo l'influenza del rumore.

Nella quinta sezione si esporranno i metodi di regolarizzazione nel discreto, portando come esempi il metodo di Tikhonov e la Total Variation.

Nella sesta sezione si vedrà un esempio di funzione di scelta del parametro di regolarizzazione.

2.1 Pseudo Inversa di Moore-Penrose

Il problema generale che si tenta di risolvere è

$$Kx = y \tag{2.1}$$

dove si suppone che K sia un operatore tra due spazi di Hilbert \mathcal{X} e \mathcal{Y} .¹

Def. 2.1. Un problema del tipo $Kx = y$ si dice *ben posto* se

- (i) $\forall g \in \mathcal{Y} \exists f \in \mathcal{X}$ per cui valga l'equazione 2.1;
- (ii) la soluzione f è unica;
- (iii) la soluzione è stabile rispetto alle perturbazioni su g ; questo significa che se $Kf^* = g^*$ e $Kf = g$, allora $f \rightarrow f^*$ per $g \rightarrow g^*$.

¹In questo elaborato, si considererà solo il caso discreto, quindi si avrà che $\mathcal{X} = \mathcal{Y} = \mathbb{R}^N$, dotato del consueto prodotto scalare e della norma euclidea.

Se (2.1) è ben posto, allora K ha un'inversa K^{-1} ben definita e continua, per la quale vale $K^{-1}(K(x)) = x$ per ogni $x \in \mathcal{X}$, e si ha $\text{Range}(K) = \mathcal{Y}$. Il problema consiste nel fatto che si ha a disposizione non il dato esatto y , ma in realtà una sua perturbazione y_δ . Come conseguenza K potrebbe non essere invertibile, quindi potrebbe non esistere una soluzione x o nel caso in cui esista potrebbe non essere unica. L'interesse consiste nel cercare una soluzione \hat{x} che meglio approssimi la soluzione vera (nel caso in cui esista). Si pongono allora le seguenti definizioni:

Def. 2.2. Sia $K : \mathcal{X} \rightarrow \mathcal{Y}$ un operatore fra i due spazi di Hilbert \mathcal{X} e \mathcal{Y} . Un elemento $x_{ls} \in \mathcal{X}$ si dice *soluzione ai minimi quadrati* per $Kx = y$ se vale

$$\|Kx_{ls} - y\|_{\mathcal{Y}} \leq \|Kx - y\|_{\mathcal{Y}}$$

per ogni $x \in \mathcal{X}$.

Una soluzione ai minimi quadrati non sempre esiste; nel caso in cui x_{ls} esista, allora l'insieme di tutte le soluzioni ai minimi quadrati è dato dallo spazio affine $x_{ls} + \text{Ker}(K)$. Si può porre quindi la seguente

Def. 2.3. Un elemento $x \in \mathcal{X}$ è la *miglior soluzione approssimante* per $Kx = y$ se è una soluzione ai minimi quadrati e vale inoltre

$$\|x\|_{\mathcal{X}} = \inf\{\|z\|_{\mathcal{X}} \mid z \text{ è una soluzione ai minimi quadrati}\}$$

Un modo per arrivare ad ottenere la miglior soluzione approssimante per l'equazione $Kx = y$ è definire l'inverso generalizzato di Monre-Penrose, restringendo il dominio e il codominio di K in modo che il nuovo operatore sia effettivamente invertibile. Si pone quindi la seguente

Def. 2.4. Sia K un operatore fra due spazi di Hilbert \mathcal{X} e \mathcal{Y} . Ponendo

$$\tilde{K} := K|_{\text{Ker}(K)^\perp} : \text{Ker}(K)^\perp \rightarrow \text{Range}(K)$$

si definisce pseudo inversa K^\dagger di K l'unica estensione lineare di \tilde{K}^{-1}

$$K^\dagger : \mathcal{D}(K^\dagger) \rightarrow \text{Range}(K^\dagger)$$

con

$$\mathcal{D}(K^\dagger) := \text{Range}(K) + \text{Range}(K)^\perp$$

$$\text{Range}(K^\dagger) = \text{Ker}(K)^\perp$$

$$\text{Ker}(K^\dagger) = \text{Range}(K)^\perp$$

L'uguaglianza $K^\dagger K = Id$ ovviamente non vale, sussistono però le seguenti equazioni:

$$K^\dagger K K^\dagger = K^\dagger$$

$$K K^\dagger K = K$$

$$K^\dagger K = Id - P$$

$$K K^\dagger = Q|_{\mathcal{D}(K^\dagger)}$$

dove P è il proiettore ortogonale su $\text{Ker}(K)$ e Q è il proiettore su $\overline{\text{Range}(K)}$.

Il seguente risultato dà un'importante connessione con la terza proprietà della definizione (2.1): scrivendo una soluzione (ad esempio nel senso dei minimi quadrati) di $Kx = y$ come $x^\dagger = K^\dagger y$, se l'operatore K^\dagger risulta continuo allora perturbando leggermente il dato y la soluzione x^\dagger varierà leggermente anch'essa.

Teorema 2.1. L'operatore K^\dagger di Moore-Penrose ha un grafico chiuso. Inoltre, K^\dagger è limitato (quindi *continuo*) se e soltanto se $\text{Range}(K)$ è chiuso.

Per una dimostrazione di questo teorema si veda [EHN]. Il teorema successivo mostra la relazione presente fra una soluzione ai minimi quadrati e l'inverso di Moore-Penrose.

Teorema 2.2. Sia $y \in \mathcal{D}(K^\dagger)$. Allora $Kx = y$ ha un'unica migliore soluzione approssimante data da

$$x^\dagger = K^\dagger y$$

L'insieme di tutte le soluzioni è dato da $x^\dagger + \text{Ker}(K)$. Inoltre, x_{ls} è una soluzione ai minimi quadrati se e soltanto se vale la seguente equazione normale²

$$K^* K x = K^* y \quad (2.2)$$

dove K^* è l'aggiunto di K .

2.2 Teoria spettrale

In questa sezione³, verranno enunciati alcuni risultati di teoria spettrale per operatori autoaggiunti in spazi di Hilbert. L'interesse verterà poi sugli operatori compatti.

Def. 2.5. Un'operatore lineare Π tra due spazi di Hilbert \mathcal{X} e \mathcal{Y} si dice *proiettore ortogonale* se soddisfa i seguenti fatti:

- (i) $\Pi^2 = \Pi$
- (ii) $\Pi^* = \Pi$

Per un proiettore ortogonale Π vale che $\text{Ker}(\Pi) \perp \text{Rank}(\Pi)$, ed inoltre la sua norma, nello spazio di Hilbert considerato, vale 0 oppure 1. Si pone quindi la seguente definizione.

Def. 2.6. Sia \mathcal{X} uno spazio di Hilbert. Una famiglia di proiettori ortogonali $\{E_\lambda\}_{\lambda \in \mathbb{R}}$ prende il nome di *famiglia spettrale* o *risoluzione dell'identità* se soddisfa le seguenti condizioni

- (i) $E_\lambda E_\mu = E_{\min\{\lambda, \mu\}}$, $\lambda, \mu \in \mathbb{R}$;
- (ii) $E_{-\infty} = 0$, $E_\infty = Id$, dove $E_\lambda(x) = \lim_{\lambda \rightarrow \pm\infty}$ per ogni $x \in \mathbb{R}$;
- (iii) $E_{\lambda-0} = E_\lambda$, dove $E_{\lambda-0} = \lim_{\epsilon \rightarrow 0^+} E_{\lambda-\epsilon}(x)$ per ogni $x \in \mathbb{R}$.

²In letteratura, è chiamata anche equazione gaussiana.

³Per un discorso maggiormente approfondito, si possono consultare [EHN] e [SR].

Utilizzando questa definizione, è possibile costruire l'integrale di una funzione f continua reale a valori reali tramite le somme di Riemann

$$\sum_{i=1}^{\infty} f(\xi_i)(E_{\lambda_i} - E_{\lambda_{i-1}})(x)$$

dove i λ_i appartengono ad una scomposizione dell'intervallo $[a, b] \subset \mathbb{R}$, con $-\infty < a < b < \infty$, $\xi_i \in (\lambda_i, \lambda_{i-1}]$. Si dimostra che il limite della somma scritta sopra esiste, e viene denotato con

$$\int_a^b f(\lambda) dE_{\lambda}(x) \quad (2.3)$$

Si può estendere nel seguente modo per gli estremi che esplodono all'infinito.

Def. 2.7. Per ogni $x \in \mathcal{X}$ e per ogni funzione continua f su \mathbb{R} a valori reali, l'integrale

$$\int_{-\infty}^{+\infty} f(\lambda) dE_{\lambda}(x) \quad (2.4)$$

è definito come il limite in \mathcal{X} , se esiste, di $\int_a^b f(\lambda) dE_{\lambda}(x)$ per $a \rightarrow -\infty$ e $b \rightarrow \infty$.

Per arrivare ad un risultato di convergenza dell'integrale con estremi non limitati, si utilizzano alcune proprietà delle famiglie spettrali.

Proposizione 2.1. Sia $\{E_{\lambda}\}_{\lambda \in \mathbb{R}}$ una famiglia spettrale. Allora valgono;

- (i) $\forall \lambda, \mu \in \mathbb{R}, \mu \leq \lambda$, l'operatore $E_{[\mu, \lambda]} := E_{\lambda} - E_{\mu}$ è ancora un proiettore ortogonale;
- (ii) $\forall \lambda, \mu, \lambda', \mu' \in \mathbb{R}, \mu \leq \lambda$ e $\mu' \leq \lambda'$ si ha $E_{[\mu, \lambda]} E_{[\mu', \lambda']} = E_{[\mu, \lambda] \cap [\mu', \lambda']}$
- (iii) Siano $\lambda_1 < \dots < \lambda_m$ e $\alpha_1, \dots, \alpha_{m-1} \in \mathbb{C}$. Allora per ogni $x \in \mathcal{X}$

$$\left\| \sum_{k=1}^{m-1} \alpha_k E_{[\lambda_{k-1}, \lambda_k]}(x) \right\|^2 = \sum_{k=1}^{m-1} |\alpha_k|^2 \left\| E_{[\lambda_{k-1}, \lambda_k]}(x) \right\|^2 \leq \sup_k |\alpha_k|^2 \|x\|^2.$$

Il seguente risultato dà una proprietà di continuità a sinistra, utile per definire il risultato di convergenza che si vuole enunciare.

Proposizione 2.2. Per ogni $\lambda \in \mathbb{R}$ e per ogni $x \in \mathcal{X}$, il limite

$$E_{\lambda-0}(x) := \lim_{\mu \rightarrow \lambda^-} E_{\mu}(x)$$

esiste in \mathcal{X} ed è un proiettore ortogonale su \mathcal{X} .

Sia ora $x \in \mathcal{X}$, si ponga $F(x) := \int_{]a, b]} f(\lambda) dE_{\lambda}(x)$ con f continua, reale a valori reali, $]a, b] \subset \mathbb{R}$. Allora si ha

1. $F(x) \in \text{Im}(E_{]a, b]});$

2. $\forall x_1 \in \mathcal{X}$

$$\langle F(x), x_1 \rangle = \int_{]a,b]} f(\lambda) d \langle E_\lambda(x), x_1 \rangle$$

dove il prodotto scalare all'interno dell'integrale è definito come somma di Riemann;

$$3. \langle F(x), x \rangle = \int_{]a,b]} f(\lambda) d \|E_\lambda(x)\|^2 = \int_{]a,b]} f(\lambda) \|dE_\lambda(x)\|^2;$$

4. per ogni $a' < b'$, per ogni $g \in \mathcal{C}([a', b'])$, vale

$$\int_{]a',b']} g(\lambda) dE_\lambda(F(x)) = \int_{]a,b] \cap]a',b']} fg(\lambda) dE_\lambda(x)$$

$$5. \|F(x)\|^2 = \int_{]a,b]} \|f(\lambda)\|^2 \|dE_\lambda(x)\|^2 = \int_{]a,b]} \|f(\lambda)\|^2 d \|E_\lambda(x)\|^2.$$

Si osservi ora che la condizione (i) nella definizione della famiglia spettrale dice che $\langle E_\lambda(x), x \rangle \leq \langle E_\mu(x), x \rangle$ per ogni x e per ogni $\mu \leq \lambda$. Allora la funzione $\lambda \mapsto \langle E_\lambda x, x \rangle = \|E_\lambda(x)\|^2$ definisce una misura su \mathbb{R} , grazie alla proposizione (2.2) e alla proprietà (3) precedente.

Si può enunciare quindi il seguente

Teorema 2.3. Sia $x \in \mathcal{X}$ e $f : \mathbb{R} \rightarrow \mathbb{C}$ continua. Allora

$$\int_{\mathbb{R}} f(\lambda) dE_\lambda(x) < +\infty$$

se e soltanto se

$$\int_{\mathbb{R}} |f(\lambda)|^2 d \|E_\lambda(x)\|^2 < +\infty$$

In particolare, se f è limitata su \mathbb{R} , allora l'integrale esiste sempre.

Ora si mostrerà il collegamento fra operatori autoaggiunti e famiglie spettrali.

Def. 2.8. Sia A un operatore lineare su uno spazio di Hilbert, di dominio $\mathcal{D}(A)$ denso in \mathcal{X} . Si pone $\mathcal{D}^*(A)$ il sottospazio di \mathcal{X} definito nel seguente modo:

$$u \in \mathcal{D}^*(A) \Leftrightarrow \exists C_u > 0 \text{ t.c. } \forall v \in \mathcal{D}(A) \|\langle u, Av \rangle\| \leq C_u \|v\|$$

L'operatore A^* si dice *aggiunto* di A se $\langle u, Av \rangle = \langle A^*u, v \rangle$ per ogni $u, v \in \mathcal{X}$. L'operatore A si dice *autoaggiunto* se e solo se vale l'uguaglianza $(A, \mathcal{D}(A)) = (A^*, \mathcal{D}^*(A))$.

Per gli operatori autoaggiunti si ha il seguente notevole risultato.

Teorema 2.4 (Teorema spettrale). Sia \mathcal{X} spazio di Hilbert, A un operatore autoaggiunto su \mathcal{X} . Allora esiste ed è unica la famiglia spettrale $\{E_\lambda\}_{\lambda \in \mathbb{R}}$ tale che

$$A = \int \lambda dE_\lambda \tag{2.5}$$

Il dominio dell'operatore A è $\mathcal{D}(A) = \left\{ x \in \mathcal{X} \mid \int \lambda^2 d \|E_\lambda(x)\|^2 < +\infty \right\}$

La (2.5) prende il nome di *decomposizione spettrale* di A . Si possono definire inoltre anche funzioni dell'operatore A , ponendo $f(A) = \int f(\lambda)dE_\lambda$, con f continua su \mathbb{R} , ridefinendo opportunamente il dominio.

Esistono delle importanti connessioni fra lo spettro dell'operatore A e la sua famiglia spettrale

Proposizione 2.3. Sia A un operatore autoaggiunto su \mathcal{X} spazio di Hilbert, e sia $\{E_\lambda\}_{\lambda \in \mathbb{R}}$ la sua famiglia spettrale. Allora valgono:

- $\lambda_0 \in \sigma(A)$ sse $E_{\lambda_0} \neq E_{\lambda_0+\epsilon}$ per ogni $\epsilon > 0$;
- λ_0 è un autovalore di A se e soltanto se $E_{\lambda_0} \neq E_{\lambda_0+0} = \lim_{\epsilon \rightarrow 0} E_{\lambda_0+\epsilon}$

Si considerino ora due casi particolari.

Sia A operatore autoaggiunto definito strettamente positivo, soddisfacente l'uguaglianza $\|Ax\| \geq \gamma\|x\|$ per ogni $x \in \mathcal{D}(A)$, $\gamma > 0$. Allora, per ogni $f \in \mathcal{M}_0 := \{f \in \mathcal{C}(\mathbb{R}) \mid \text{misurabile rispetto alla misura } d\|E_\lambda(x)\|^2 \ \forall x \in \mathcal{X}\}$ si ha che

$$\int_{\mathbb{R}} f(\lambda)dE_\lambda(x) = \int_{\gamma}^{\infty} f(\lambda)dE_\lambda(x)$$

Quindi l'integrazione può essere ristretta all'intervallo $[\gamma, +\infty)$.

Sia ora $T : \mathcal{X} \rightarrow \mathcal{Y}$ lineare e limitato, si consideri $A := T * T$, che risulterà autoaggiunto. Si ha, per ogni $f \in \mathcal{M}_0$

$$\int_{\mathbb{R}} f(\lambda)dE_\lambda(x) = \int_0^{\|T\|^2} f(\lambda)dE_\lambda(x) = \lim_{\epsilon \rightarrow 0} \int_0^{\|T\|^2+\epsilon} f(\lambda)dE_\lambda(x) \quad (2.6)$$

Quindi, ancora una volta la funzione può essere ristretta all'intervallo $[0, \|T\|^2 + \epsilon]$ per un qualche $\epsilon > 0$.

Si riconsideri il caso di un operatore compatto (definizione (2.9)): le uguaglianze scritte in (2.6) rimangono le stesse se al posto di T viene posto K operatore compatto. Quindi è possibile scrivere

$$K^*K(x) = \int_0^{\|K\|^2} \lambda dE_\lambda(x)$$

con $\{E_\lambda\}_{\lambda \in \mathbb{R}}$ famiglia spettrale associata a K^*K , $f = Id$. L'interesse consiste nel risolvere l'equazione normale

$$K^*K(x) = K^*(y)$$

cioè nell'invertire K^*K . Ricordando la proprietà (3), si può scrivere

$$(K^*K)^{-1} = \int_0^{\|K\|^2} \frac{1}{\lambda} dE_\lambda(x)$$

Il problema sorge quindi in questo integrale, perchè vicino a zero la funzione λ^{-1} esplose all'infinito. È necessario trovare quindi dei metodi che consentano il controllo vicino a zero dell'integrale soprascritto, sostituendo a λ^{-1} una funzione $g(\alpha)$ che assicuri la convergenza dell'integrale, e tale che $g(\alpha) \rightarrow \lambda^{-1}$ per $\alpha \rightarrow 0$.

2.3 Operatori Compatti e Sistemi Singolari

Nelle applicazioni, i problemi maggiori di mal posizione vengono da operatori compatti.

Def. 2.9. Un operatore limitato $K : \mathcal{X} \rightarrow \mathcal{Y}$ si dice *compatto* se e soltanto se l'immagine di un qualsiasi insieme compatto di \mathcal{X} è relativamente compatto⁴ in \mathcal{Y} .

Ad esempio, l'operatore $K(f)$ visto nel precedente capitolo dato dall'equazione integrale di prima specie di Fredholm

$$K(f)(x) = \int_{\Omega} k(x, y)f(y)dy$$

è compatto, perchè si era posto $\int_{\Omega} k^2(x, y)dxdy < \infty$. Nel caso dell'acquisizione di immagini, quest'ultimo integrale vale 1.

Il seguente teorema dà l'importante connessione tra operatori compatti e mal posizione per operatori lineari in spazi infinito dimensionali.

Teorema 2.5. Sia $K : \mathcal{X} \rightarrow \mathcal{Y}$ un operatore lineare compatto fra spazi di Hilbert infinito-dimensionali. Se $\dim(\text{Range}(K)) = \infty$, allora l'equazione $K(x) = y$ è mal posta, nel senso che le condizioni (i) e (iii) della definizione (2.1) sono violate. In questo caso, $\text{Range}(K)$ non è chiuso; se $\text{Range}(K)$ ha dimensione finita, allora la condizione (ii) non è soddisfatta.

Si consideri il caso di un operatore compatto autoaggiunto K : esso possiede un *autosistema* $(\lambda_n, v_n)_{n \in \mathbb{N}}$ dove i λ_n sono gli *autovalori* (diversi da zero) e gli *autovettori* v_n formano un sistema ortonormale completo. In questo caso, l'operatore si dice diagonalizzabile e può essere scritto

$$Kx = \sum_{n=1}^{\infty} \lambda_n \langle x, v_n \rangle v_n \quad (2.7)$$

per ogni $x \in \mathcal{X}$. Si noti l'analogia fra quest'ultima formula e

$$Kx = \int_{\mathbb{R}} \lambda dE_{\lambda}(x)$$

con $\{E_{\lambda}\}_{\lambda \in \mathbb{R}}$ famiglia spettrale associata a K .

Sorgono problemi quando l'operatore non è autoaggiunto. Si può aggirare questo fatto osservando le connessioni tra l'equazione normale (2.2) e $K(x) = y$: si può definire un sistema che funge da 'sostituto' per l'autosistema. Come già osservato, se K è compatto allora K^*K è autoaggiunto.

Def. 2.10. Un sistema singolare per un operatore lineare compatto K è un insieme numerabile costituito dalle triplette $(u_j, \sigma_j, v_j)_j$, dove

- (i) i vettori v_j sono detti *vettori destri singolari* e formano una base ortonormale completa per $\text{Ker}(K)^{\perp}$;

⁴Un insieme E si dice relativamente compatto in Ω , Ω aperto, se vale $E \subset \overline{E} \subset \Omega$.

- (ii) i vettori u_j sono detti *vettori singolari sinistri* e formano una base ortonormale completa per $\overline{\text{Range}(K)}$;
- (iii) i valori σ_j sono detti *valori singolari*, sono reali positivi e in ordine crescente $\sigma_1 \geq \sigma_2 \geq \dots \geq 0$.

Si può notare che i valori σ_n^2 sono in realtà gli autovalori dell'operatore autoaggiunto K^*K , i v_n sono i corrispondenti autovettori che formano una base completa ortonormale di $\overline{\text{Range}(K^*K)}$. I vettori u_n vengono definiti come

$$u_n := \frac{K(v_n)}{\|K(v_n)\|}$$

e risultano essere gli autovettori di KK^* ; essi formano una base ortonormale completa per $\overline{\text{Range}(KK^*)}$.

Valgono inoltre le seguenti proprietà:

$$\begin{aligned} K v_n &= \sigma_n u_n \\ K^* u_n &= \sigma_n v_n \\ K x &= \sum_{n=1}^{\infty} \sigma_n \langle x, v_n \rangle u_n, \quad x \in \mathcal{X} \\ K^* y &= \sum_{n=1}^{\infty} \sigma_n \langle y, u_n \rangle v_n, \quad y \in \mathcal{Y} \end{aligned}$$

dove le serie convergono nello spazio di Hilbert in questione. Se $\text{Range}(K)$ è infinito-dimensionale, si ha

$$\lim_{j \rightarrow \infty} \sigma_j = 0$$

Una scomposizione di questo genere prende il nome di *espansione ai valori singolari*.

Si riconsideri il problema originario: si ha che $\text{Range}(K)$ è chiuso se e soltanto se ha dimensione finita. Quindi, nel caso generale di dimensione infinita (cioè di infiniti valori singolari), $\text{Range}(K)$ non è chiuso. Ricordando il teorema (2.5), si ha il seguente risultato:

Proposizione 2.4. Sia $K : \mathcal{X} \rightarrow \mathcal{Y}$, $\dim(\text{Range}(K)) = \infty$. Allora K^\dagger è un operatore compatto non limitato con grafico chiuso.

Con quest'ultima proposizione, si ha che per un operatore con codominio non chiuso la migliore soluzione approssimante non dipende continuamente dai dati: il problema cioè è mal posto. Infatti, tentando di scrivere esplicitamente la soluzione si ottiene:

$$x = \sum_{k=1}^{\infty} \frac{\langle y, u_k \rangle_{\mathcal{Y}} v_k}{\sigma_k}$$

ma per k tendente a infinito si ha che $\sigma_k \rightarrow 0$, quindi la serie esplose all'infinito. Si riconosca ancora l'analogia con quanto succede quando si cerca di scrivere l'inverso di un operatore compatto e autoaggiunto la famiglia spettrale ad esso

associata.

Avendo un sistema singolare per un operatore lineare e compatto, allora è possibile scomporre in serie l'inverso generalizzato di Moore-Penrose, come si enuncia nel seguente teorema.

Teorema 2.6. Sia $\{\sigma_n, u_n, v_n\}_{n \in \mathbb{N}}$ un sistema singolare per l'operatore lineare compatto K ; sia $y \in \mathcal{Y}$. Allora

$$(i) \quad y \in \mathcal{D}(K^\dagger) \Leftrightarrow \sum_{k=1}^{\infty} \frac{|\langle y, u_n \rangle|^2}{\sigma_n^2} < \infty$$

(ii) per $y \in \mathcal{D}(K^\dagger)$

$$K^\dagger(y) = \sum_{k=1}^{\infty} \frac{\langle y, u_n \rangle v_n}{\sigma_n}$$

La condizione (i) nel precedente teorema prende il nome di *criterio di Picard*: esso dice che la miglior soluzione approssimata per $K(x) = y$ esiste se e soltanto se i coefficienti⁵ $\langle y, u_n \rangle$ vanno a zero più velocemente rispetto ai valori singolari σ_n .

2.4 Operatori di Regolarizzazione

Il problema che si pone è risolvere l'equazione

$$Kx = y$$

quando si ha a disposizione non il dato esatto y ma una sua perturbazione y_δ , tale che

$$\|y - y_\delta\| \leq \delta$$

Il dato y_δ prende il nome di *dato perturbato* o *rumoroso* e δ invece prende il nome di *rumore* o *livello di rumore*. Si cerca quindi la migliore approssimazione di x^\dagger tale che $x^\dagger = K^\dagger y$, avendo a disposizione l'operatore⁶ K , il rumore δ e il dato perturbato. Nel caso malposto, $K^\dagger(y_\delta)$ non è una buona approssimazione di $K^\dagger y$ a causa della non limitatezza di K^\dagger , anche nel caso in cui esso esista.

Si cerca quindi un'approssimazione $x_{\delta, \alpha}$ di x^\dagger , dove il parametro α prende il nome di *parametro di regolarizzazione*. Si vuole avere $x_{\delta, \alpha}$ tale che dipenda continuamente dai dati y_δ , in modo da poter essere calcolata in maniera stabile, ed inoltre tenda a x^\dagger per $\delta \rightarrow 0$ e per una scelta di α appropriata. Si pone quindi la seguente definizione:

Def. 2.11. Sia $\alpha_0 \in (0, +\infty]$; per ogni $\alpha \in I := (0, \alpha_0)$, sia l'operatore

$$R_\alpha : \mathcal{Y} \rightarrow \mathcal{X}$$

continuo (non necessariamente lineare). La famiglia $\{R_\alpha\}_{\alpha \in I}$ è detta *famiglia di operatori regolarizzanti* per K^\dagger se per ogni $y \in \mathcal{D}(K^\dagger)$ esiste una funzione

⁵In letteratura, prendono il nome di coefficienti generalizzati di Fourier.

⁶Si suppone che l'operatore K sia a.a., altrimenti si considera l'operatore $T = K^*K$.

$$\underline{\alpha} : (0, +\infty) \times \mathcal{Y} \rightarrow (0, \alpha_0) \quad (2.8)$$

tale che soddisfi le seguenti condizioni:

$$\lim_{\delta \rightarrow 0} \sup_{y_\delta \in \mathcal{B}} \underline{\alpha}(\alpha, y_\delta) = 0$$

$$\lim_{\delta \rightarrow 0} \sup_{y_\delta \in \mathcal{B}} \|R_{\underline{\alpha}(\alpha, y_\delta)}(y_\delta) - K^\dagger(y)\| = 0$$

dove $\mathcal{B} = \overline{\mathcal{B}(y, \delta)}$. La funzione $\underline{\alpha}$ è detta *scelta del parametro*. Per $y \in \mathcal{D}(\mathcal{Y})$, la coppia $(R_\alpha, \underline{\alpha})$ è detta *metodo di regolarizzazione*; se $\underline{\alpha}$ soddisfa le due condizioni sopra, il metodo è *convergente*.

Quindi si cerca di trovare una successione di operatori R_α tali che approssimino la pseudo inversa K^\dagger , dove per ogni dato perturbato si va a scegliere un parametro di regolarizzazione α dato da $\underline{\alpha}$. Le condizioni poste sulla funzione $\underline{\alpha}$ soddisfano le richieste sull'approssimazione cercata per x^\dagger .

Si pone inoltre la seguente definizione.

Def. 2.12. Sia $\underline{\alpha}$ una scelta del parametro come nella definizione (2.11). Se $\underline{\alpha}$ dipende esclusivamente dal livello di rumore δ e non dal dato perturbato y_δ , essa prende il nome di scelta del parametro *a priori*. In caso contrario, prende il nome di scelta del parametro *a posteriori*.

Si potrebbe quindi pensare di creare un metodo di regolarizzazione indipendente dal livello di rumore presente, ma il seguente risultato impedisce la costruzione di un metodo siffatto.

Teorema 2.7 (Bakushinskii). Sia $\{R_\alpha\}$ una famiglia di operatori regolarizzanti per T^\dagger tale che per ogni $y \in \mathcal{D}(\mathcal{Y})$ esista una scelta del parametro $\underline{\alpha}$ che dipenda solo da y_δ . Allora K^\dagger è limitato.

Quindi, nel caso malposto non è possibile trovare una scelta del parametro indipendente dal rumore. Non è detto però che una scelta euristica del parametro non conduca ad una buona approssimazione per x^\dagger .

La via per capire come un metodo di regolarizzazione converge alla soluzione cercata è passare per la convergenza puntuale: i risultati riguardanti questo tipo di convergenza vengono mostrati nelle prossime proposizioni.

Proposizione 2.5. Sia $\{R_\alpha\}$ una famiglia di operatori che converge puntualmente a K^\dagger per ogni $y \in \mathcal{D}(K^\dagger)$ per $\alpha \rightarrow 0$. Allora $\{R_\alpha\}$ è un metodo di regolarizzazione per K^\dagger e per ogni $y \in \mathcal{D}(K^\dagger)$ esiste una scelta del parametro a priori tale che $(\{R_\alpha\}, \underline{\alpha})$ è un metodo convergente per risolvere $Kx = y$. Analogamente, se $(\{R_\alpha\}, \underline{\alpha})$ è un metodo convergente per risolvere $Kx = y$ con $y \in \mathcal{D}(K^\dagger)$ e $\underline{\alpha}$ è continuo rispetto a δ , allora $R_\alpha(y)$ converge a $K^\dagger(y)$ per $\alpha \rightarrow 0$.

Quindi, una famiglia di operatori regolarizzanti è un'approssimazione puntuale per l'inversa generalizzata di Monre-Penrose.

Un ulteriore risultato si ha per una famiglia di operatori lineari.

Lemma 2.1. Sia $\{R_\alpha\}$ una famiglia di operatori di regolarizzazione lineari. $\forall y \in \mathcal{D}(K^\dagger)$ sia $\underline{\alpha} : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ una funzione indipendente da y_δ . Allora $\underline{\alpha}$ è una scelta del parametro a priori se e soltanto se valgono

$$\lim_{\delta \rightarrow 0} \underline{\alpha}(\delta) = 0$$

$$\lim_{\delta \rightarrow 0} \delta \|R_{\alpha(\delta)}\| = 0$$

Proposizione 2.6. Sia $\{R_\alpha\}$ un metodo di regolarizzazione lineare, e sia $x_\alpha := R_\alpha(y)$ per ogni $y \in \mathcal{Y}$. Allora $\{x_\alpha\}$ converge a $K^\dagger(y)$ per ogni $y \in \mathcal{D}(K^\dagger)$; se $\underline{\alpha}$ (la scelta del parametro associata alla famiglia di operatori) è continua rispetto a δ e se vale

$$\sup_{\alpha > 0} |KR_\alpha| < \infty$$

allora $\lim_{\alpha \rightarrow 0} |x_\alpha| = +\infty$ per $y \notin \mathcal{D}(K^\dagger)$.

La scelta del parametro per una famiglia di operatori lineari descritti nella proposizione appena esposta è caratterizzata dal lemma precedente.

2.5 Regolarizzazione nel Discreto

Si consideri ora l'equazione

$$K\mathbf{f} = \mathbf{g}$$

nel caso discreto, dove cioè \mathbf{f}, \mathbf{g} sono vettori di dimensione N (matrici opportunamente vettorizzate rappresentanti delle immagini digitali) e K è un operatore compatto (in \mathbf{g} è considerato anche l'eventuale errore additivo). Essendo l'operatore compatto, per trovare una soluzione al problema si considera il funzionale

$$\begin{aligned} T_\alpha : \mathcal{X} \times \mathcal{Y} &\rightarrow \mathbb{R} \\ T_\alpha : (\mathbf{f}, \mathbf{g}) &\mapsto \rho(K\mathbf{f}, \mathbf{g}) + \alpha J(\mathbf{f}) \end{aligned} \tag{2.9}$$

tra i due spazi di Hilbert $\mathcal{X} = \mathcal{Y} = \mathbb{R}^N$ e con $\alpha \in \mathbb{R}$, si cerca un punto di minimo per T_α . Questo significa cercare un'immagine che si discosti il meno possibile dall'immagine reale che dà come risultato il dato \mathbf{g} . Si pone la seguente

Def. 2.13. Sia dato un funzionale come in (2.9); α è il *parametro di regolarizzazione*, J prende il nome di *penalty functional* mentre ρ di *data discrepancy functional* o *fit-to-data functional*.

Lo scopo del funzionale ρ è di quantificare quanto la predizione $K\mathbf{f}_\alpha$ approssima il dato osservato \mathbf{g} ; il compito del funzionale J è di dare stabilità al metodo e inoltre permettere di aggiungere informazioni note sulla soluzione f . La coppia $(T_\alpha, \underline{\alpha})$ è un **metodo di regolarizzazione** del tipo descritto nella sezione precedente quando $\underline{\alpha}$ è un metodo di regolarizzazione.

Alcuni esempi di fit-to-data functionals sono i seguenti

- la norma al quadrato in uno spazio di Hilbert \mathcal{H}

$$\rho_{ls}(g_1, g_2) = \frac{1}{2} \|g_1 - g_2\|_{\mathcal{H}}^2, \quad g_1, g_2 \in \mathcal{H}$$

che porta alla ricerca di una soluzione ai minimi quadrati;

- la divergenza di Kullbach-Leibler

$$\rho_{KL}(g_1, g_2) = \langle g_1, \log \left(\frac{g_1}{g_2} \right) \rangle_{\mathcal{H}}, \quad g_1, g_2 \in \mathcal{C}$$

dove \mathcal{C} è chiuso e convesso. Viene utilizzato nel caso di rumore poissoniano, principalmente;

- la log-verosimiglianza negativa di Poisson

$$\rho_{LHD}(g_1, g_2) = \langle g_1, \mathbf{1} \rangle_{\mathcal{H}} - \langle g_2, \log(g_1) \rangle_{\mathcal{H}}, \quad g_1, g_2 \in \mathcal{C}$$

dove $\mathbf{1}$ denota il vettore unità.

2.5.1 SVD di una Matrice: metodo di Tikhonov

Sia $K \in \mathcal{M}_{m \times n}(\mathbb{R})$: essa si può vedere come un operatore lineare compatto che mappa $\mathcal{X} = \mathbb{R}^n$ in $\mathcal{Y} = \mathbb{R}^m$. Essa ammette una *scomposizione ai valori singolari*⁷

$$\{u_n, s_n, v_n\}_{n=1}^r$$

dove r è il rango della matrice, cioè la dimensione di $Range(K)$. Si denoti con $U \in \mathcal{M}_{m \times r}(\mathbb{R})$ la matrice in cui la j -esima colonna è data dal vettore u_j , mentre con $V \in \mathcal{M}_{r \times n}(\mathbb{R})$ si denota la matrice in cui la j -esima colonna è data dal vettore v_j ; sia Σ la matrice di dimensioni $m \times n$ della forma

$$\Sigma = \begin{pmatrix} diag(s_1, \dots, s_r) & 0 \\ 0 & 0 \end{pmatrix}$$

Inoltre, si ha che $U^T = U^{-1}$ e $V^T = V^{-1}$, sono cioè matrici ortogonali. La matrice originaria si può scrivere

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

Nel caso complesso, U e V sono unitarie e Σ ha rango massimo: in questo elaborato si considererà solo il caso reale in quanto i dati digitali saranno sempre reali.

É possibile allora scrivere l'operazione matrice-vettore nel seguente modo:

$$K\mathbf{f} = (U\Sigma V^T)\mathbf{f} = \sum_j s_j \langle \mathbf{f}, v_j \rangle u_j \quad (2.10)$$

Tornando al caso delle immagini digitali, con la notazione $\mathbf{f} = \mathbf{vec}(f)$, $\mathbf{g} = \mathbf{vec}(g)$, il modello matematico assumerà la forma

$$\mathbf{g} = K\mathbf{f} + \eta = \sum_j s_j \langle \mathbf{f}, v_j \rangle u_j + \eta \quad (2.11)$$

⁷Single Value Decomposition, SVD.

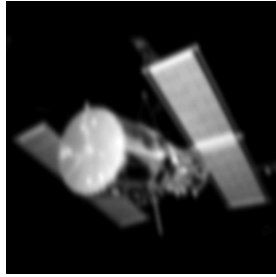
e utilizzando le proprietà della SVD si può calcolare

$$K^{-1}\mathbf{g} = \mathbf{f} + \sum_j \frac{\langle u_j^T, \boldsymbol{\eta} \rangle v_j}{s_j}$$

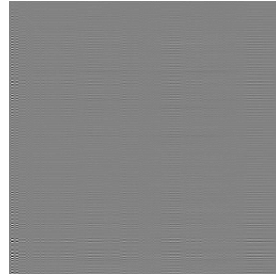
Il problema nasce nel secondo addendo nel membro sinistro dell'equazione: infatti la condizione di Picard non è soddisfatta: al crescere di j , i valori singolari si accumulano sempre più a zero, facendo esplodere i coefficienti della sommatoria

$$\lim_{j \rightarrow \infty} \frac{\langle u_j^T, \boldsymbol{\eta} \rangle v_j}{s_j} = +\infty$$

e quindi i le componenti del rumore prendono il sopravvento sull'immagine ricostruita.



(a) Immagine perturbata.



(b) Immagine ricostruita tramite SVD.

Figura 2.1: Ricostruzione tramite l'inversione della SVD. L'immagine blurred ha un livello di errore pari a $1e-16$, ma si nota che una perturbazione così piccola influisce in maniera massiccia sulla soluzione trovata.

Un primo tentativo per regolarizzare la soluzione è prendere solo un numero prefissato degli addendi nella sommatoria $\sum_j \frac{\langle u_j^T, \boldsymbol{\eta} \rangle v_j}{s_j}$: questo metodo prende il nome di TSVD, Truncated Single Value Decomposition.

Un'altra strada è prendere una funzione dei valori singolari g , dipendente da un parametro α , tale che $g(\alpha) \rightarrow s_i^{-1}$ per ogni s_i . Una scelta di questo tipo è

$$g(\alpha) = \frac{s_i}{s_i^2 + \alpha}$$

e quindi, andando a sostituire nella formula risolutiva si ha

$$\mathbf{f}_\alpha = \sum_j \frac{s_j}{s_j^2 + \alpha} \langle u_j^T, \mathbf{g} \rangle v_j$$

dove con \mathbf{f}_α si indica la soluzione regolarizzata.

Utilizzando le proprietà dei vettori singolari destri e sinistri si può scrivere

$$\mathbf{f}_\alpha = (K^T K + \alpha Id)^{-1} K^T \mathbf{d} \quad (2.12)$$

L'operatore che appare in (2.12) fa parte di una famiglia di operatori regolarizzanti $\{R_\alpha\}$ del tipo descritto prima, e il parametro α può essere scelto tramite varie funzioni di scelta dei parametri, come ad esempio il metodo GCV (Generalized Cross Validation), il principio di discrepanza, il metodo L-curva e molti altri⁸. La funzione di scelta del parametro va presa ad hoc, e dipende dal problema e dal tipo di immagine che si sta considerando. Il metodo appena descritto prende il nome di *metodo di Tikhonov*.

La soluzione \mathbf{f}_α scritta in (2.12) equivale alla soluzione del problema

$$\min_f \frac{1}{2} \|K\mathbf{f} - \mathbf{g}\|^2 + \alpha \|A(\mathbf{f})\|^2$$

con $A = Id$; si noti che in questo caso $\rho = \rho_{ls}$ e $J(\cdot) = \|\cdot\|^2$.

2.5.2 Total Variation

In questa sezione si indicheranno con $\Omega_{[a,b]}$ l'insieme della partizioni σ dell'intervallo $[a, b]$, dove $\sigma = \{a = x_0, x_1, \dots, x_{n-1}, x_n = b\}$.

Def. 2.14. Sia $f : [a, b] \rightarrow \mathbb{R}$; si pone

$$\bigvee_a^b(f) = \sup_{\sigma \in \Omega_{[a,b]}} \sum_{k=1}^p |f(x_k) - f(x_{k-1})| \quad (2.13)$$

Se vale $\bigvee_a^b(f) < \infty$, allora la funzione f si dice a *variazione limitata* e $\bigvee_a^b(f)$ è la *variazione totale* di f su $[a, b]$.

Si espongono i seguenti risultati⁹ sulle funzioni a variazione limitata.

Teorema 2.8 (di Jordan). Una funzione $f : [a, b] \rightarrow \mathbb{R}$ è a variazione limitata su $[a, b]$ se e soltanto se esistono due funzioni ϕ_1, ϕ_2 monotone crescenti che mappano $[a, b]$ in \mathbb{R} tali che

$$f = \phi_1 - \phi_2$$

Teorema 2.9. Sia $f \in \mathcal{C}^1([a, b])$; allora

$$\bigvee_a^b(f) = \int_a^b |f'(x)| dx$$

Per funzioni continue con derivata continua la variazione totale vale la lunghezza della curva definita dal grafico della funzione calcolata nell'intervallo $[a, b]$ (fig. 2.2).

⁸Per una discussione più dettagliata, si possono consultare [CV], [EHN]

⁹Per una dimostrazione, si consulti [EL].

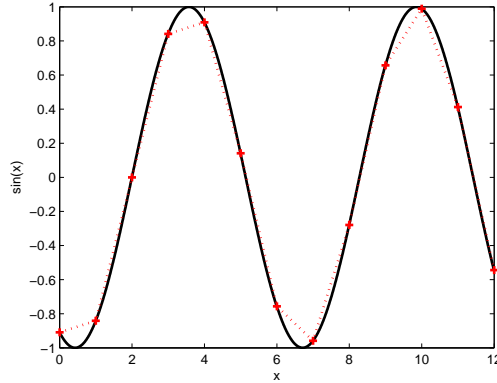


Figura 2.2: $y = \sin(x)$ nell'intervallo $[0, 12]$.

Nel caso bidimensionale (che è quello di interesse) considerando l'intervallo $\Omega := [a, b] \times [c, d] \subset \mathbb{R}^2$, se $f : \Omega \rightarrow \mathbb{R}$ è continua con derivata continua, allora la definizione viene estesa nel modo seguente:

$$\mathbf{TV}(f) = \int_{\Omega} |\nabla f(x, y)| dx dy$$

Una generalizzazione per il caso in cui f non sia liscia è la seguente

$$\mathbf{TV}(f) = \sup_{\nu \in \mathcal{V}} \int_{\Omega} f(x, y) \operatorname{div} \vec{\nu} dx dy$$

dove \mathcal{V} è l'insieme delle normali esterne ad Ω .

La variazione totale di una funzione f di due variabili reali può essere vista come la misura della superficie laterale del grafico di f . Nel caso di ricostruzione delle immagini il funzionale correlato alla $\mathbf{TV}(f)$ è

$$T_{\alpha}(\mathbf{f}) = \frac{1}{2} \|K(\mathbf{f}) - \mathbf{g}\|^2 + \alpha \mathbf{TV}(\mathbf{f}) \quad (2.14)$$

quindi se sono presenti forti oscillazioni in intensità nell'immagine, allora il funzionale \mathbf{TV} avrà un forte peso sulla soluzione regolarizzata; tende quindi a fornire una buona ricostruzione per immagini con contorni ben definiti e grosse variazioni di intensità.

Si hanno i seguenti risultati¹⁰ di convergenza, unicità e stabilità per il funzionale descritto in (2.14).

Teorema 2.10. Sia \mathcal{C} un sottoinsieme chiuso e convesso di $\mathcal{L}^p(\Omega)$, $\Omega \subset \mathbb{R}^d$ e $1 \leq p < \frac{d}{d-1}$. Sia $K : \mathcal{L}^p(\Omega) \rightarrow \mathcal{L}^2(\Omega)$ lineare e limitato, tale che $K(\mathbf{1}) \neq 0$. Allora il funzionale in (2.14) ha un unico minimo su \mathcal{C} .

Teorema 2.11. Le soluzioni regolarizzate con il metodo della variazione totale sono stabili rispetto ad alcune perturbazioni nel penalty functional in (2.14).

¹⁰Per le dimostrazioni, si consulti [CV].

Le soluzioni per l'equazione $K(\mathbf{f}) = \mathbf{d}$ allora si possono ottenere minimizzando il funzionale (2.14): sorge un problema quando però si vanno ad applicare i metodi di ottimizzazione descritti prima, in quanto la norma euclidea che appare nella definizione non è differenziabile attorno all'origine. Per superare questa difficoltà, si considera un'approssimazione di tale norma: $|x| \sim \sqrt{x^2 + \beta^2}$, con β positivo. Quindi l'approssimazione bidimensionale per il funzionale di total variation assume la forma

$$\mathbf{TV}_\beta(f) = \int_\Omega \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 + \beta^2} dx dy$$

Il funzionale da minimizzare sarà quindi

$$T_\alpha(\mathbf{f}) = \frac{1}{2} \|K(\mathbf{f}) - \mathbf{g}\|^2 + \alpha \mathbf{TV}_\beta(\mathbf{f}) \quad (2.15)$$

La discretizzazione di questo funzionale avviene nel seguente modo: considerando nel modo usuale \mathbf{d} i dati discreti in forma matriciale e K la discretizzazione della PSF, si definisce il seguente penalty functional

$$J(\mathbf{f}) = \frac{1}{2} \sum_i \sum_j \psi \left((D_{i,j}^x f)^2 + (D_{i,j}^y f)^2 \right) \quad (2.16)$$

dove¹¹

$$D_{i,j}^x f = \frac{f_{i,j} - f_{i-1,j}}{\Delta x}, \quad D_{i,j}^y f = \frac{f_{i,j} - f_{i,j-1}}{\Delta y}$$

e ψ è un'approssimazione del doppio della radice quadrata tale che $\psi'(t) > 0$ per $t > 0$, del tipo descritto sopra. Per semplicità di notazione, in (2.16) si è ommesso una fattore $\Delta x \Delta y$ che si può inglobare nel parametro di regolarizzazione α .

Andando a calcolare il gradiente¹² di J , si ottiene

$$\nabla J = L(\mathbf{f})\mathbf{f}$$

dove

$$L(\mathbf{f}) = \begin{bmatrix} D_x^T & D_y^T \end{bmatrix} \begin{bmatrix} \text{diag}(\psi'(\mathbf{f})) & \mathbf{0} \\ \mathbf{0} & \text{diag}(\psi'(\mathbf{f})) \end{bmatrix} \begin{bmatrix} D_x \\ D_y \end{bmatrix} \quad (2.17)$$

con D_x e D_y matrici che denotano gli operatori di derivazione.

Forti del gradiente, è possibile calcolare la matrice Hessiana del funzionale, data da

$$L(\mathbf{f}) + L'(\mathbf{f})\mathbf{f} \quad (2.18)$$

L'espressione del secondo addendo di (2.18) è

$$L'(\mathbf{f})\mathbf{f} = \begin{bmatrix} D_x^T & D_y^T \end{bmatrix} \begin{bmatrix} \text{diag}(2(D_x \mathbf{f})^2 \psi''(\mathbf{f})) & \text{diag}(2(D_x \mathbf{f})(D_y \mathbf{f}) \psi''(\mathbf{f})) \\ \text{diag}(2(D_y \mathbf{f})(D_x \mathbf{f}) \psi''(\mathbf{f})) & \text{diag}(2(D_y \mathbf{f})^2 \psi''(\mathbf{f})) \end{bmatrix} \begin{bmatrix} D_x \\ D_y \end{bmatrix}$$

¹¹Si possono prendere anche differenze all'avanti o all'indietro.

¹² $\frac{d}{d\tau} J(f + \tau v)|_{\tau=0}$.

Riunendo il tutto, si ha a disposizione il gradiente e l'Hessiana del funzionale (2.15) e si possono quindi applicare i metodi di ottimizzazione descritti nelle sezioni precedenti.

$$\begin{aligned}\nabla \mathbf{TV}_\beta(\mathbf{f}) &= K^T(K\mathbf{f} - \mathbf{d}) + \alpha L(\mathbf{f})\mathbf{f} \\ \text{Hess}_{\mathbf{TV}_\beta}(\mathbf{f}) &= K^T K + \alpha L(\mathbf{f}) + \alpha L'(\mathbf{f})\mathbf{f}\end{aligned}$$

2.6 Scelta del Parametro

In questa sezione si riporta l'esempio della GCV come funzione per la scelta del parametro.

Indicando conf_α la soluzione regolarizzata per un funzionale del tipo (2.9), allora una sorta di indice dell'errore commesso nell'approssimazione si può indicare con

$$\mathbf{e}_\alpha = \mathbf{f}_\alpha - \mathbf{f}_{true}$$

dove \mathbf{f}_{true} è la soluzione reale del problema. Ovviamente, questo tipo di indice non è possibile calcolarlo, quindi si preferisce definire un'altra stima dell'errore compiuto, l'errore predittivo:

$$\mathbf{p}_\alpha = K\mathbf{e}_\alpha = K\mathbf{f}_\alpha - K\mathbf{f}_{true} \quad (2.19)$$

Anche questo non è calcolabile direttamente, ma è più facile averne una stima accurata.

La quantità

$$\frac{1}{n}\|\mathbf{p}_\alpha\|^2 = \frac{1}{n}\|K\mathbf{f}_\alpha - K\mathbf{f}_{true}\|^2$$

è chiamata *rischio predittivo*. Il metodo che si va ad esporre mira a minimizzare questo rischio.

Posto

$$\mathbf{f}_\alpha = R_\alpha \mathbf{d}$$

$$A_\alpha = K R_\alpha$$

dove R_α è l'operatore di regolarizzazione, l'errore predittivo prende la forma

$$\mathbf{p}_\alpha = (A_\alpha - Id)K\mathbf{f}_{true} + A_\alpha\eta$$

con η errore additivo di varianza σ^2 .

Il valore atteso del rischio predittivo è allora

$$\mathbf{E}\left[\frac{1}{n}\|\mathbf{p}_\alpha\|^2\right] = \frac{1}{n}\|(A_\alpha - Id)K\mathbf{f}_{true}\|^2 + \frac{\sigma^2}{n}\text{trace}(A_\alpha^2)$$

Ponendo $\mathbf{r}_\alpha = K\mathbf{f}_\alpha - \mathbf{d}$ e svolgendo i calcoli¹³ si ha

$$\mathbf{E}\left[\frac{1}{n}\|\mathbf{p}_\alpha\|^2\right] = \mathbf{E}\left[\frac{1}{n}\|\mathbf{r}_\alpha\|^2\right] + \frac{2\sigma^2}{n}\text{trace}(A_\alpha) - \sigma^2$$

¹³Per i dettagli, vedere [CV].

Il *metodo della stima dell'errore predittivo non distorto* (UPRE, dall'inglese Unbiased Predictive Risk Estimator) consiste quindi nel considerare

$$U(\alpha) = \frac{1}{n} \|\mathbf{r}_\alpha\|^2 + \frac{2\sigma^2}{n} \text{trace}(A_\alpha) - \sigma^2 \quad (2.20)$$

e trovare $\tilde{\alpha}$ tale che

$$\tilde{\alpha} = \arg \min_{\alpha} U(\alpha) \quad (2.21)$$

Questo metodo ha bisogno di una stima della varianza dell'errore sull'immagine. Il metodo GCV, *Generalized Cross Validation*, invece non ha bisogno di questa stima. Il parametro $\tilde{\alpha}$ scelto da questo metodo è tale per cui

$$\tilde{\alpha} = \arg \min_{\alpha} GCV(\alpha) \quad (2.22)$$

dove

$$GCV(\alpha) = \frac{\frac{1}{n} \|\mathbf{r}_\alpha\|^2}{\left[\frac{1}{n} \text{trace}(Id - A_\alpha)\right]^2} \quad (2.23)$$

Anche questo metodo fornisce una stima per il rischio predittivo.

Capitolo 3

Blind Deconvolution

In questo capitolo verranno presentati alcuni algoritmi che vengono utilizzati nel trattare immagini digitali.

Nella prima sezione verranno esposti alcuni metodi numerici di ottimizzazione, metodi cioè che permettono di trovare il minimo di funzionali. Verranno affrontati i metodi del Gradiente Coniugato, della Discesa Ripida, di Newton e di Newton Proiettato.

Nella seconda sezione verrà esposto il problema di blind deconvolution, il problema cioè di image deblurring nel caso in cui i parametri della Point Spread Function non siano noti.

3.1 Metodi di Ottimizzazione

Il cercare la soluzione per un problema del tipo

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

porta spesso alla ricerca della soluzione di un sistema lineare o non lineare

$$Ax = b$$

Una strada per risolvere sistemi di questo tipo può essere l'applicazione di *metodi iterativi*, metodi cioè che ricercano la soluzione mediante una opportuna successione x_k . La successione in generale può avere due forme

$$\triangleright x_{k+1} = \mathbf{H}x_k + d;$$

$$\triangleright x_{k+1} = x_k + \lambda_k p_k.$$

Esempi di metodi del primo tipo¹ possono essere metodo di Jacobi, Gauss-Siedel, SOR; metodi del secondo tipo possono essere Gradiente Coniugato, Discesa Ripida, metodo GMRES.

In questo elaborato, ci si concentrerà su metodi del secondo tipo:

$$x_{k+1} = x_k + \lambda_k p_k$$

p_k è detta *direzione di ricerca*, mentre λ_k è il *passo di ricerca*.

¹Per approfondimenti su metodi di questo tipo, si può consultare [?].

Def. 3.1. La direzione d è detta direzione di discesa se esiste $\delta > 0$ tale che

$$J(x_k + \tau d) < J(x_k)$$

per ogni $0 < \tau < \delta$.

Una richiesta ovvia per la direzione di ricerca è che sia anche una direzione di discesa. Si ha la seguente

Proposizione 3.1. Sia $J \in \mathcal{C}^1$; allora il vettore p è una direzione di discesa per J in x se

$$p^T \nabla J(x) < 0$$

3.1.1 Discesa Ripida

L'algoritmo di discesa ripida prende il nome dalla scelta che si fa per la direzione di ricerca: il gradiente negativo. L'algoritmo ha la seguente forma:

▷ si sceglie una stima iniziale x_0 per la soluzione;

▷ per $k = 0, 1, \dots$

- $p_k = -\nabla J(x_k)$
- $\lambda_k = \arg \min_{\lambda} J(x_k + \lambda p_k)$
- $x_{k+1} = x_k + \lambda_k p_k$
- controllo per la convergenza.

La scelta del passo di ricerca viene effettuata tramite una *ricerca in linea*; nelle applicazioni, spesso esso viene scelto tramite la regola di Armijo.

Nel caso in cui il funzionale J assuma la forma

$$J(x) = \frac{1}{2} \langle Ax, x \rangle + \langle b, x \rangle + c \quad (3.1)$$

con A simmetrica e definita positiva, allora J è strettamente convesso ed ha un unico punto di minimo x^* ; grazie a ciò è possibile calcolare il passo migliore λ_k per ogni iterazione:

$$\lambda_k = \frac{\|g_k\|^2}{\langle Ag_k, g_k \rangle} \quad (3.2)$$

dove $g_k = \nabla J(x_k)$. In questo caso, si ha il seguente risultato di convergenza

Teorema 3.1. Sia J un funzionale definito come in (3.1); per ogni dato iniziale x_0 l'iterato $x_{k+1} = x_k + \lambda_k g_k$ (con λ_k definito come in (3.2)) converge con velocità di convergenza

$$\|x_k - x^*\|_A \leq \left(\frac{\text{cond}(A) - 1}{\text{cond}(A) + 1} \right)^k \|x_0 - x^*\|_A$$

dove $\|\cdot\|_A$ è la norma energia di A , definita come $\sqrt{\langle A\cdot, \cdot \rangle}$.

Il numero $\text{cond}(A)$ è il cosiddetto *numero di condizionamento* della matrice A ed è definito come

$$\text{cond}(A) = \frac{\max \sigma_0(A)}{\min \sigma_0(A)}$$

dove $\sigma_0(A)$ denota l'insieme dei valori singolari diversi da zero. Se A è simmetrica definita positiva, allora $\text{cond}(A)$ diventa il rapporto fra il massimo e il minimo autovalore di A .

Si noti che la matrice A nella definizione (3.1) è la matrice Hessiana del funzionale: se essa è mal condizionata, il metodo della discesa ripida ha una velocità di convergenza molto lenta.

3.1.2 Gradiente Coniugato

Si consideri ancora il funzionale definito in (3.1) con A simmetrica e definita positiva. Il metodo del gradiente coniugato è un metodo iterativo della forma

$$x_{k+1} = x_k + \alpha_k p_k$$

dove p_k è una combinazione lineare della direzione precedente e del gradiente di J nel punto x_k :

$$p_k = -\nabla J(x_k) + \beta_{k-1} p_{k-1}$$

I coefficienti β_k sono scelti in modo che la direzione p_k sia A -coniugata² alle direzioni precedenti p_0, p_1, \dots :

$$\beta_{k-1} = \frac{(\nabla J(x_k))^T A p_{k-1}}{p_{k-1}^T A p_{k-1}}$$

Gli α_k sono definiti da

$$\alpha_k = \frac{(\nabla J(x_k))^T p_k}{p_k^T A p_k}$$

In realtà, i coefficienti α_k e β_k vengono scelti in un altro modo, equivalente, ma meno costoso dal punto di vista computazionale:

$$\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k} \quad \beta_{k-1} = \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}}$$

dove $r_k = b + Ax_k$ è il residuo del sistema.

L'algoritmo ha la seguente forma:

- ▷ $k = 0$;
- ▷ scelta di un iterato iniziale x_0 ;
- ▷ gradiente iniziale $g_0 = Ax_0 + b$;
- ▷ $p_0 = -g_0$;
- ▷ $\delta_0 = \|g_0\|^2$

²Se A è simmetrica definita positiva, due vettori u e v si dicono A -coniugati se $u^T A v = 0$.

▷ inizio delle iterazioni

- $h_k = Ap_k$;
- $\alpha_k = \frac{\delta_k}{\langle p_k, h_k \rangle}$;
- $x_{k+1} = x_k + \alpha_k p_k$;
- $g_{k+1} = g_k + \alpha_k h_k$;
- $\delta_{k+1} = \|g_{k+1}\|^2$;
- $\beta_k = \frac{\delta_{k+1}}{\delta_k}$;
- $p_{k+1} = -g_{k+1} + \beta_k p_k$;
- controllo dei criteri di convergenza;

Si hanno i seguenti risultatti di convergenza per il metodo del gradiente coniugato.

Def. 3.2. Sia A una matrice simmetrica, $\mathbf{v} \in \mathbb{R}^n$. Il sottospazio di Krylov generato da A e \mathbf{v} è dato da

$$S_\nu(A, \mathbf{v}) = \text{span}\{\mathbf{v}, A\mathbf{v}, A^2\mathbf{v}, \dots, A^{\nu-1}\mathbf{v}\}$$

Teorema 3.2. Per $k = 1, 2, \dots$ l'iterato k -esimo del GC soddisfa

$$f_k = \arg \min_{f \in f_0 + S_k(A, g_0)} \|f - f^*\|_A$$

dove f^* è la soluzione del sistema. L'errore corrispondente all'iterazione k $e_k := f_k - f^*$ soddisfa

$$\|e_k\|_A = \min_{q \in \Pi_1^k} \|q(A)e_0\|_A$$

dove Π_1^k denota i polinomi di gradi al più k e tali che $q(0) = 1$, mentre $\|\cdot\|_A$ è la norma energia di matrice A .

Corollario 3.2.1. Se A è simmetrica e definita positiva, allora per ogni $b \in \mathbb{R}^n$ e per ogni scelta iniziale per $x_0 \in \mathbb{R}^n$ il metodo del gradiente coniugato converge alla soluzione esatta del sistema $Ax = -b$ in al più n iterazioni.

Si può mostrare che l'errore all'iterazione k può essere stimato con

$$\|e_k\|_A \leq \left(\frac{\sqrt{\text{cond}(A) - 1}}{\sqrt{\text{cond}(A) + 1}} \right)^k \|e_0\|_A$$

Per accelerare la computazione (avere cioè un numero di condizione più basso), si può attuare un preconditionamento della matrice A , cioè trovare una matrice simmetrica e definita positiva M tale che $M^{-1}A$ abbia una distribuzione migliore degli autovalori³.

³Per i dettagli, vedere [CV].

3.1.3 Metodo di Newton

Sia J un funzionale differenziabile⁴ due volte nel senso di Fréchet su di uno spazio di Hilbert \mathcal{H} . Allora è possibile scrivere una sua approssimazione al secondo ordine nel seguente modo

$$J(x+h) = J(f) + \langle \nabla J(f), h \rangle + \frac{1}{2} \langle \text{Hess}_J(f)h, h \rangle + o(\|h\|_{\mathcal{H}}^2) \quad (3.3)$$

per $\|h\|_{\mathcal{H}} \rightarrow 0$. Si può considerare quindi l'approssimazione quadratica di $J(x_k + s)$, dove x_k è una stima per il minimo di J :

$$Q_k(s) = J(x_k) + \langle \nabla J(x_k), s \rangle + \frac{1}{2} \langle \text{Hess}_J(x_k)s, s \rangle \quad (3.4)$$

Se l'Hessiana di J è definita positiva, allora $Q_k(s)$ ha un'unico punto di minimo che soddisfa

$$\nabla J(x_k) + \text{Hess}_J(x_k)s = 0$$

Prendendo $x_k + s$ come nuova stima per il punto di minimo di J , si pone l'iterazione $k+1$ -esima di Newton come

$$x_{k+1} = x_k - [\text{Hess}_J(x_k)]^{-1} \nabla J(x_k) \quad (3.5)$$

La direzione $p_k = [\text{Hess}_J(x_k)]^{-1} \nabla J(x_k)$ è di discesa se e soltanto se l'Hessiana è definita positiva. Il costo del metodo di Newton è piuttosto alto: ad ogni iterazione, è necessario valutare l'Hessiana e risolvere un sistema per la direzione di ricerca. Esistono dei metodi per velocizzare questi passaggi nell'algoritmo.

Il metodo di Newton ha una velocità di convergenza quadratica, nel caso in cui l'iterato iniziale x_0 venga scelto in un intorno del minimo x^* .

Teorema 3.3. Si supponga che $\nabla J(x^*) = 0$ e che $\text{Hess}_J(x^*)$ sia simmetrica, definita positiva e lipschitziana:

$$\|\text{Hess}_J(x+h) - \text{Hess}_J(x)\| \leq \gamma \|h\|$$

dove γ è una costante di Lipschitz. Allora, se l'iterato iniziale x_0 è scelto in un intorno opportuno di x^* il metodo di Newton è convergente e la sua velocità di convergenza è quadratica:

$$\|x_{k+1} - x^*\| = o(\|x_k - x^*\|^2)$$

Il metodo di Newton quindi ha i seguenti possibili svantaggi:

- matrice Hessiana da calcolare ad ogni passo;
- ricerca della soluzione di un sistema ad ogni passo;
- scelta non opportuna dell'iterato iniziale.

Per ovviare a questi problemi si possono adottare le seguenti strategie:

- al posto di considerare la matrice Hessiana del funzionale, si considera una sua approssimazione (metodo BFGS⁵);

⁴Per nozioni relative ad Analisi Funzionale, si può consultare [CV].

⁵Per un approfondimento, consultare [CV].

- la soluzione del sistema per trovare la direzione di ricerca può essere molto dispendiosa per problemi di grandi dimensioni, quindi metodi come l'eliminazione di Gauss o metodi di fattorizzazione della matrici sono inefficienti. Si preferisce quindi andare a risolvere il sistema in maniera non esatta tramite metodi iterativi (ad esempio, discesa ripida o gradiente coniugato). I metodi di Newton che utilizzano questa strategia per trovare la direzione di ricerca vengono chiamati *metodi di Newton inesatti*, e mirano a trovare una direzione tale che soddisfi la condizione del residuo:

$$\|\nabla J(x_k) + \text{Hess}_J(x_k)p_k\| < \eta_k \|\nabla J(x_k)\|$$

dove $\eta_k \in [0, \eta]$, $\eta \in [0, 1)$ costante; η viene detto *termine forzante*. La scelta del termine forzante influenza la velocità di convergenza dell'algoritmo.

- per poter permettere una certa libertà di scelta si modifica l'iterazione del metodo nel seguente modo

$$x_{k+1} = x_k - \lambda_k [\text{Hess}_J(x_k)]^{-1} \nabla J(x_k)$$

dove λ_k viene scelto in modo che $m(x_{k+1}) < m(x_k)$ per una certa funzione di merito m . La scelta per la lunghezza del passo λ_k nelle applicazioni viene effettuata implementando un algoritmo di backtracking in modo che il parametro trovato soddisfi la regola di Armijo. I metodi che attuano questa opzione per permettere una qualsiasi scelta per l'iterato iniziale prendono il nome di *metodi di Newton modificati*; nel caso in cui la direzione di ricerca sia scelta con un algoritmo iterativo, prendono il nome di *metodi di Newton inesatti modificati*.

3.1.4 Metodo di Newton Proiettato

Nel problema di image deblurring, l'obiettivo è trovare il minimo di un funzionale che abbia come argomento immagini digitali. Come si è visto nel primo capitolo, gli elementi delle immagini digitali risiedono in un range $[0, L]$, quindi sono tutti positivi. Nell'applicare metodi di ottimizzazione, è possibile trovare soluzioni di minimo con alcuni elementi negativi. Per ovviare a questo problema, si pongono dei vincoli sulla soluzione: si richiede di minimizzare il funzionale richiedendo che la soluzione sia positiva. Il problema si tramuta quindi in

$$\tilde{x} = \min_{x \geq 0} \frac{1}{2} \|F(x)\|^2 \quad (3.6)$$

I metodi visti finora sono detti metodi non vincolati, in quanto si permetteva che la soluzione fosse totalmente libera da restrizioni.

Il problema generale⁶ è il seguente

$$\begin{aligned} \min_x f(x) \\ x \in \mathcal{X} \end{aligned} \quad (3.7)$$

dove \mathcal{X} è un sottoinsieme non vuoto, chiuso e convesso di \mathbb{R}^n . Nel caso in cui $f(x)$ sia una distanza, il seguente teorema viene in aiuto

⁶Per una visione della teoria più approfondita, si può consultare [DB].

Teorema 3.4. Sia \mathcal{X} un sottoinsieme non vuoto, chiuso e convesso di \mathbb{R}^n . Allora

- (i) per ogni $z \in \mathbb{R}^n$ esiste un unico $x^* \in \mathcal{X}$ tale che minimizza $\|z - x\|$ su tutti gli $x \in \mathcal{X}$. Questo vettore è chiamato la *proiezione* di z su \mathcal{X} ed è denotato con $[z]^+$;
- (ii) dato un vettore $z \in \mathbb{R}^n$, un vettore $x^* \in \mathcal{X}$ è equivalente alla proiezione di z se e soltanto se

$$(z - x^*)(x - x^*) \leq 0 \quad \forall x \in \mathcal{X}$$

- (iii) la mappa $f : \mathbb{R}^n \rightarrow \mathcal{X}$ definita da $f(x) = [x]^+$, dove $[\cdot]^+$ denota la proiezione su \mathcal{X} , è continua e non espansiva, cioè

$$\|[x]^+ - [y]^+\| \leq \|x - y\| \quad x, y \in \mathbb{R}^n$$

- (iv) nel caso in cui \mathcal{X} sia un sottospazio, un vettore $x^* \in \mathcal{X}$ è uguale alla proiezione $[z]^+$ se e soltanto se $z - x^*$ è ortogonale a \mathcal{X} , cioè

$$(z - x^*)^T x = 0 \quad \forall x \in \mathcal{X}$$

Si richiede che i metodi che si andranno a studiare per risolvere (3.7) abbiano le seguenti caratteristiche:

1. non siano dipendenti dalla struttura dell'insieme che definisce i vincoli, solo dalla sua convessità;
2. generano una sequenza di punti ammissibili (definizione (3.3)) cercando lungo direzioni di discesa.

Si pone allora la seguente definizione.

Def. 3.3. Un vettore x che soddisfa i vincoli posti dal problema (3.7) viene detto *ammissibile*. Dato un vettore ammissibile, una *direzione ammissibile* è un vettore d tale che $x + \alpha d$ è ancora un vettore ammissibile per ogni $\alpha > 0$ abbastanza piccolo. Un *metodo a direzione ammissibile* è un metodo che ha come punto iniziale x_0 un vettore ammissibile e genera una successione di iterati $\{x_k\}$ ammissibili nel modo seguente

$$x_{k+1} = x_k + \alpha_k d_k$$

dove, se x_k non è stazionario, d_k è una direzione ammissibile, che è anche una direzione di discesa. Il passo α_k è scelto in modo che $x_k + \alpha_k d_k \in \mathcal{X}$.

Si cerca ovviamente di costruire un algoritmo che sia anche un metodo di discesa, cioè tale che verifichi

$$f(x_k + \alpha_k d_k) < f(x_k) \quad \forall k$$

La scelta del passo può essere effettuata tramite varie opzioni: con la ricerca in linea, con la regola di Armijo oppure prendendo un passo costante che comunque assicuri di avere vettori ammissibili e convergenza.

Si ha il seguente risultato di convergenza per i metodi appena descritti.

Def. 3.4. Una sequenza $\{d_k\}$ di direzioni ammissibili si dice *gradient related* a $\{x_k\}$ se e soltanto se vale che per ogni sottosuccessione $\{x_k\}_{k \in K}$ che converge a un punto non stazionario, la corrispondente sottosuccessione $\{d_k\}_{k \in K}$ è limitata e soddisfa

$$\limsup_{k \rightarrow \infty} \sup_{k \in K} \nabla f(x_k)^T d_k < 0$$

Teorema 3.5. Sia $\{x_k\}$ una successione generata da un metodo a direzione ammissibile $x_{k+1} = x_k + \alpha_k d_k$. Si assuma che $\{d_k\}$ sia gradient related e che α_k sia scelto con la regola di Armijo. Allora ogni punto limite del metodo è un punto stazionario.

Esistono risultati di convergenza anche per metodi con scelta del passo costante, ma la loro trattazione va fatta caso per caso.

La maniera più naturale per costruire una direzione ammissibile della forma $\bar{x}_k - x_k$ che soddisfi la condizione di discesa e che può essere usata nel metodo $x_{k+1} = x_k + \alpha_k (\bar{x}_k - x_k)$ è risolvere il problema di ottimizzazione

$$\begin{aligned} \min_x \nabla f(x)^T (x - x_k) \\ x \in \mathcal{X} \end{aligned} \tag{3.8}$$

e ottenere la soluzione $\bar{x}_k = \arg \min_{x \in \mathcal{X}} \nabla f(x)^T (x - x_k)$. Si è assunta l'ipotesi che \mathcal{X} sia compatto, quindi è possibile trovare una direzione che soddisfi (3.8). Questo metodo prende il nome di *metodo del gradiente condizionato*, o *metodo di Frank-Wolfe*. In particolare, il punto \bar{x}_k è il punto più lontano lungo la direzione del gradiente negativo.

È possibile dimostrare che la successione delle direzioni di ricerca del metodo del gradiente condizionato è gradient related, quindi si può applicare il teorema precedentemente illustrato e avere un risultato di convergenza per questo algoritmo; la velocità di convergenza dipende comunque dall'insieme definente i vincoli.

Si consideri ora all'iterazione k -esima una matrice H_k definita positiva, e si attui la trasformazione di variabili

$$x = (H_k)^{-\frac{1}{2}} y$$

in modo che il problema (3.7) venga riscritto

$$\begin{aligned} \min_y h(y) = f((H_k^{-\frac{1}{2}})y) \\ y \in Y_k \end{aligned}$$

dove $Y_k = \{y | (H_k)^{-\frac{1}{2}} y \in \mathcal{X}\}$. L'iterazione del gradiente proiettato prende la forma

$$y_{k+1} = y_k + \alpha_k (\bar{y}_k - y_k)$$

con

$$\bar{y}_k = [y_k - s_k(\nabla h_k(y_k))]^+$$

con s_k e α_k passi. Utilizzando⁷ la trasformazione descritta sopra e la definizione di Y_k , il problema si può scrivere come

$$x_{k+1} = x_k + \alpha_k(\bar{x}_k - x_k)$$

dove \bar{x}_k è dato da

$$\bar{x}_k = \arg \min_{x \in \mathcal{X}} \left\{ \nabla f(x_k)^T (x - x_k) + \frac{1}{2s_k} (x - x_k)^T H_k (x - x_k) \right\} \quad (3.9)$$

Un metodo di questo genere prende il nome di *metodo del gradiente scalato proiettato*. La seguente proposizione dà un risultato di convergenza.

Proposizione 3.2. Sia $\{x_k\}$ una successione generata dal metodo del gradiente scalato proiettato con α_k scelto tramite la regola di Armijo o una ricerca in linea. Si supponga che per alcuni scalari positivi c_1, c_2 le matrici H_k soddisfino

$$c_1 \|z\|^2 \leq z^T H^k z \leq c_2 \|z\|^2$$

per ogni $z \in \mathbb{R}^n$, per $k = 0, 1, 2, \dots$. Allora ogni punto limite di $\{x_k\}$ è un punto stazionario.

Supponendo di avere f in (3.7) appartenente a \mathcal{C}^2 con Hessiana positiva, è possibile considerare la matrice $H_k = \text{Hess}_f(x_k)$: il metodo prende la forma

$$x_{k+1} = x_k + \alpha_k(\bar{x}_k - x_k)$$

$$\bar{x}_k = \arg \min_{x \in \mathcal{X}} \left\{ \nabla f(x_k)^T (x - x_k) + \frac{1}{2s_k} (x - x_k)^T \text{Hess}_f(x_k) (x - x_k) \right\}$$

Si ha così il *metodo di Newton vincolato*.

Si ritorna ora al problema originario. L'insieme rispetto al quale si vuole andare a performare la minimizzazione è $\mathcal{X} = \{x \geq 0\}$. Per risolvere questo problema, si implementa un metodo della forma

$$x_{k+1} = [x_k - \alpha_k D_k \nabla f(x_k)]^+ \quad (3.10)$$

dove D_k è una matrice definita positiva, non necessariamente diagonale, e $[\cdot]^+$ denota la proiezione sull'ortante positivo rispetto alla norma euclidea. Questo metodo viene definito *metodo di proiezione a due metriche* in quanto coinvolge due matrici differenti di scala: la D_k , che opera sul gradiente, e la matrice identità che viene usata nella proiezione euclidea. La difficoltà principale di questo metodo consiste nel fatto che le matrici di scala D_k non sempre restituiscono una direzione di discesa, ma per fortuna esiste una classe di matrici siffatte che garantiscono la discesa.

Def. 3.5. Si pone per ogni $x \geq 0$

$$I^+(x) = \left\{ i \mid x_i = 0, \frac{\partial f(x)}{\partial x_i} > 0 \right\}$$

Si dice che una matrice simmetrica di dimensioni $n \times n$ è *diagonale rispetto ad un sottoinsieme di indici* $I \subset \{1, 2, 3, \dots, n\}$ se e soltanto se $d_{ij} = 0 \forall i \in I, j = 1, \dots, n, i \neq j$.

⁷Per i dettagli relativi a questa sezione, si può consultare [DB].

Proposizione 3.3. Sia $x \geq 0$ e D una matrice simmetrica, definita positiva e diagonale rispetto a I^+ . Si ponga

$$x(\alpha) = [x - \alpha D \nabla f(x)]^+ \quad \forall \alpha \geq 0$$

Il vettore x è stazionario se e soltanto se $x = x(\alpha) \quad \forall \alpha \geq 0$. Se x non è stazionario, esiste uno scalare $\underline{\alpha}$ tale che $f(x(\alpha)) < f(x) \quad \forall \alpha \in (0, \underline{\alpha}]$.

Questa proposizione dice che per assicurarsi di avere una direzione di discesa la matrice D_k in (3.10) deve essere scelta diagonale rispetto all'insieme degli indici $I^+(x_k)$.

Il metodo implementato al calcolatore (che verrà preso in considerazione nel prossimo capitolo) prende il nome di *Fast Projected Quasi Newton method* (FPQN), introduce alcuni accorgimenti per garantire la convergenza del metodo.

La definizione dell'insieme I^+ viene modificata⁸

$$I^+(x^k) = \left\{ i \mid 0 \leq x_i^k \leq \epsilon_k, \frac{\partial f(x^k)}{\partial x_i} > 0 \right\}$$

con

$$\epsilon^k = \min \epsilon, w^k, \quad w^k = |x^k - [x^k - \nabla f(x^k)]^+|$$

e con ϵ piccolo e positivo.

Il metodo FPQN per il problema generale ha la forma

$$x^{k+1} = [x^k + \alpha^k p^k]^+$$

dove p^k è la direzione di discesa calcolata tramite

$$p^k = D^k g^k$$

Nell'espressione precedente, $g^k := \nabla f(x^k)$, e D^k è diagonale rispetto a $I^+(x^k)$ ed è definita come

$$d_{ij}^k = \begin{cases} \delta_{ij} & i, j \in I^+(x^k) \\ \{Q^{-1}\}_{ij} & \text{altrimenti} \end{cases}$$

dove Q è un'approssimazione circolante per l'Hessiana del funzionale che si va a minimizzare.

Si può provare⁹

Proposizione 3.4. Le matrici D^k sono definite positive e soddisfano

$$\mu_1 \|z\|^2 \leq z^T D^k z \leq \mu_2 \|z\|^2$$

per ogni $z \in \mathbb{R}^N$, $k = 0, 1, \dots$ e per μ_1, μ_2 scalari positivi.

Questa proposizione, unita alla (3.2), dà la convergenza del metodo FPQN.

⁸Per comodità di notazione, l'apice k in x^k denoterà l'iterato k -esimo, mentre x_i la componente i -esima del vettore.

⁹Si veda [LP].

3.2 Problema di Blind Deconvolution

La Point Spread Function può avere diverse forme, come spiegato nei capitoli precedenti, dipendenti dal tipo di strumento che si sta utilizzando per l'acquisizione di immagini. Inoltre, essa dipende da un certo numero di parametri: indicando con $K(\mathbf{y})$ la PSF dipendente dal vettore di parametri \mathbf{y} , il modello di formazione di un'immagine digitale prende la seguente forma:

$$\mathbf{g} = K(\mathbf{y})\mathbf{f} + \eta \quad (3.11)$$

dove η è il rumore additivo. Il vettore \mathbf{y} normalmente ha un numero basso di componenti: nel caso gaussiano, ad esempio, ha 3 componenti (s_1, s_2, ρ), mentre se si considera il Moffat Blur (1.8) ha 4 componenti (s_1, s_2, ρ, β).

Una volta che si ha a disposizione la forma della PSF e i parametri da cui essa dipende, si può attuare un algoritmo per poter risalire ad una stima dell'immagine originaria. Questo è il caso migliore in cui ci si possa trovare: il problema sorge quando si conosce il modello matematico del tipo di blur presente sull'immagine ma non si conoscono i parametri da cui esso dipende.

L'interesse consiste quindi nel ricavare f nell'equazione (3.11): essendo $K(\mathbf{y})$ compatto, il problema è mal posto, come spiegato nel capitolo precedente. Non si possono usare quindi metodi diretti per trovare f , si ottengono risultati disastrosi (vedasi l'uso dell'antitrasformata di Fourier o dell'inversione della SVD). Ci si conduce quindi al problema (2.9), dove $\mathcal{X} = \mathcal{Y} = \mathbb{R}^N$ e si applicano algoritmi di ottimizzazione per trovare il minimo del funzionale in questione.

Una procedura per risolvere tale problema in assenza di parametri della PSF è la seguente: si pone $z = (x, y)$, dove x rappresenta un'immagine, y i parametri della PSF; si definisce

$$F(z) = F(x, y) = \begin{bmatrix} K(\mathbf{y}) \\ \alpha \mathbf{A} \end{bmatrix} x - \begin{bmatrix} b \\ 0 \end{bmatrix} \quad (3.12)$$

che sarà quindi il funzionale T_α in (2.9), e A rappresenta il penalty functional; si cerca perciò \tilde{z} tale che

$$\tilde{z} = \min_z \frac{1}{2} \|F(z)\|^2 \quad (3.13)$$

Si è tradotto quindi il problema originario ad un problema di minimi quadrati non lineare, che può essere risolto con un metodo iterativo di tipo Gauss-Newton: si può calcolare cioè

$$z_{k+1} = z_k + d_k$$

dove d_k è una direzione di ricerca. Ponendo $\psi(z) = \frac{1}{2} \|F(z)\|^2$, si ha che

$$\psi' = J_\psi^T F$$

$$\psi'' = J_\psi^T J_\psi$$

dove J_ψ è la matrice Jacobiana e ψ'' è un'approssimazione della matrice Hessiana. La direzione di ricerca può essere scritta

$$d_k = -(\psi''(z_k))^{-1} \psi'(z_k)$$

Ponendo $r = -F$, il calcolo per la direzione di ricerca è equivalente a risolvere una minimizzazione ai minimi quadrati

$$d_k = \min_d \frac{1}{2} \|J_\psi d - r\|^2$$

Riassumendo, l'algoritmo diventa

▷ scelta di una stima iniziale z_0 ;

▷ per $k = 0, 1, 2, \dots$

- $r_k = b - K(\mathbf{y}_k)x_k$;
- $d_k = \arg \min_d \|J_\psi d - r_k\|^2$;
- $z_{k+1} = z_k + d_k$

Questo è un procedimento molto generale, ma il costo computazionale per calcolare J_ψ e risolvere i conseguenti sistemi è molto alto. Inoltre, è necessario conoscere a priori i parametri di regolarizzazione α presente nella definizione di F . La strada che si segue quindi è quella di separare¹⁰ le due variabili x e y , notando che il funzionale F è lineare nella x . Questo porta a definire un nuovo funzionale

$$\phi(y) := \psi(x(y), y)$$

dove $x(y)$ è soluzione del seguente problema di minimo

$$\min_x \psi(x, y)$$

Per usare l'algoritmo di Gauss-Newton, è necessario calcolare $\phi'(y)$, e questo vale $F_y^T F$. Infine, per calcolare la direzione di ricerca si necessita di J_ϕ che vale $F_y = K'(\mathbf{y})x$. L'algoritmo finale diventa quindi il seguente:

▷ scelta di una stima iniziale y_0 per i parametri;

▷ per $k = 0, 1, 2, \dots$

- $x_k = \arg \min_x \left\| \begin{bmatrix} K(\mathbf{y}_k) \\ \alpha \mathbf{A} \end{bmatrix} x - \begin{bmatrix} b \\ 0 \end{bmatrix} \right\|^2$
- $r_k = b - K(\mathbf{y}_k)x_k$
- $d_k = \arg \min_d \|J_\rho d - r_k\|^2$
- $y_{k+1} = y_k + d_k$

Per una trattazione più approfondita, si può vedere [CN].

I due sottoproblemi di minimo possono essere affrontati con metodi di ottimizzazione diversa.

¹⁰Si può consultare [SVH] per i particolari.

Capitolo 4

Risultati Numerici

In questo capitolo verranno esposti i risultati numerici trovati eseguendo i test.

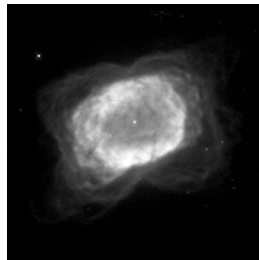
Nella prima sezione sono presentati i dati iniziali, con le PSF utilizzate e i tipi di rumore.

Nella seconda sezione viene vista un'applicazione del metodo FPQN nel caso di rumore gaussiano e poissoniano.

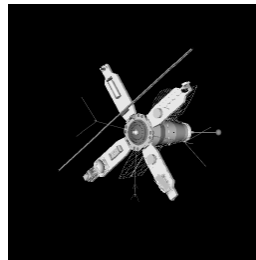
Nella terza sezione sono presentati i risultati dell'algoritmo per risolvere il problema di blind deconvolution.

4.1 Dati iniziali

Le immagini test utilizzate per effettuare gli esperimenti sono le seguenti:



(a) Nebula.



(b) Immagine di un satellite

Figura 4.1: Immagini test.

Queste due immagini sono state perturbate con tre diverse PSF: due di tipo gaussiano con parametri diversi in modo da avere lo stesso tipo di blurr ma più o meno marcato, mentre la terza viene presa dal set di dati `sim.mat` fornito con il pacchetto `Restore Tools`¹. Ad ognuna di esse viene aggiunto un rumore

¹<http://www.mathcs.emory.edu/~nagy/>.

bianco gaussiano oppure poissoniano, in modo da ottenere un SNR di circa 30. Le immagini che si sono ottenute sono nelle figure (4.2) e (4.5).

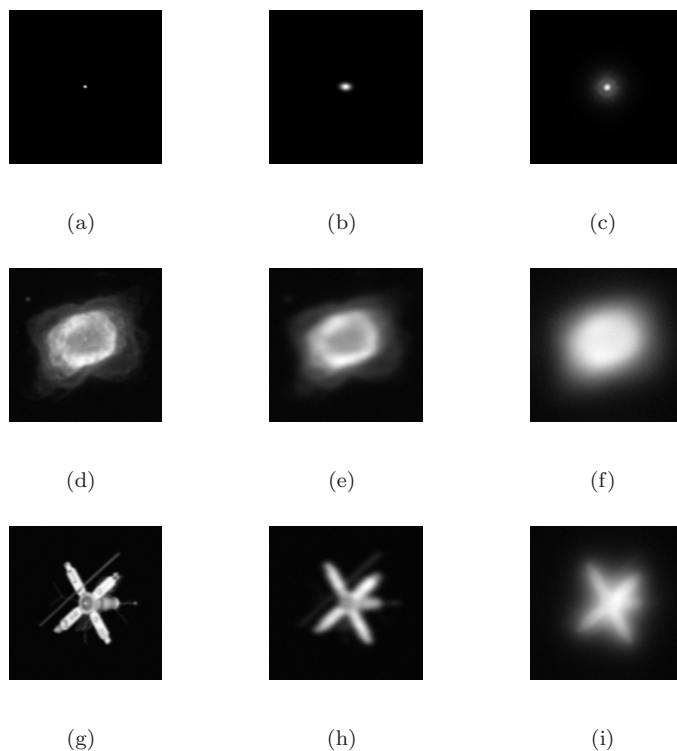


Figura 4.2: Dati dei problemi test. Le figure sulla prima riga rappresentano le immagini delle PSF considerate; la seconda riga è ottenuta dall'immagine della nebulosa perturbata con ognuna delle funzioni di blurr precedenti con l'aggiunta di rumore gaussiano bianco; analogo discorso per le immagini del satellite sulla terza e ultima riga.

4.2 FPQN

4.2.1 Rumore Gaussiano

Il primo test è consistito nel comparare i risultati ottenuti tramite l'applicazione dell'algoritmo FPQN e del metodo di Tikhonov alle immagini perturbate mostrate in figura (4.2) (nel caso cioè di blurr gaussiano e rumore bianco gaussiano). In entrambi i casi, il parametro di regolarizzazione non è stato preso tramite una funzione di scelta, ma empiricamente: si sono effettuati test² per ricercare il parametro α ottimale, in modo che l'errore relativo

$$e_{rel} = \frac{\|x_\eta - x_{true}\|_{fro}}{\|x_{true}\|_{fro}}$$

²Si sono testati parametri nel range $[0, 1]$ con un passo di $5 \cdot 10^{-5}$.

fosse il minimo possibile.

In figura (4.3) è presentato un esempio di comportamento dell'errore relativo rispetto alla scelta del parametro α . Si può notare che scelte non opportune possono portare a errori piuttosto alti, quindi la funzione di scelta del parametro deve essere molto efficiente per poter far restituire al metodo risultati buoni.

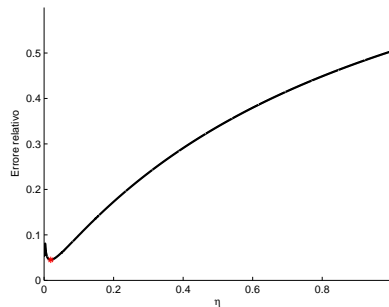


Figura 4.3: Andamento dell'errore relativo al variare del parametro η ; il grafico si riferisce ai test per la ricerca del parametro relativo all'immagine (4.2(d)).

Nella tabella (4.1) sono presentati a confronto i risultati ottenuti con i due metodi.

Il primo fatto che si può notare è che l'algoritmo FPQN dà risultato migliori rispetto al metodo di Tikhonov: questo è lecito aspettarselo in quanto la richiesta di avere elementi positivi nell'immagine dà un grado di accuratezza migliore. Nei grafici di figura (4.4) sono mostrati gli andamenti della funzione obiettivo (la funzione che si va a minimizzare) e del gradiente proiettato al variare delle iterazioni nel caso dell'immagine della nebulosa.



						
	Dato	FPQN	Tik	Dato	FPQN	Tik
PSF 4.2(a)	0.054205	0.045126	0.048234	0.264264	0.168433	0.197667
PSF 4.2(b)	0.108603	0.068175	0.070181	0.432993	0.294653	0.322825
PSF 4.2(c)	0.329015	0.109183	0.116639	0.704641	0.483467	0.362615

Tabella 4.1: Metodi a confronto. Nella colonna denominata **Dato** è presente l'errore dell'immagine perturbata su quella reale.

Un altro fatto importante da notare è che il metodo di Newton proiettato è molto veloce: in poche iterazioni giugne a convergenza restituendo un risultato soddisfacente. Nella tabella (4.2) sono riportati i parametri ottimale trovati tramite i test; il fatto interessante è la discrepanza presente fra il parametro ottimale trovato e il parametro fornito dalla funzione di scelta GCV: essendo un metodo statistico, non sempre fornisce parametri buoni per regolarizzare il

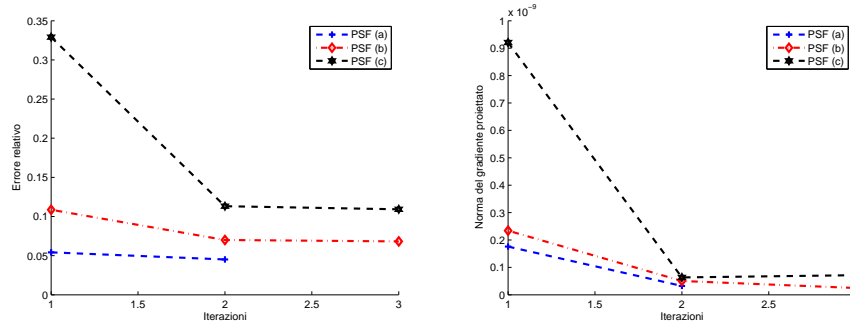


Figura 4.4: Andamento dell'errore relativo e del gradiente proiettato al variare delle iterazioni.

funzionale utilizzato.



						
	FPQN	Tik	GCV	FPQN	Tik	GCV
PSF 4.2(a)	0.019200	0.147500	0.006434	0.007000	0.070050	0.006449
PSF 4.2(b)	0.004550	0.065750	0.006324	0.002150	0.016150	0.007398
PSF 4.2(c)	0.000700	0.032150	0.006455	0.054650	0.015200	0.006177

Tabella 4.2: Parametri ottimali.

4.2.2 Rumore Poissoniano

Il secondo test viene effettuato su immagini con rumore poissoniano, perturbate con le medesime PSF presentate in figura (4.2).



				
	Dato	FPQN	Dato	FPQN
PSF 4.2(a)	0.355600	0.085616	0.372800	0.189578
PSF 4.2(b)	0.365296	0.083243	0.482803	0.340490
PSF 4.2(c)	0.478261	0.153435	0.730373	0.570818

Tabella 4.3: Risultati dell'applicazione del metodo FPQN nel caso di rumore poissoniano.

L'algoritmo considerato per il restauro di questo tipo di immagini è ancora un metodo FPQN, implementato diversamente da quello precedente in quanto il funzionale da minimizzare non è più quello dei minimi quadrati ma il funzionale di Kullbach-Leibler descritto nei capitoli precedenti.

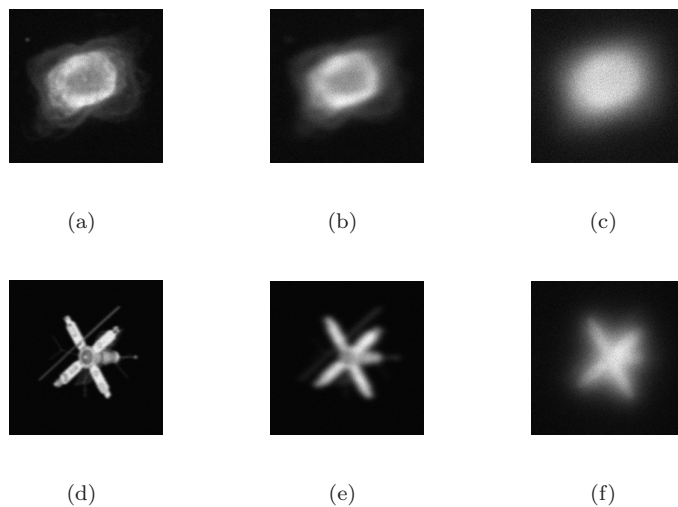


Figura 4.5: Immagini con rumore di Poisson; le PSF utilizzate sono quelle della prima riga della figura 4.2. In ogni figura è indicato l'errore relativo sull'esatta.

Anche in questo caso, si sono effettuati dei test per avere il parametro ottimale di regolarizzazione per applicare l'algoritmo.

4.3 Blind Deconvolution

In questa sezione vengono presentati i risultati dell'algoritmo di blind deconvolution.

La stima iniziale per i parametri della Point Spread Function è il vettore $\mathbf{y} = (10, 12, 5)$, molto lontano dai parametri veri³. L'algoritmo utilizzato è il seguente:

▷ $\mathbf{y}_0 = (10, 12, 5)$;

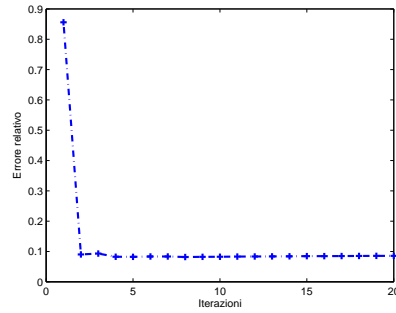
▷ per $k = 0, 1, 2, \dots$

- $x_k = \text{FPQN}\left(\left\| \begin{bmatrix} K_{Gauss}(\mathbf{y}_k) \\ \alpha_{opt} \end{bmatrix} x - \begin{bmatrix} blurred \\ 0 \end{bmatrix} \right\|^2\right)$
- $r_k = blurred - K_{Gauss}(\mathbf{y}_k)x_k$
- $d_k = \text{Tik}\left(\|J_\rho d - r_k\|^2\right)$
- $\mathbf{y}_{k+1} = \mathbf{y}_k + \lambda_k d_k$

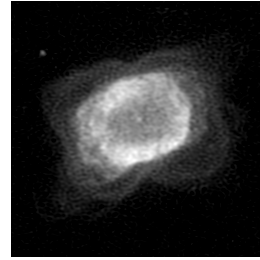
dove

▷ $K_{Gauss}(\mathbf{y}_k)$ rappresenta la PSF di tipo gaussiano di parametri \mathbf{y}_k ;

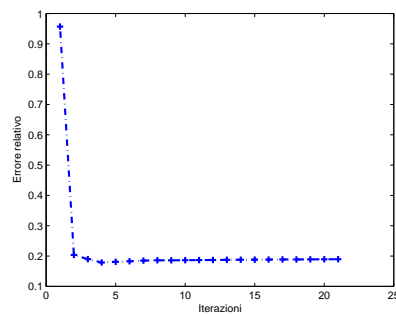
³È stato testato che partire da una stima di parametri vicina a quelli esatti porta come soluzione l'immagine di partenza.



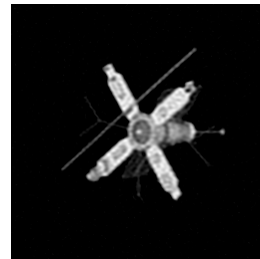
(a) Parametro ottimale: 0.00015.



(b) SSIM: 0.711054.



(c) Parametro ottimale: 0.00005



(d) SSIM: 0.928346.

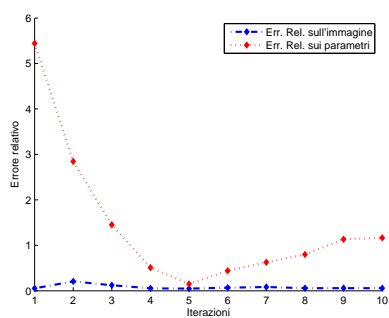
Figura 4.6: Risultati del metodo FPQN per la minimizzazione del funzionale Kullback-Leibler con la scelta ottimale del parametro. Le immagini mostrete si riferiscono ai dati presenti nelle figure (4.5(a)) e 4.5(d).

- ▷ $\text{FPQN}(\rho)$ rappresenta la ricerca del punto di minimo del funzionale ρ tramite l'algoritmo FPQN; nelle righe scritte sopra, $\rho = \rho_{ls}$;
- ▷ $\text{Tik}(\psi)$ rappresenta la ricerca del punto di minimo del funzionale ψ con il metodo di Tikhonov.

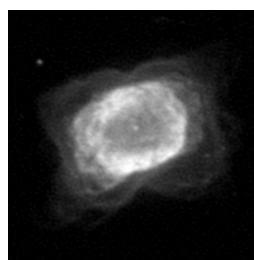
Il parametro λ_k è scelto tramite la regola di Armijo. Il parametro α_{opt} è il parametro ottimale trovato con i test precedenti, ed è tenuto fisso per l'intero algoritmo.

4.3.1 Rumore Gaussiano

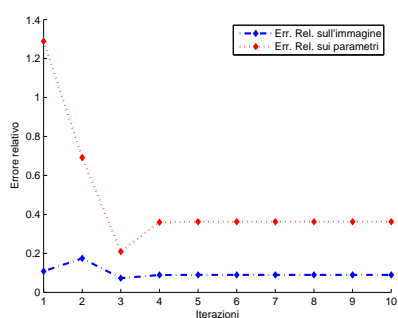
Il primo test effettuato è sulle immagini (4.2(d)), (4.2(e)), (4.2(g)), (4.2(h)). I risultati sono presentati in figura (4.7). Nella prima colonna sono presentati i grafici dell'andamento degli errori relativi rispetto all'immagine reale e rispetto ai parametri reali. Si è quindi individuato il numero di iterazioni che fornisce il risultato migliore e quindi si è mostrato nella seconda colonna l'immagine ottenuta.



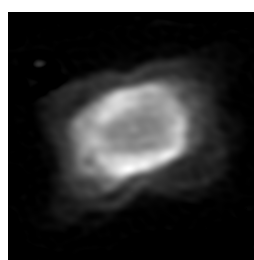
(a) Errore sui parametri: 0.148900.



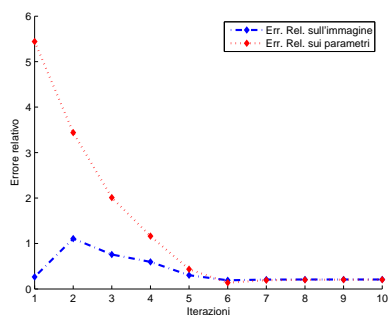
(b) Errore sull'immagine: 0.047441.



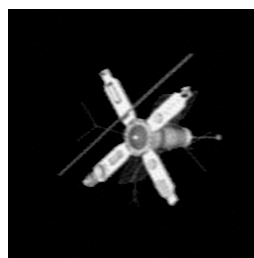
(c) Errore sui parametri: 0.219955.



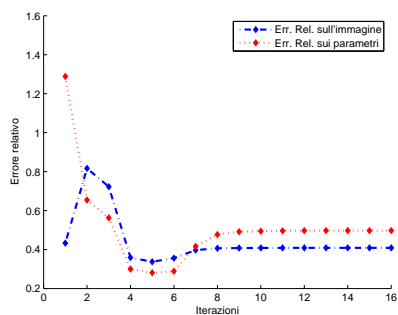
(d) Errore sull'immagine: 0.071480.



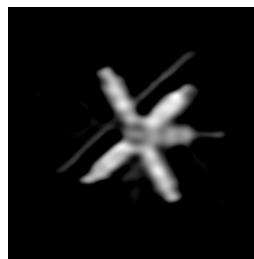
(e) Errore sui parametri: 0.137329.



(f) Errore sull'immagine: 0.190498.

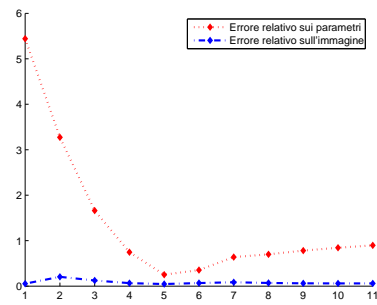


(g) Errore sui parametri: 0.280920.

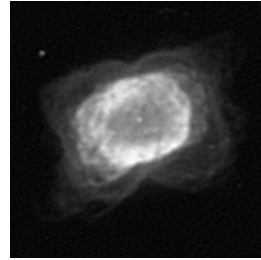


(h) Errore sull'immagine: 0.336920.

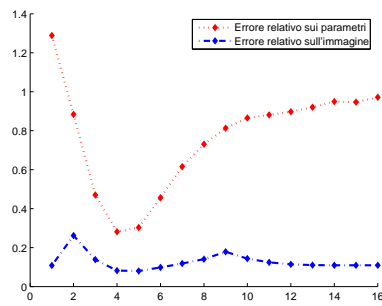
Figura 4.7: Risultati dell'algorithm di blind deconvolution.



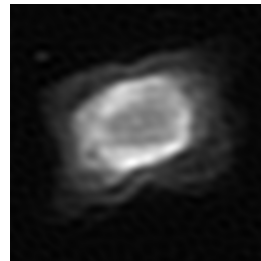
(a) Errore sui parametri: 0.284088.



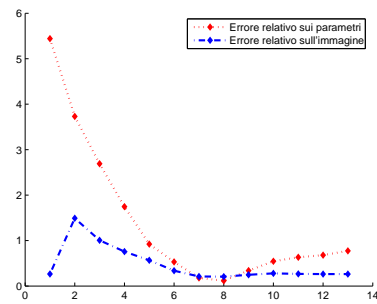
(b) Errore sull'immagine: 0.071855.



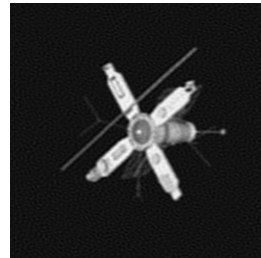
(c) Errore sui parametri: 0.273797.



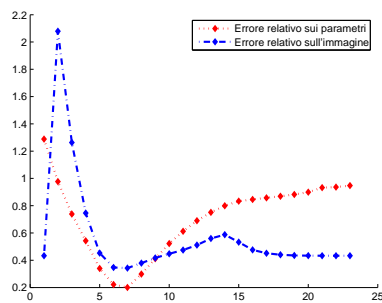
(d) Errore sull'immagine: 0.079495.



(e) Errore sui parametri: 0.117281.



(f) Errore sull'immagine: 0.205487.



(g) Errore sui parametri: 0.214842.



(h) Errore sull'immagine: 0.351817.

Figura 4.8: Risultati dell'algoritmo di blind deconvolution con il metodo di Tikhonov.

Si può notare anche in questo caso che l'algoritmo è molto veloce: in poche iterazioni giunge a risultati soddisfacenti.

Nella figura (4.8) sono posti i risultati ottenuti applicando lo stesso algoritmo ma utilizzando il metodo di Tikhonov sia per risolvere il passo interno che quello esterno. La patina grigia che appare è dovuta alla scalatura dei valori dell'immagine: il metodo di Tikhonov può restituire valori negativi, non avendo posto il vincolo di positività.

Le immagini 4.7(d) e 4.7(h) non sono molto piacevoli da vedere in quanto il blurr presente sull'immagine è molto forte. Un valore numerico che può dare un'indicazione di quanto l'immagine trovata sia simile all'immagine originale rispetto ai parametri visivi dell'occhio umano è l'indice SSIM⁴: nella tabella (4.4) sono presenti i valori di questo indicatore per ogni immagine trovata.

Immagine	Indice SSIM	Errore Relativo
4.7(b)	0.897586	0.047441
4.7(d)	0.869724	0.071480
4.7(f)	0.926884	0.190498
4.7(h)	0.835918	0.336920

Tabella 4.4: Confronto fra indici di bontà dei risultati: l'errore relativo dà un risultato numerico che non sempre coincide con i parametri visivi dell'occhio umano.

Si può notare che nel caso di uno sfocamento più forte, l'indice SSIM è (relativamente) più basso: questo è dovuto al fatto che un blurr così marcato impedisce una ricostruzione dei dettagli più minuti, mentre i dettagli maggiori sono resi più facilmente. Nel caso del satellite, si può notare che ad un errore relativo piuttosto alto (0.19) corrisponde un buon indice SSIM (0.926884).

4.3.2 Rumore poissoniano

Come nel caso precedente, nell'algoritmo di blind deconvolution si è utilizzato come parametro di regolarizzazione α il parametro ottimale ottenuto nel test descritto precedentemente. I risultati sono in figura (4.9) per quanto riguarda la figura 4.5(a).

Il primo fatto da notare è che il risultato migliore (nel senso dell'errore relativo minore) non coincide nel numero di iterazioni: l'aver l'errore più basso sull'immagine non corrisponde con l'aver l'errore più basso sui parametri. Si osserva che comunque la differenza non è molto marcata (tabella (4.5)).

I risultati dell'algoritmo applicato all'immagine 4.5(d) sono mostrati nella prima riga della figura 4.10.

In questo caso al medesimo numero di iterazioni si ottiene il risultato migliore sia per quanto riguarda l'immagine sia per quanto riguarda i parametri.

⁴Vedasi [WBSS]

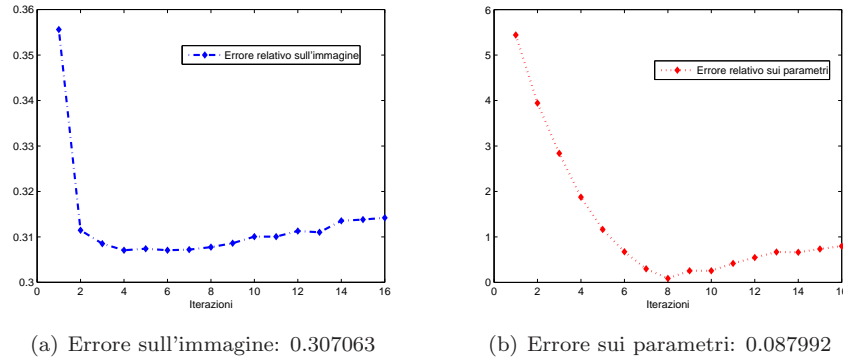


Figura 4.9: Risultati dell'algorithmo di blind deconvolution per l'immagine (4.5(a))

Iter.	E.R.Img	E.R. Par.	Iter.	E.R. Img.	E.R. Par.
1	0.3556	5.4420	9	0.3086	0.2551
2	0.3110	3.9444	10	0.3101	0.2551
3	0.3085	2.8379	11	0.3101	0.4180
4	0.3071	1.8750	12	0.3113	0.5467
5	0.3074	1.1652	13	0.3110	0.6684
6	0.3071	0.6743	14	0.3136	0.6610
7	0.3072	0.2981	15	0.3138	0.7337
8	0.3078	0.0880	16	0.3142	0.7968

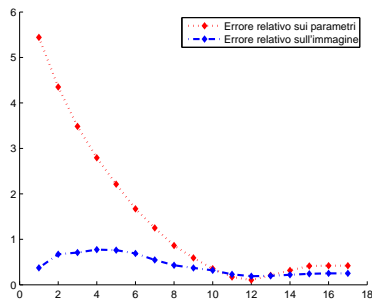
Tabella 4.5: Confronto fra gli errori sull'immagine e sui parametri.

Nella seconda riga della figura (4.10) sono presentati invece i risultati per l'applicazione dell'algorithmo all'immagine perturbata con la Point Spread Function 4.2(b). I risultati visivi non sono soddisfacenti, in quanto la PSF che interviene in questo caso dà uno sfocamento molto incisivo sull'immagine, impedendo, come nel caso gaussiano, una buona ricostruzione dell'immagine⁵ Il fatto da notare con una certa curiosità è che l'errore sui parametri è molto basso nel caso dell'immagine del satellite.

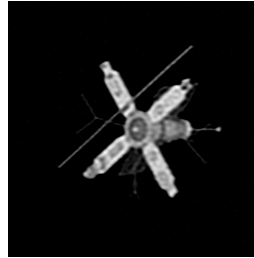
4.4 Conclusioni

I risultati presentati in questo elaborato riguardanti il problema di image deblurring mostrano che l'aggiunta del vincolo di positività nel caso del rumore gaussiano produce risultati migliori rispetto ad un metodo che non presenta questo tipo di vincolo; nel caso del rumore di Poisson vale una conclusione simile: si ottengono buoni risultati sia dal punto di vista numerico (errore relativo) che visivo.

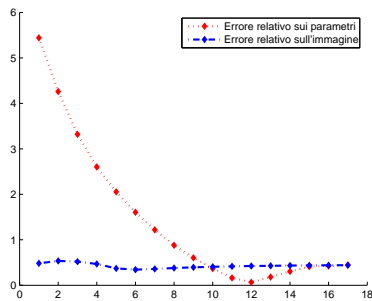
⁵Utilizzando come indicatore della bontà della ricostruzione l'indice SSIM, si ottiene un valore di 0.6181 che è in verità molto basso.



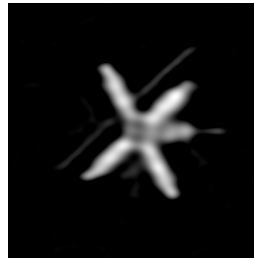
(a) Errore sui parametri:0.105381.



(b) Errore sull'immagine:0.189247.



(c) Errore sui parametri: 0.066080.



(d) Errore sull'immagine: 0.342663.

Figura 4.10: Risultati dell'algoritmo di blind deconvolution per il rumore poissoniano.

Confrontando ancora i risultati ottenuti con il metodo FPQN e con quello di Tikhonov nell'algoritmo di blind deconvolution l'aggiunta del vincolo di positività si rivela ancora molto efficace, sia in presenza di rumore gaussiano che di rumore poissoniano. Nel caso gaussiano, le differenze sono evidenti a occhio nudo: in assenza del vincolo, la presenza di elementi negativi produce un effetto sulla scalatura dei valori dell'immagine che restituisce un risultato falsato a monitor.

Nel caso del rumore poissoniano i risultati ottenuti sull'immagine del satellite sono molto buoni; nel caso di blur poco incisivo, i dettagli sono abbastanza nitidi, mentre un blur molto più marcato induce ovviamente un risultato peggiore ma i particolari maggiori dell'immagine emergono con facilità. L'immagine della nebulosa sembra produrre qualche problema, l'errore relativo minore si ha per un'immagine che sembra essere troppo rumorosa: questo può essere dovuto al comportamento del metodo FPQN nel caso del funzionale Kullbach-Leibler e del tipo di immagine considerata. Questo aspetto richiede uno studio successivo più approfondito, analizzando ad esempio con più cura l'andamento del metodo al variare del parametro di regolarizzazione.

Bibliografia

- [CV] Curtis R. Vogel, *Computational Methods for Inverse Problems*, SIAM
- [HNO] Per Christiam Hansen, James G. Nagy, Dianne P.O'Leary, *Deblurring Images. Matrices, spectra and filtering*, SIAM
- [BP] M. Bertero, P. Boccacci, *Image Deconvolution*
- [BLZ] M. Bertero, Henri Lantéri, Luca Zanni, *Iterative image reconstruction: a point of view*
- [BL] Johnathan M. Bardsley, N'djekornom Laobeul, *Tikhonov Regularized Poisson Likelihood Estimation: Theoretical Justification and a Computational Method*, Inverse Problems in Science and Engineering Vol. 00, No. 00, December 2006, 1-19
- [EHN] H.W. Engl, M. Hanke, A. Neubauer, *Regularization of Inverse Problems*, Kluwer Academic Publishers
- [SR] B. Simon, M. Reed, *Methods of Modern Mathematical Physics, Vol. 1*, Elsevier
- [CN] J. Chung, J.G. Nagy, *Efficient Iterative Approach for Large Scale Separable Nonlinear Inverse Problems*, SIAM J.SCI.COMPUT. Vol.31, No. 6, pp4654-4674
- [LP] G. Landi, E. Loli Piccolomini, *A Fast Projected Quasi-Newton Method for Nonnegative Tikhonov Regularization*
- [EL] E. Lanconelli, *Lezioni di ANALISI MATEMATICA 2 prima parte*, Pitagora Editrice Bologna
- [DB] D. Bertsekas, *Nonlinear Programming*, Athena Scientific
- [WBSS] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, *Image Quality Assessment: From Error Visibility to Structural Similarity*, IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 13, NO. 4, APRIL 2004
- [SVH] D.M. Sima, S. Van Huffel, *Separable nonlinear least square fitting with linear bound constraints and its application in magnetic resonance spectroscopy data quantification*, Journal of Computational and Applied Mathematics 203 (2007) 264-278