

Università degli Studi di Padova
Dipartimento di Scienze Statistiche

Corso di Laurea Magistrale in
Scienze Statistiche



Le basi statistiche della tomografia muonica: aspetti metodologici e computazionali

Relatrice: Prof.ssa Alessandra R. Brazzale
Dipartimento di Scienze Statistiche

Correlatrice: Prof.ssa Giovanna Menardi
Dipartimento di Scienze Statistiche

Laureanda: Marta Ferrari
Matricola n. 2055116

Anno Accademico 2023/2024

Indice

Introduzione	1
1 La ricostruzione delle traiettorie nella tomografia muonica	5
1.1 La tomografia muonica per <i>scattering</i>	5
1.1.1 I muoni	5
1.1.2 L'interazione dei muoni con la materia, tecniche di analisi e applicazioni	7
1.2 Caratterizzazione topografica	9
1.2.1 L'approccio metodologico di riferimento e le sue assun- zioni	9
1.2.2 <i>Scattering</i> all'interno di un volume omogeneo	10
1.2.3 <i>Scattering</i> all'interno di un volume eterogeneo rappre- sentato con voxel 3D	13
1.3 Formalizzazione statistica	15
1.4 Ricostruzione della traiettoria del muone nel volume	19
2 L'attuale modello statistico di riferimento	27
2.1 Specificazione del modello	27
2.1.1 Dati e modello	27
2.1.2 Stima di massima verosimiglianza	29
2.2 Algoritmo EM	31
2.2.1 Definizione	31
2.2.2 Passi E e M	31
3 Il modello statistico proposto	37
3.1 Il modello mistura	37

3.1.1	Formalizzazione statistica della proposta	37
3.1.2	Specificazione del modello	40
3.2	Stima di massima verosimiglianza	42
3.2.1	Verosimiglianza completa e algoritmo EM	42
3.3	Varianti dell'algoritmo EM	45
3.3.1	Lo <i>Stochastic</i> EM	45
3.3.2	I passi dell'algoritmo	47
3.4	Alcuni algoritmi di simulazione	52
3.4.1	Algoritmi MCMC	52
3.4.2	Algoritmi ABC	55
4	Un'applicazione a dati simulati	57
4.1	Descrizione dei dati	57
4.2	Alcune analisi preliminari	59
4.3	Applicazione dei modelli	61
4.3.1	Alcuni dettagli di implementazione	61
4.3.2	Ricostruzione dell'intero tubo isolante	63
4.3.3	Ricostruzione di una porzione del tubo isolante	67
4.4	Discussione	70
4.5	Direzioni future della ricerca	74
	Conclusioni	77
	Bibliografia	79
A	Approfondimenti algebrici	83
A.1	Calcolo della matrice Σ	83
A.2	Calcolo della matrice di covarianza $\Sigma_{D_i H_{ij}}$	85
A.3	Calcolo della matrice $\tilde{\Sigma}_i$	86
A.4	Calcolo della matrice di covarianza $\tilde{\Sigma}_{D_i H_{ij}}$	87
B	Listato R	91
B.1	Preparazione dei dati	91
B.2	Algoritmo EM	97
B.3	Algoritmo SEM e algoritmi di simulazione per il passo S	100

Elenco delle tabelle

4.1	Risultati degli algoritmi EM e SEM per la ricostruzione del volume ristretto.	64
4.2	Risultati degli algoritmi EM e SEM per la ricostruzione della porzione di volume.	68

Elenco delle figure

1.1	Rappresentazione dello <i>scattering</i> in tre dimensioni di un muone che attraversa un volume omogeneo. In blu si osserva la vera traiettoria del muone, in rosa la traiettoria proiettata, ovvero quella che il muone avrebbe se non subisse deviazioni.	11
1.2	Rappresentazione dello <i>scattering</i> lungo la coordinata x di un muone che attraversa un volume omogeneo. In blu si osserva la vera traiettoria del muone, in rosa la traiettoria proiettata, ovvero quella che il muone avrebbe se non subisse deviazioni. .	12
1.3	Rappresentazione dello <i>scattering</i> in tre dimensioni di un muone in un volume eterogeneo ripartito in 3 voxel. In blu si osserva la vera traiettoria del muone, in rosa la traiettoria proiettata, ovvero quella che il muone avrebbe se non subisse deviazioni.	14
1.4	Rappresentazione delle componenti dello <i>scattering</i> lungo la coordinata x di un muone in un volume eterogeneo ripartito in 3 voxel.	15
1.5	Rappresentazione nel piano xz dei possibili modi per approssimare la traiettoria del muone all'interno del volume: in verde la traiettoria rettilinea che congiunge i punti di entrata e uscita, mentre in rosa e blu la traiettoria che interseca il PoCA. . . .	20
1.6	Rappresentazione del volume attraversato da un muone e le quantità necessarie allo sviluppo dell'algoritmo.	21
4.1	Schema illustrativo del volume da ricostruire	58

4.2	Istogramma di frequenza degli angoli di <i>scattering</i> relativi alla coordinata x a sinistra e y a destra.	61
4.3	Istogramma di frequenza dei dislocamenti relativi alla coordinata x a sinistra e y a destra.	62
4.4	Area di base del volume ridotto.	63
4.5	Istogramma di frequenza di λ stimato tramite l'algoritmo EM (in alto) e SEM (in basso), nel caso del volume ridotto.	65
4.6	Sezioni del volume ricostruito, a sinistra tramite l'algoritmo EM e a destra tramite l'algoritmo SEM. In blu i confini del tubo isolante.	66
4.7	Area di base del volume ridotto.	68
4.8	Istogramma di frequenza di λ stimato tramite l'algoritmo EM (in alto) e SEM (in basso), nel caso della porzione di volume.	69
4.9	Sezioni della porzione di volume ricostruita tramite l'algoritmo EM. In blu i confini del tubo isolante.	71
4.10	Sezioni della porzione di volume ricostruita tramite l'algoritmo SEM. In blu i confini del tubo isolante.	72

Introduzione

I muoni sono particelle subatomiche appartenenti alla famiglia dei leptoni e si producono naturalmente nelle interazioni dei raggi cosmici con l'atmosfera terrestre. Grazie alla loro massa superiore rispetto agli elettroni, i muoni hanno un potere penetrante molto elevato e possono attraversare diversi metri di materiale, anche particolarmente denso, prima di essere assorbiti o di decadere. Questa caratteristica li rende particolarmente utili in diversi ambiti scientifici ed applicativi quali, ad esempio, la *tomografia muonica*, che rappresenta l'oggetto di questo lavoro.

La tomografia muonica è una tecnica emergente volta ad identificare l'interno di strutture complesse o inaccessibili attraverso la ricostruzione della traiettoria dei muoni. Essa trova applicazione in una vasta gamma di settori, tra cui la vulcanologia, per mappare l'interno dei vulcani, permettendo di monitorare l'attività magmatica e prevedere potenziali eruzioni; l'archeologia, per investigare l'interno di strutture archeologiche senza necessità di scavi invasivi; la sicurezza, per ispezionare l'eventuale presenza di materiali nucleari o comunque non autorizzati all'interno di volumi monitorati. Un'importante caratteristica di questa tecnica è infatti di non essere nè invasiva nè distruttiva, riuscendo dunque a investigare l'interno di volumi ignoti senza doverli danneggiare o nuocendo alla salute di esseri viventi.

La tomografia muonica si basa sul principio che i muoni, interagendo con i nuclei degli atomi dei materiali attraversati, subiscono deviazioni (*scattering*) che dipendono dalla densità e dalla composizione del materiale. Misurando l'angolo di *scattering* e l'energia residua dei muoni prima e dopo aver attraversato un oggetto, è possibile ricostruire un'immagine tridimensionale della distribuzione di densità all'interno dell'oggetto stesso. L'uso di tecniche statistiche e algoritmi appropriati diventa pertanto necessario per una accurata

ricostruzione. In questa tesi si fa riferimento alla metodologia proposta da Schultz et al. (2007), che propongono di ricostruire la composizione interna di un volume mediante la stima di massima verosimiglianza dei parametri della distribuzione degli angoli di *scattering* e del dislocamento dei muoni che attraversano il volume. Tuttavia, se è possibile registrare la posizione e l'angolo di entrata e uscita dei muoni dal volume, la traiettoria interna ad esso non è osservabile. Per questa ragione, gli autori propongono una opportuna applicazione dell'algoritmo *Expectation Maximization* (EM), volto alla massimizzazione di una verosimiglianza statistica in presenza di dati mancanti. Assumendo una distribuzione Gaussiana per l'angolo di *scattering* e per il dislocamento dei muoni, ad ogni passo dell'algoritmo EM l'espressione di stima del parametro di interesse è esplicita. Inoltre, in relazione ad altri metodi operativi allo stato dell'arte, come quello di Newton, l'algoritmo EM dimostra essere maggiormente adattabile nel caso di geometrie complesse.

Questo lavoro ha come scopo esplorare un'alternativa al modello di riferimento specificato da Schultz et al. (2007), poichè è noto dalla letteratura (Lynch e Dahl (1990)) che la distribuzione generatrice degli angoli di *scattering* presenta code più pesanti di una Gaussiana. Si propone in alternativa una distribuzione mistura di due Gaussiane aventi la stessa media e varianze differenti. Il problema principale legato a questa specificazione alternativa è che la funzione ausiliaria da massimizzare per la stima dei parametri comporta una importante difficoltà di calcolo. Tale difficoltà viene affrontata sviluppando alcune varianti, alternative all'algoritmo EM, che fanno ricorso ad un algoritmo di simulazione (*Stochastic EM*, SEM): alcuni esempi proposti e discussi sono il *Metropolis-Hastings* e il *Gibbs sampler* (Zanghi et al. (2008)).

Oltre ad una dettagliata descrizione del metodo di riferimento e del nuovo modello proposto, per i quali vengono sviluppati i calcoli per la determinazione delle quantità di interesse e implementati gli algoritmi di stima, nella tesi si presenta un'applicazione pratica delle metodologie a un insieme di dati simulati, per poter effettuare un confronto sull'efficacia e sull'accuratezza e il costo computazionale degli algoritmi EM e SEM, così come per gli algoritmi di simulazione proposti. Nello specifico, i dati simulati riguardano un esperimento effettuato presso l'Università di Cantabria al Dipartimento di

Fisica Moderna. Lo scopo dello studio è di valutare il livello di usura nelle pareti interne di un tubo isolante tramite la ricostruzione dei materiali che ne compongono gli strati con i relativi spessori.

L'elaborato si sviluppa nel seguente modo. Nel Capitolo 1 si espongono i concetti base della tomografia muonica per *scattering* e si impostano le basi topografiche e statistiche dei metodi descritti nei capitoli seguenti. Nel Capitolo 2 si presenta il metodo di riferimento sviluppato da Schultz et al. (2007): si specifica il modello e si espone l'approccio di stima attraverso l'algoritmo EM. Nel Capitolo 3 si espone ed elabora la modifica alla metodologia di riferimento, specificando un modello mistura a due componenti Gaussiane e si sviluppa il procedimento di stima di massima verosimiglianza tramite l'algoritmo SEM. Si espongono inoltre alcuni algoritmi di simulazione necessari per il passo S del SEM. Nel Capitolo 4 infine si presentano e discutono i risultati dell'applicazione delle metodologie descritte all'insieme di dati simulati. Il lavoro si conclude con alcune considerazioni finali.

Capitolo 1

La ricostruzione delle traiettorie nella tomografia muonica

In questo capitolo si espongono i concetti base della tomografia muonica per *scattering*, includendo una panoramica sui muoni e le loro principali proprietà. In questa prima parte si fa riferimento ad alcuni lavori: Schultz (2003) e Morris et al. (2008). Successivamente si impostano le basi dei metodi descritti nei capitoli seguenti, in particolare si forniscono i dettagli geometrici e fisici essenziali alle metodologie, facendo riferimento in particolare al metodo di Schultz et al. (2007).

1.1 La tomografia muonica per *scattering*

1.1.1 I muoni

La tomografia è una tecnica volta alla ricostruzione di un'immagine della struttura interna di un oggetto tridimensionale, tramite la visualizzazione di sue sezioni trasversali, senza la necessità di tagliarlo fisicamente.

Questa metodologia è utilizzata in svariati campi. Ad esempio, in ambito medico ci si avvale di questa tecnica per ottenere immagini dettagliate del corpo umano per la diagnosi di tumori, fratture ossee, malattie cardiovascolari o lesioni cerebrali. Dal punto di vista ingegneristico la tomografia permette ad esempio il controllo dell'integrità di edifici e la rilevazione di difetti interni

alle strutture. In geologia, grazie alle tecniche tomografiche è possibile esplorare il sottosuolo, ottenendo immagini dettagliate della struttura interna della Terra, utili alla ricerca di risorse minerarie e petrolifere; aiuta inoltre a comprendere la propagazione delle onde sismiche e a identificare faglie e zone di frattura. Un ultimo esempio riguarda l'archeologia, che fa uso della tomografia per indagare la struttura sotterranea di tombe, muri e antiche costruzioni senza dover effettuare scavi invasivi.

La tomografia muonica per *scattering* si basa in particolare sullo studio delle traiettorie dei muoni, particelle prodotte dai raggi cosmici che costantemente colpiscono l'atmosfera. I raggi cosmici sono costituiti principalmente da protoni, che interagiscono con l'alta atmosfera producendo cascate di particelle, tra cui i pioni, i quali decadono e producono muoni. Essi hanno una massa a riposo di circa 106 MeV e una vita media di 2.2 microsecondi.

I muoni piovono incessantemente sulla superficie terrestre, più precisamente il loro flusso a livello del mare è pari a 1 muone al minuto per cm^2 ; la nostra mano è attraversata da un muone al secondo e il nostro intero corpo da 600 muoni al minuto. Questo tasso è abbastanza basso da permettere di distinguere, attraverso specifici rilevatori, il passaggio di ogni singolo muone ed è allo stesso tempo abbastanza alto per produrre l'informazione necessaria allo scopo tomografico in un ragionevole intervallo di tempo.

I muoni che cadono dall'atmosfera hanno in genere un'energia di qualche GeV e tra loro spiccano muoni ad alta energia, cioè maggiore di 100 GeV . In ogni caso, anche un muone di soli 3 GeV riesce a penetrare più di 1000 g/cm^2 di materiale ed è questo potere penetrativo a renderli convenienti per la ricostruzione di volumi internamente ignoti. Inoltre, queste particelle perdono energia più lentamente rispetto ad altre particelle e riescono ad attraversare oggetti molto densi e spessi, come la roccia e il cemento. In aggiunta risultano essere completamente innocue per gli esseri viventi che ne vengono attraversati, così come non danneggiano gli oggetti che incrociano lungo la loro traiettoria. Queste ultime peculiarità non sono condivise ad esempio dai raggi X, anch'essi strumento delle tecniche tomografiche. Si rimanda per maggiori dettagli a Roentgen (1989).

Per tali metodologie sono stati presi in considerazione anche i protoni che, pur riuscendo a penetrare oggetti molto densi, risultano però pericolosi per

gli esseri umani e possono deteriorare oggetti molto delicati, come le pellicole fotografiche.

La tomografia muonica, oltre a risultare una tecnica non nociva e non distruttiva, è anche poco costosa in quanto ricorre a particelle che già naturalmente cadono sulla superficie terrestre.

Il passaggio dei muoni può essere osservato grazie a rilevatori, che possono essere lastre di emulsione fotografica, rilevatori a gas o contatori di scintillazione. I rilevatori sono posizionati all'esterno o, se possibile, all'interno del volume da ispezionare, e registrano la posizione e l'energia dei muoni che li attraversano.

Per l'esplorazione di grandi volumi, come montagne o vulcani, è possibile registrare il passaggio dei muoni tramite rilevatori posti lateralmente al volume; i muoni infatti raggiungono la superficie terrestre con diverse angolazioni e talvolta sono quasi orizzontali al suolo, intersecando così questi grandi oggetti lateralmente anzichè verticalmente.

Spesso ai lati opposti del volume sono collocate più file di rilevatori, così da rendere possibile il calcolo di alcune quantità necessarie ai metodi tomografici, quali l'angolo della traiettoria del muone e il suo momento (Hogan et al. (2004)).

Altri dettagli tecnici sulla costruzione di un rilevatore sono esplorati ad esempio in Green et al. (2006).

1.1.2 L'interazione dei muoni con la materia, tecniche di analisi e applicazioni

I muoni sono stati sfruttati per le loro proprietà già dalla metà del secolo scorso nella cosiddetta *Cosmic Ray Muon Radiography*; in questo contesto i muoni sono trattati a scopi radiografici come alternativa ai raggi X, riuscendo però a investigare volumi molto più imponenti. In particolare nella radiografia muonica ci si avvale della caratteristica dei muoni di perdere energia interagendo con i materiali. Più il materiale è denso o spesso, più assorbe o devia i muoni che lo attraversano. Si può quindi affermare che ci sia una relazione tra la densità e lo spessore del materiale e l'energia persa dal muone che lo attraversa.

Se i muoni vengono frenati o perdono molta energia attraversando un materiale, non lasciano traccia sul rilevatore, poichè questo è stato costruito in modo tale da processare i muoni solo sopra una certa energia. Dunque studiando le differenze in energia dei muoni e il loro conteggio, si possono riconoscere le diverse densità della struttura interna del volume, dalle zone meno dense a quelle cave.

La prima applicazione della radiografia muonica venne svolta da George (1955) per determinare la profondità di una roccia sopra un tunnel sotterraneo. Un altro studio degno di nota è quello svolto da Alvarez et al. (1970) per la ricerca di cripte nella piramide di Chefren: fu possibile ottenere informazioni dettagliate sulla struttura interna della piramide, altrimenti difficili o impossibili da raccogliere con metodi tradizionali di esplorazione archeologica. Altri lavori riguardano l'ispezione di vulcani e montagne, come ad esempio il monte Tsukuba in Nagamine et al. (1995).

La *Muon scattering Tomography* (MTS) invece è stata introdotta successivamente da Borozdin et al. (2003) e non si basa sulla perdita di energia dei muoni, bensì sullo *scattering* di Coulomb che essi subiscono mentre attraversano il volume da ispezionare.

I muoni interagiscono principalmente attraverso la forza di Coulomb con i nuclei delle particelle che costituiscono la materia attraversata. La forza di Coulomb non solo fa sì che il muone decelerì e perda energia, ma ne devia anche la traiettoria. In particolare, il muone che attraversa uno specifico materiale subisce il processo di *scattering* multiplo di Coulomb e la sua traiettoria subisce tante deflessioni di piccola entità. È quindi il materiale a determinare la deviazione totale del muone. Materiali caratterizzati da un alto numero atomico Z , come il tungsteno e il piombo, deviano in maniera importante la traiettoria del muone. Invece materiali definiti a medio Z , come l'acciaio e il ferro, e a basso Z , come il carbone e l'alluminio, provocano una deflessione minore al percorso del muone.

L'idea della tomografia muonica per *scattering* è dunque quella di porre dei rilevatori agli estremi opposti del volume da ispezionare e, tramite essi, tracciare la posizione e l'angolo dei muoni. Dalle misure ottenute si può derivare numericamente quanto ogni muone è stato deviato rispetto alla sua traiettoria in entrata dal volume. Lo scopo della tomografia si traduce quindi nell'analisi

dello *scattering* dei muoni rispetto alla loro traiettoria originale e le informazioni risultanti vengono utilizzate per stimare le proprietà del materiale.

Nello specifico, studiando il percorso di ogni muone all'interno del volume da ispezionare, si può stimare in quali zone sono stati scatterati i muoni e con quale magnitudine. Si associa dunque a materiali più densi uno *scattering* di maggiore entità.

La tomografia muonica risulta conveniente quando si lavora con volumi molto grandi in cui si vogliono localizzare oggetti ad alta densità, ma risulta efficace anche per studiare nel dettaglio la composizione eterogenea di oggetti complessi. Ad esempio, la tomografia muonica si è molto diffusa nell'ambito della sicurezza per ispezionare il contenuto di container e cargo, per rilevare la presenza di materiali radioattivi all'interno di depositi o camion senza correre così il rischio di esposizione a radiazioni, e per misurare con molta precisione la struttura di edifici per valutarne la stabilità.

1.2 Caratterizzazione topografica

1.2.1 L'approccio metodologico di riferimento e le sue assunzioni

Per la ricostruzione di un volume ignoto tramite l'analisi delle traiettorie dei muoni, sono stati sviluppati diversi metodi statistici a partire da una caratterizzazione topografica del volume da investigare. In particolare, tale volume viene suddiviso in voxel, ovvero pixel in versione tridimensionale. Per ognuno di essi si vuole stimare la densità del materiale che lo compone e, rappresentando visivamente i risultati di tutti i voxel riuniti nel volume, si può osservare l'immagine tridimensionale ricostruita.

A partire dalla caratterizzazione geometrica del problema, si specifica il modello statistico per lo *scattering* dei muoni. Per la ricostruzione di immagini, soprattutto in campo medico come per esempio nelle *Positron emission tomography* (PET) e *Single-photon emission computed tomography* (SPECT), viene generalmente utilizzata la stima di massima verosimiglianza come metodo di stima dei parametri di interesse. Schultz (2003) ricorre al metodo di Newton per la massimizzazione della verosimiglianza: questo è risultato

adatto solo nel caso in cui l'oggetto da rilevare all'interno del volume sia geometricamente semplice.

In questa tesi si fa invece riferimento all'approccio proposto da Schultz et al. (2007) per la ricostruzione del contenuto ignoto di un volume, che può essere anche complesso a livello geometrico. Questo metodo propone inizialmente la specificazione del modello per lo *scattering* dei muoni all'interno di un volume e ricorre alla stima di massima verosimiglianza per la stima in ogni voxel del parametro di interesse, la cosiddetta *densità di scattering* del materiale, quantità che verrà illustrata nel dettaglio successivamente. L'ottimizzazione della funzione di verosimiglianza viene effettuata per mezzo dell'algoritmo EM.

Il metodo proposto da Schultz et al. (2007) è stato delinato per angoli di *scattering* e dislocamenti di piccola entità. Infatti, alcune quantità necessarie sono state calcolate approssimando la traiettoria interna al volume con una linea retta, oppure tramite il punto *PoCA* che verrà introdotto nel dettaglio nel paragrafo 1.4. Se gli eventi di *scattering* fossero però molto ampi, la traiettoria del muone si discosterà molto da quella approssimata e ciò comporterà in un errato procedimento di stima della densità di *scattering*.

Inoltre, il dislocamento viene definito in funzione dello *scattering* angolare ignorando i termini di derivazione di secondo ordine: questi termini sarebbero invece risultati significativi nel caso di scattering più importanti.

Una seconda considerazione riguarda i rilevatori che registrano il passaggio dei muoni. A queste strumentazioni è associato un grado di precisione nelle misurazioni, dovuto alla risoluzione con cui queste vengono processate. L'errore di misurazione causato dal rilevatore si propaga dunque nel valore calcolato dello *scattering* del muone. In questa metodologia si assume che i rilevatori siano perfetti, dunque non causino un errore di misura, come può capitare nelle applicazioni dei metodi a dati simulati.

1.2.2 *Scattering* all'interno di un volume omogeneo

Si consideri un singolo muone e la sua traiettoria nello spazio tridimensionale (x, y, z) . Le coordinate x e z sono perpendicolari e si estendono rispettivamente in orizzontale e in verticale, mentre l'asse y è perpendicolare

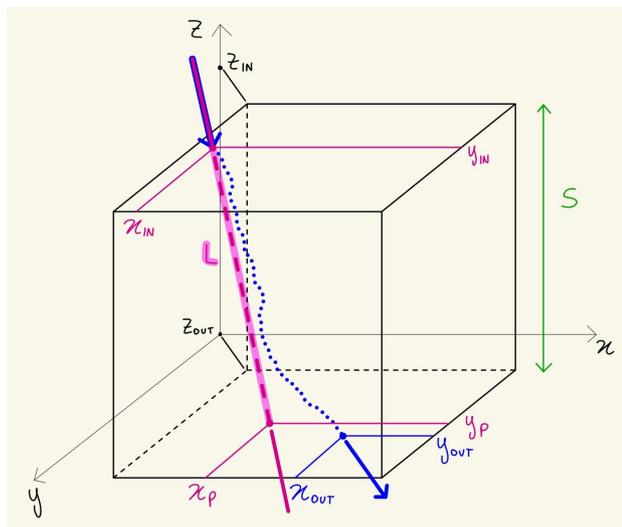


Figura 1.1: Rappresentazione dello *scattering* in tre dimensioni di un muone che attraversa un volume omogeneo. In blu si osserva la vera traiettoria del muone, in rosa la traiettoria proiettata, ovvero quella che il muone avrebbe se non subisse deviazioni.

al piano xz . La traiettoria del muone incide sul piano xy coincidente con l'estremo superiore del volume da investigare e si sviluppa per valori minori di z fino a intersecare il piano xy che combacia con la faccia inferiore del volume.

Come delineato in precedenza, la tomografia muonica utilizza dei rilevatori che permettono di registrare le seguenti quantità derivanti dall'intersezione della traiettoria del muone e il volume da ispezionare: le posizioni (x_{IN}, y_{IN}, z_{IN}) e $(x_{OUT}, y_{OUT}, z_{OUT})$ e gli angoli di incidenza $(\theta_{x,IN}, \theta_{y,IN})$ e $(\theta_{x,OUT}, \theta_{y,OUT})$, rispettivamente in entrata e in uscita dal volume. Gli angoli vengono misurati secondo le prospettive determinate dai piani xz e yz .

Si esamina inizialmente la situazione più semplice, ovvero il caso in cui il muone attraversi un volume composto da un unico specifico materiale, dunque omogeneo (si veda (1.1)). La lunghezza della linea retta definita tra i punti (x_{IN}, y_{IN}, z_{IN}) e (x_p, y_p, z_p) descrive la traiettoria tridimensionale che il muone percorre attraverso il volume senza subire deviazioni, si veda Figura 1.2, ed è pari a

$$L = s \sqrt{1 + \tan^2_{\theta_{x,IN}} + \tan^2_{\theta_{y,IN}}} = sl_{xy}, \quad (1.1)$$

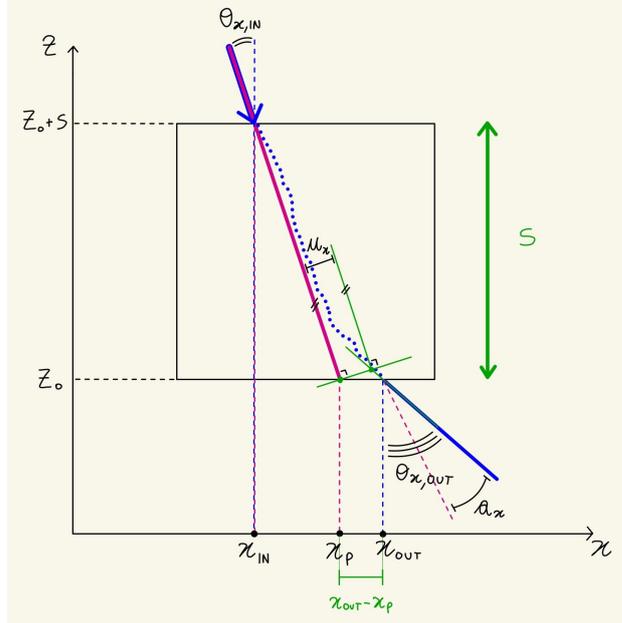


Figura 1.2: Rappresentazione dello *scattering* lungo la coordinata x di un muone che attraversa un volume omogeneo. In blu si osserva la vera traiettoria del muone, in rosa la traiettoria proiettata, ovvero quella che il muone avrebbe se non subisse deviazioni.

dove s misura l'altezza del volume e si chiama $l_{xy} = \sqrt{1 + \tan^2_{\theta_{x,IN}} + \tan^2_{\theta_{y,IN}}}$.

Nell'eventualità in cui il muone subisca uno *scattering*, il suo punto di uscita $(x_{OUT}, y_{OUT}, z_{OUT})$ non coincide con il punto (x_p, y_p, z_p) . In questo contesto, si definiscono a_x e a_y come le misure dello *scattering* angolare lungo ciascuna coordinata; esse rappresentano la differenza tra l'angolo di entrata e di uscita del muone dal volume rispetto al piano xz e il piano yz

$$\begin{aligned} a_x &= \theta_{x,OUT} - \theta_{x,IN} \\ a_y &= \theta_{y,OUT} - \theta_{y,IN}. \end{aligned} \tag{1.2}$$

Allo stesso modo si ottengono le misure di dislocamento, ovvero la differenza lungo ogni coordinata tra il punto (x_p, y_p, z_p) e la posizione reale di uscita

del muone

$$\begin{aligned} u_x &= (x_{OUT} - x_p) \cos(\theta_{x,IN}) l_{xy} \frac{\cos(a_x + \theta_{x,IN})}{\cos(a_x)} \\ u_y &= (y_{OUT} - y_p) \cos(\theta_{y,IN}) l_{xy} \frac{\cos(a_y + \theta_{y,IN})}{\cos(a_y)}, \end{aligned} \quad (1.3)$$

dove i due termini centrali di ogni formula correggono le differenze $(x_{OUT} - x_p)$ e $(y_{OUT} - y_p)$ per la lunghezza della traiettoria tridimensionale e il termine finale proietta il risultato nel piano ortogonale alla traiettoria del muone.

Si sottolinea che il dislocamento rispetto alla coordinata z non viene calcolato. I suoi valori, che individuano i limiti inferiore e superiore del volume in studio, $z = z_0$ e $z = z_0 + s$, rimangono invariati indipendentemente dallo *scattering* del muone che interessa unicamente le variazioni rispetto alle coordinate x e y . Un ulteriore aspetto da mettere in risalto è il fatto che le misure dello *scattering* lungo le coordinate x e y , tra loro ortogonali, sono state effettuate in modo indipendente.

1.2.3 *Scattering* all'interno di un volume eterogeneo rappresentato con voxel 3D

Si estende la caratterizzazione precedente all'eventualità che il volume da ispezionare non sia omogeneo. In questo contesto lo *scattering* totale del muone deriva dalla combinazione degli *scattering* avvenuti singolarmente in ogni specifica regione del volume, ognuna composta da un diverso materiale. Per studiare la composizione di un volume potenzialmente eterogeneo, esso viene suddiviso in voxel, ovvero unità elementari tridimensionali: in Figura 1.3 è rappresentato un esempio in cui la traiettoria di un muone incrocia un volume ripartito in 3 voxel.

Per semplicità si assume di seguito che ogni voxel abbia una forma cubica di lato s ; con alcune complicazioni matematico/geometriche sono comunque possibili ripartizioni di forme diverse.

Per generalizzare quanto esposto nel paragrafo 1.2.2, si considera il volume omogeneo di altezza s come un singolo voxel, il quale è appunto composto da un unico materiale.

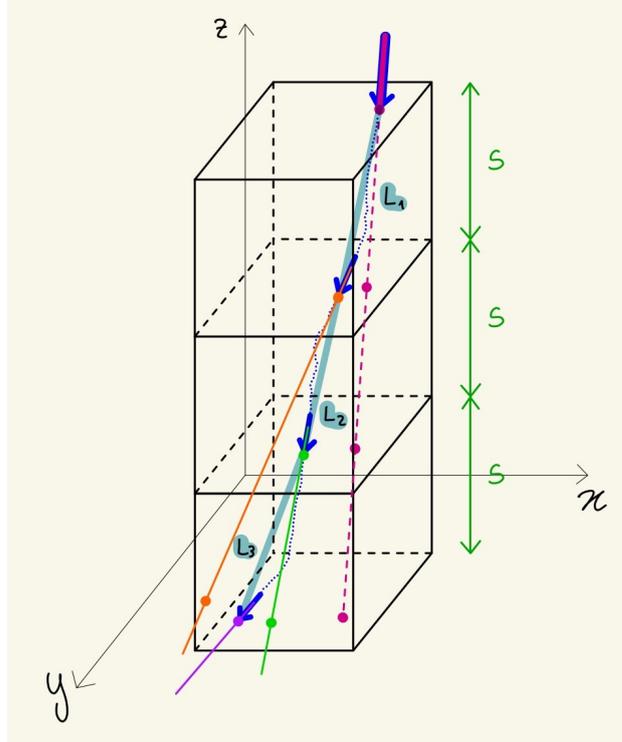


Figura 1.3: Rappresentazione dello *scattering* in tre dimensioni di un muone in un volume eterogeneo ripartito in 3 voxel. In blu si osserva la vera traiettoria del muone, in rosa la traiettoria proiettata, ovvero quella che il muone avrebbe se non subisse deviazioni.

Si consideri più in generale un volume suddiviso verticalmente in M voxel (si veda Figura 1.4): all'interno di ogni voxel j il muone subisce un dislocamento $(u_{x,j}, u_{y,j})$ e un angolo di *scattering* $(a_{x,j}, a_{y,j})$. Le misure di *scattering* totali (u_x, u_y) e (a_x, a_y) , che rappresentano lo *scattering* del muone dopo aver attraversato l'intero volume, si ottengono con

$$a_x = \sum_{j=1}^M a_{x,j}, \quad a_y = \sum_{j=1}^M a_{y,j}; \quad (1.4)$$

$$u_x \approx \sum_{j=1}^M (u_{x,j} + T_j a_{x,j}), \quad u_y \approx \sum_{j=1}^M (u_{y,j} + T_j a_{y,j}). \quad (1.5)$$

L'approssimazione per il calcolo dei dislocamenti espressi in (1.5) è valida sotto l'assunzione di *scattering* di piccola entità.

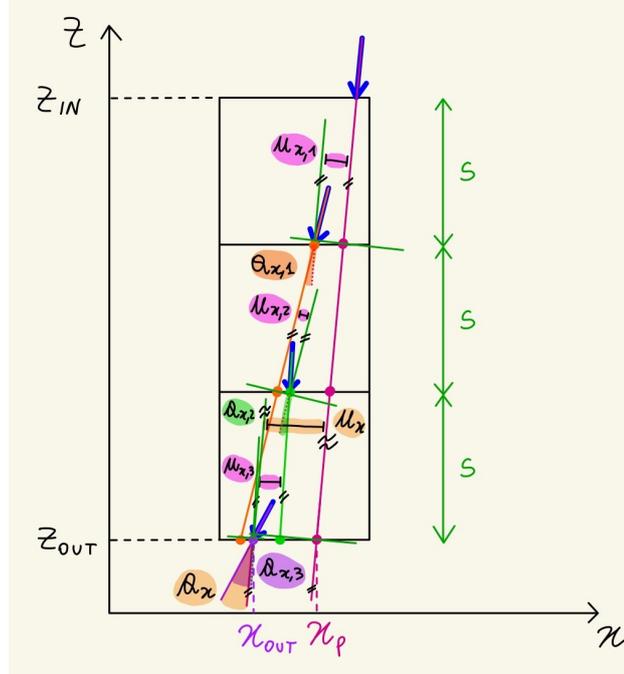


Figura 1.4: Rappresentazione delle componenti dello *scattering* lungo la coordinata x di un muone in un volume eterogeneo ripartito in 3 voxel.

La quantità T_j calcola la lunghezza della traiettoria tridimensionale percorsa dal muone, dal punto di uscita dal j -esimo voxel al punto di uscita dal volume totale, e si esprime come

$$T_j = \sum_{h=j+1}^M L_h \quad 1 \leq j \leq M-1 \quad (1.6)$$

$$T_M = 0,$$

dove L_j misura la traiettoria tridimensionale percorsa dal muone all'interno del j -esimo voxel, quantità che è pari a 0 per i voxel non colpiti dal raggio.

1.3 Formalizzazione statistica

Si consideri inizialmente lo *scattering* del muone che attraversa o un volume omogeneo o un singolo voxel di un volume eterogeneo, dunque un unico materiale. Per il momento si fa riferimento a una deviazione bidimensionale,

ovvero rispetto unicamente alla coordinata x .

In letteratura è uso approssimare la variabile aleatoria che genera gli angoli di *scattering* con una distribuzione Gaussiana di media 0

$$p_A(a) \cong \frac{1}{\sigma_a \sqrt{2\pi}} \exp \left\{ -\frac{a^2}{2\sigma_a^2} \right\}. \quad (1.7)$$

La deviazione standard σ_a della distribuzione di A in (1.7) viene espressa in funzione di alcune proprietà del materiale:

$$\sigma_a \cong \frac{15 \text{ MeV}}{\beta c p} \sqrt{\frac{s}{r}}. \quad (1.8)$$

Nel dettaglio, gli elementi coinvolti nella definizione di σ_a in (1.8) rappresentano:

- c , la velocità della luce;
- β , la velocità del muone nel vuoto in rapporto a quella della luce, assumendo $\beta c \cong 1$;
- p , il momento del muone in MeV/c ;
- s , l'ampiezza del materiale in cm ;
- r , la lunghezza di radiazione del materiale in g/cm^2 , pari alla distanza dopo la quale l'energia dell'elettrone si riduce mediamente di $1/e$ rispetto a quella iniziale.

La quantità di interesse, ai fini dell'intero processo, è rappresentata dalla densità di *scattering* λ , espressa in mrad^2/cm , che caratterizza il materiale attraverso il quale passano i muoni e rappresenta l'angolo di *scattering* quadratico medio tra tutti i muoni con momento nominale p_0 che attraversano una profondità unitaria dello specifico materiale

$$\lambda(r) = \left(\frac{15}{p_0} \right)^2 \frac{1}{r}.$$

La varianza (1.8) può essere allora espressa come

$$\begin{aligned} \sigma_a^2 &= \frac{15^2}{\beta^2 c^2 p^2} \frac{s}{r} = \frac{15^2}{\beta^2 c^2 p^2} \frac{s \lambda p_0^2}{15^2} \\ &= s \lambda \left(\frac{p_0}{p} \right)^2 = s \lambda p_r^2, \end{aligned} \quad (1.9)$$

dove $p_r^2 = \left(\frac{p_0}{p}\right)^2$.

È noto da Rossi (1952) che la lunghezza di radiazione r diminuisce monotonicamente all'aumentare del numero atomico Z e della densità del materiale. La densità di *scattering* λ cresce in maniera non monotona con l'aumento del numero atomico Z , ma appare comunque evidente la differente densità di *scattering* indotta da materiali a basso, medio e alto numero atomico. Dunque, essenzialmente, la densità di *scattering* fornisce indicazioni accurate sul materiale attraversato dal muone.

Oltre all'angolo di *scattering*, per il muone si calcola anche il dislocamento u : questo si considera correlato all'angolo di *scattering* a e dunque essi congiuntamente danno informazioni sulla posizione locale dello *scattering* del muone nel volume. Si chiami U la distribuzione dei dislocamenti dovuti allo *scattering* del muone: essa è una Gaussiana di media 0 e varianza σ_u^2

$$\sigma_u^2 = \frac{s^2}{3}\sigma_a^2 = \frac{s^3}{3}\lambda p_r^2. \quad (1.10)$$

La covarianza tra l'angolo di *scattering* a e il dislocamento u è pari a

$$\sigma_{au} = \frac{s}{2}\sigma_a^2 = \frac{s^2}{2}\lambda p_r^2. \quad (1.11)$$

La distribuzione congiunta (A, U) è ben caratterizzata da una Gaussiana bivariata di media 0 e matrice di covarianza

$$\Sigma_{au} = \begin{bmatrix} \sigma_a^2 & \sigma_{au} \\ \sigma_{au} & \sigma_u^2 \end{bmatrix} = \lambda B p_r^2, \quad (1.12)$$

con

$$B = \begin{bmatrix} s & s^2/2 \\ s^2/2 & s^3/3 \end{bmatrix}. \quad (1.13)$$

Nel contesto più reale, in cui il muone subisce *scattering* anche lungo la coordinata y , si distinguono gli angoli di *scattering* a_x e a_y e i dislocamenti u_x e u_y . La matrice dei pesi di covarianza in (1.13) deve essere quindi ridefinita nel contesto di *scattering* in tre dimensioni, facendo riferimento alla traiettoria tridimensionale del muone

$$B = \begin{bmatrix} L & L^2/2 \\ L^2/2 & L^3/3 \end{bmatrix}.$$

Le misure dello *scattering* lungo ogni coordinata sono indipendenti e identicamente distribuite (Eidelman (1952)) e dunque tutti i risultati ottenuti per la variabile (A_x, U_x) si applicano anche alla distribuzione (A_y, U_y) , tra cui ad esempio l'espressione della matrice di covarianza Σ_{au} .

Si estende ora l'ultimo sviluppo in tre dimensioni nel caso più generale di un volume eterogeneo suddiviso in M voxel tridimensionali composti ognuno di materiale omogeneo. Le variabili aleatorie $(A_{x,j}, U_{x,j})$, assunte tra loro indipendenti (Schultz et al. (2007)), descrivono lo *scattering* nel j -esimo voxel, per ogni $j = 1, \dots, M$, e hanno matrice di covarianza

$$\Sigma_j = \lambda_j B_j p_r^2, \quad (1.14)$$

con

$$B_j = \begin{bmatrix} L_j & L_j^2/2 \\ L_j^2/2 & L_j^3/3 \end{bmatrix}. \quad (1.15)$$

La distribuzione congiunta di (A_x, U_x) relativa allo *scattering* del muone in seguito al passaggio dell'intero volume è sempre una Gaussiana bidimensionale di media 0 e matrice di covarianza Σ che, combinando le equazioni (1.4), (1.5) e (1.12) e in virtù dell'indipendenza,

$$\Sigma = p_r^2 \sum_{j=1}^M \lambda_j W_j, \quad (1.16)$$

dove W_j è la matrice dei pesi

$$W_j = \begin{bmatrix} L_j & L_j^2/2 + L_j T_j \\ L_j^2/2 + L_j T_j & L_j^3/3 + L_j^2 T_j + L_j T_j^2 \end{bmatrix}. \quad (1.17)$$

Per maggiori dettagli si veda l'Appendice A.1.

Ogni voxel j è caratterizzato dunque da una specifica densità di *scattering* λ_j , per $j = 1, \dots, M$: il parametro M -dimensionale $\lambda = (\lambda_1, \dots, \lambda_M)$ è proprio il parametro di interesse che si vuole stimare.

Ottenuta una stima per il valore di λ , il volume da investigare può essere raffigurato tramite sue sezioni. Ognuna di esse è un'immagine grezza composta da voxel rappresentati in base al loro valore stimato di densità di *scattering*.

1.4 Ricostruzione della traiettoria del muone nel volume

Ai fine della ricostruzione del volume di interesse, è necessario fare un'ipotesi sulla traiettoria tridimensionale di ciascun muone all'interno dello stesso, ovvero sulla determinazione dei voxel attraversati dal singolo muone e sulla conseguente lunghezza del percorso. Tale assunzione è necessaria perchè la definizione della distribuzione dello *scattering*, in particolare la sua varianza, coinvolge le misure L e T .

Una procedura accurata volta a questo scopo è nota come *Point of closest approach* (PoCA, Schultz et al. (2004)). Si considerino la traiettoria rettilinea che il muone mantiene se non viene deviato e la traiettoria delineata dall'angolo in uscita del muone dal volume, rette queste rappresentate rispettivamente in rosa e in blu in Figura 1.5. Il punto di incrocio tra queste due rette è definito PoCA. In questo caso la traiettoria che si assegna al muone è l'unione del segmento che va dal punto di entrata nel voxel al punto PoCA con il segmento tra il PoCA e il punto di uscita. Sebbene non sia complesso trovare il punto di incrocio tra due rette incidenti o il punto più vicino tra due rette sghembe nello spazio tridimensionale, il calcolo si tramuta in un complicato problema di ottimizzazione numerica quando questo punto deve essere individuato all'interno di un volume delimitato; al riguardo sono in via di sviluppo diversi algoritmi, come ad esempio Strong et al. (2023).

In questo lavoro si utilizzerà pertanto un approccio più semplice, proposto da Siddon (1985) per ricostruire la traiettoria rettilinea di un raggio che attraversa un volume. Nel dettaglio, la traiettoria del muone all'interno del volume viene delineata tracciando una linea retta dal punto di entrata P_{IN} al punto di uscita P_{OUT} , in verde in Figura 1.5. È noto infatti che lo *scattering* angolare misura poche decine di milliradiani e di conseguenza la traiettoria di un muone che subisce uno o più eventi di *scattering* non si discosta particolarmente dalla traiettoria che avrebbe mantenuto se non fosse stato deviato.

Si presenta l'algoritmo adattandolo a questo contesto della tomografia muonica, apportando alcune modifiche ritenute necessarie per l'applicazione pratica dei metodi che saranno discussi nei capitoli seguenti.

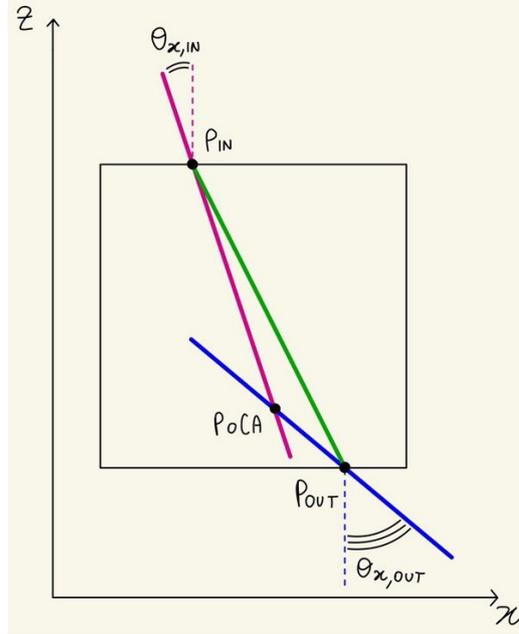


Figura 1.5: Rappresentazione nel piano xz dei possibili modi per approssimare la traiettoria del muone all'interno del volume: in verde la traiettoria rettilinea che congiunge i punti di entrata e uscita, mentre in rosa e blu la traiettoria che interseca il PoCA.

In Figura 1.6 è rappresentato il volume attraversato dalla traiettoria con la notazione utilizzata nello sviluppo del metodo.

Sia suddiviso il volume in studio in $(N_x - 1, N_y - 1, N_z - 1)$ voxel cubici di lato s : questi sono definiti dall'intersezione di un insieme di piani paralleli, ortogonali ed equispaziati, che suddividono lo spazio secondo le tre coordinate x, y, z .

L'insieme di piani si può esprimere come

$$\begin{aligned}
 X_{plane}(i) &= X_0 + (i - 1)s, \quad i = 1, \dots, N_x \\
 Y_{plane}(j) &= Y_0 + (j - 1)s, \quad j = 1, \dots, N_y \\
 Z_{plane}(k) &= Z_0 + (k - 1)s, \quad k = 1, \dots, N_z.
 \end{aligned}
 \tag{1.18}$$

Si chiamano $P_{IN}(X_{IN}, Y_{IN}, Z_{IN})$ e $P_{OUT}(X_{OUT}, Y_{OUT}, Z_{OUT})$ i punti rispettivamente iniziale e finale della traiettoria. L'equazione della retta tridimen-

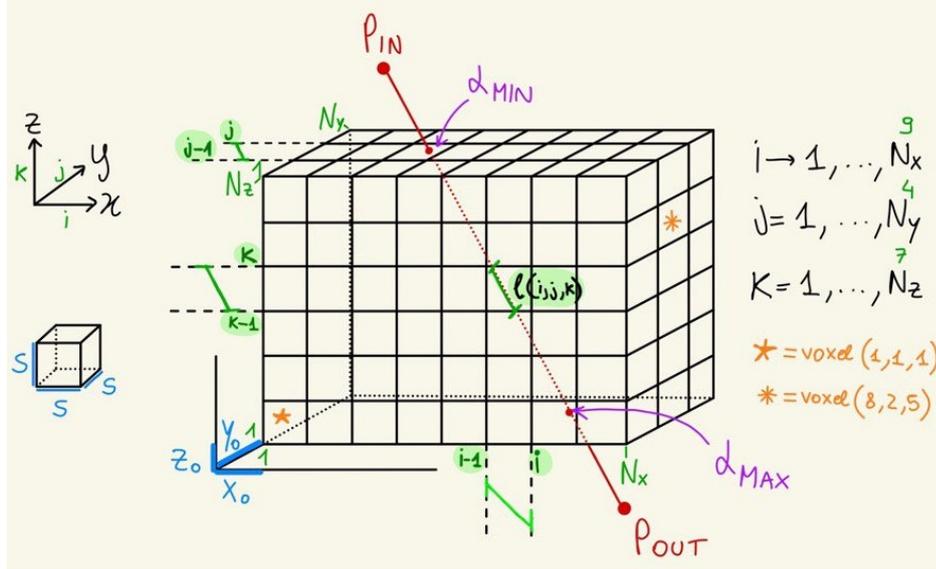


Figura 1.6: Rappresentazione del volume attraversato da un muone e le quantità necessarie allo sviluppo dell'algoritmo.

sionale che definisce il percorso del muone all'interno dello spazio è

$$\begin{cases} p_x(\alpha) = X_{IN} + \alpha(X_{OUT} - X_{IN}) \\ p_y(\alpha) = Y_{IN} + \alpha(Y_{OUT} - Y_{IN}) \\ p_z(\alpha) = Z_{IN} + \alpha(Z_{OUT} - Z_{IN}), \end{cases} \quad (1.19)$$

dove il parametro $\alpha \in [0, 1]$ rappresenta la distanza relativa da P_{IN} di un punto sulla linea definita tra P_{IN} e P_{OUT} : è pari a 0 in P_{IN} e 1 in P_{OUT} , altrimenti assume valori al di fuori dell'intervallo.

Si assegnano il valore $\alpha = \alpha_{min}$ al punto in cui la traiettoria interseca il volume in entrata, $\alpha = \alpha_{max}$ al punto di intersezione in uscita. Dunque, ogni punto della traiettoria all'interno del volume assume valori di α tra $(\alpha_{min}, \alpha_{max})$.

Le equazioni tramite le quali si può calcolare il valore di α in corrispondenza dell'intersezione dell' i -esimo piano, x , y o z , con la traiettoria definita dalla

linea tra P_{IN} e P_{OUT} , si definiscono a partire dalle equazioni (1.19) e (1.18)

$$\begin{aligned}\alpha_x(i) &= \frac{X_{plane}(i) - X_{IN}}{X_{OUT} - X_{IN}} = \alpha_x(i-1) + \frac{s}{X_{OUT} - X_{IN}} = \alpha_x(1) + \frac{(i-1)s}{X_{OUT} - X_{IN}}, \\ \alpha_y(j) &= \frac{Y_{plane}(j) - Y_{IN}}{Y_{OUT} - Y_{IN}} = \alpha_y(i-1) + \frac{s}{Y_{OUT} - Y_{IN}} = \alpha_y(1) + \frac{(i-1)s}{Y_{OUT} - Y_{IN}}, \\ \alpha_z(k) &= \frac{Z_{plane}(k) - Z_{IN}}{Z_{OUT} - Z_{IN}} = \alpha_z(i-1) + \frac{s}{Z_{OUT} - Z_{IN}} = \alpha_z(1) + \frac{(i-1)s}{Z_{OUT} - Z_{IN}},\end{aligned}$$

dove

$$\begin{aligned}\alpha_x(1) &= \frac{[X_{plane}(1) - X_{IN}]}{X_{OUT} - X_{IN}}, \\ \alpha_y(1) &= \frac{[Y_{plane}(1) - Y_{IN}]}{Y_{OUT} - Y_{IN}}, \\ \alpha_z(1) &= \frac{[Z_{plane}(1) - Z_{IN}]}{Z_{OUT} - Z_{IN}}.\end{aligned}$$

Si noti che non è necessario che i valori ottenuti risultino nell'intervallo $[0, 1]$.

Da queste misure si ottengono

$$\begin{aligned}\alpha_{min} &= \max\{0, \alpha_{x,min}, \alpha_{y,min}, \alpha_{z,min}\}, \\ \alpha_{max} &= \min\{1, \alpha_{x,max}, \alpha_{y,max}, \alpha_{z,max}\},\end{aligned}$$

dove

$$\begin{aligned}\alpha_{x,min} &= \min[\alpha_x(1), \alpha_x(N_x)], & \alpha_{x,max} &= \max[\alpha_x(1), \alpha_x(N_x)] \\ \alpha_{y,min} &= \min[\alpha_y(1), \alpha_y(N_y)], & \alpha_{y,max} &= \max[\alpha_y(1), \alpha_y(N_y)] \\ \alpha_{z,min} &= \min[\alpha_z(1), \alpha_z(N_z)], & \alpha_{z,max} &= \max[\alpha_z(1), \alpha_z(N_z)]\end{aligned}$$

Si evidenziano alcune situazioni particolari:

- se $\alpha_{max} \leq \alpha_{min}$, la traiettoria non interseca il volume;
- se $X_{OUT} - X_{IN} = 0$, la traiettoria è perpendicolare all'asse x e il corrispondente valore di α_x non è determinabile; analogamente dicasi per le altre coordinate;
- come conseguenza del punto precedente, se α_x, α_y o α_z non sono definite, sono escluse nel calcolo di α_{min} e α_{max} .

Tra tutti i piani intersecati dalla traiettoria, solo alcuni avranno un valore del parametro α tra $(\alpha_{min}, \alpha_{max})$. Per l'insieme dei piani paralleli a x intersecati dalla traiettoria, il più vicino all'origine degli assi e il più lontano da esso sono indicizzati rispettivamente da $(i_{min}$ e $i_{max})$.

Se $X_{OUT} - X_{IN} > 0$

$$\begin{aligned} i_{min} &= \left[1 + \frac{p_x(\alpha_{min}) - X_0}{s} \right] \\ i_{max} &= \left[1 + \frac{p_x(\alpha_{max}) - X_0}{s} \right], \end{aligned}$$

altrimenti se $X_{OUT} - X_{IN} < 0$

$$\begin{aligned} i_{min} &= \left[1 + \frac{p_x(\alpha_{max}) - X_0}{s} \right] \\ i_{max} &= \left[1 + \frac{p_x(\alpha_{min}) - X_0}{s} \right]. \end{aligned}$$

Analogamente per le altre coordinate si ottengono gli indici (j_{min}, j_{max}) e (k_{min}, k_{max}) .

Si ottengono dunque gli insiemi di valori dei parametri $\{\alpha_x\}$, $\{\alpha_y\}$ e $\{\alpha_z\}$ che contengono ognuno i valori del parametro α nel punto di intersezione della traiettoria con i piani paralleli della rispettiva coordinata, a partire nell'ordine dal punto di entrata del muone nel volume.

Ad esempio per la coordinata x , se $X_{OUT} - X_{IN} > 0$

$$\{\alpha_x\} = \{\alpha_x(i_{min}), \alpha_x(i_{min} + 1), \dots, \alpha_x(i_{max})\},$$

altrimenti se $X_{OUT} - X_{IN} < 0$

$$\{\alpha_x\} = \{\alpha_x(i_{max}), \alpha_x(i_{max} - 1), \dots, \alpha_x(i_{min})\}.$$

Si deriva infine l'insieme unione α_{xyz} , i cui elementi presentati in ordine crescente sono i valori del parametro α in tutti i punti di intersezione della traiettoria con un qualsiasi piano

$$\begin{aligned} \{\alpha_{xyz}\} &= \{\alpha_{min}, \{\alpha_x\}, \{\alpha_y\}, \{\alpha_z\}, \alpha_{max}\} \\ &= \{\alpha_{xyz}[0], \dots, \alpha_{xyz}[N]\}. \end{aligned}$$

I termini doppi all'interno dell'insieme non sono ripetuti: essi si riferiscono a punti di intersezione della traiettoria con un punto di incrocio tra due piani ortogonali.

Due termini adiacenti nell'insieme $\{\alpha_{xyz}\}$ si riferiscono all'intersezione della traiettoria con un particolare voxel. Quindi, date due intersezioni m e $m - 1$, la lunghezza della traiettoria in quel voxel $l(m)$ è data da

$$l(m) = d_{OI}(\alpha_{xyz}[m] - \alpha_{xyz}[m - 1]) , \quad m = 1, \dots, N ,$$

dove d_{OI} è la distanza euclidea tra P_{OUT} e P_{IN}

$$d_{OI} = \sqrt{(X_{OUT} - X_{IN})^2 + (Y_{OUT} - Y_{IN})^2 + (Z_{OUT} - Z_{IN})^2}.$$

Gli indici (i_m, j_m, k_m) identificano il voxel che corrisponde all'intersezione m e $m - 1$ e tale voxel contiene il punto centrale α_{mid} tra le due

$$\begin{aligned} i_m &= \left\lceil 1 + \frac{p_x(\alpha_{mid}) - X_0}{s} \right\rceil \\ j_m &= \left\lceil 1 + \frac{p_y(\alpha_{mid}) - Y_0}{s} \right\rceil \\ k_m &= \left\lceil 1 + \frac{p_z(\alpha_{mid}) - Z_0}{s} \right\rceil, \end{aligned}$$

dove

$$\alpha_{mid} = \frac{\alpha_{xyz}[m] + \alpha_{xyz}[m - 1]}{2}.$$

È necessario prestare attenzione a un aspetto non preso in considerazione in questa metodologia. Nel caso in cui la traiettoria del muone non intersechi nessuno tra i piani paralleli a una coordinata, ad esempio x , si incorre in una incoerenza. Infatti il metodo è costruito in modo tale che risultino valori di α fuori dall'intervallo $[0, 1]$. Di conseguenza al piano intersecato più vicino all'origine degli assi verrebbe assegnato un indice più grande di quello del piano intersecato più lontano dall'origine. In questo contesto, non essendoci alcun indice del piano da registrare, è sufficiente memorizzare l'indice x del voxel.

Non è complesso riuscire a rilevare questa eventualità, ma è importante che l'algoritmo derivante dal metodo ne tenga conto per correggere opportunamente l'errore.

Per passare dalle quantità risultanti da questa metodologia a quelle utilizzate nei metodi descritti nei capitoli precedenti, occorre innanzitutto numerare i voxel da 1 a M , anzichè identificarli tramite l'incrocio dei 3 indici (i_m, j_m, k_m) . Le misure $l(m)$ corrispondono inoltre proprio alle misure L_j , per $j = 1, \dots, M$, introdotte alla fine del Paragrafo 1.2.3. Nel caso in esame, infine, si nota che $X_0 = Y_0 = Z_0 = 0$.

Capitolo 2

L'attuale modello statistico di riferimento

In questo capitolo si riprendono i contenuti della paragrafo 1.3 e si espongono la specificazione del modello e l'approccio di stima proposti da Schultz et al. (2007): a partire dalle informazioni sullo *scattering* dei muoni, vengono stimati i valori delle densità di *scattering* dei vari materiali e dunque si può ricostruire un'immagine della composizione del volume.

2.1 Specificazione del modello

2.1.1 Dati e modello

Si consideri un insieme di n muoni che attraversa un volume da ispezionare. Essendo ignoto il contenuto del volume, esso viene suddiviso in M voxel cubici di lato s e si assume che ognuno di essi sia composto di materiale omogeneo. In questo contesto è possibile misurare la deviazione angolare e il dislocamento totali di ogni muone, rilevando per ciascuno la posizione e l'angolo di entrata e uscita dal volume lungo le coordinate x e y . Dunque, per ogni muone i si definiscono i vettori bidimensionali di dati

$$d_{x,i} = \begin{bmatrix} a_{x,i} \\ u_{x,i} \end{bmatrix}, \quad d_{y,i} = \begin{bmatrix} a_{y,i} \\ u_{y,i} \end{bmatrix}, \quad i = 1, \dots, n,$$

dove $a_{x,i}$ e $a_{y,i}$ misurano lo *scattering* angolare del muone i lungo l'asse specificato, mentre $u_{x,i}$ e $u_{y,i}$ quantificano il dislocamento. Considerando l'intero insieme di n muoni, si rilevano in totale i campioni di osservazioni indipendenti e identicamente distribuite $d_x = (d_{x,1}, \dots, d_{x,n})$ e $d_y = (d_{y,1}, \dots, d_{y,n})$.

Dalle formulazioni espone nel paragrafo 1.3, si evince che la distribuzione congiunta $D_{x,i} = (A_{x,i}, U_{x,i})$ è una Gaussiana di media 0 e varianza Σ_i

$$p_{A_{x,i}, U_{x,i}}(d_{x,i}) = \frac{1}{2\pi|\Sigma_i|^{1/2}} \exp\left(-\frac{1}{2}d_{x,i}^T \Sigma_i^{-1} d_{x,i}\right). \quad (2.1)$$

Il valore Σ_i si riprende dalle equazioni (1.16) e (1.17) incorporando l'informazione dello specifico muone i

$$\Sigma_i = p_{r,i}^2 \sum_{j=1}^M \lambda_j W_{ij}, \quad (2.2)$$

con $p_{r,i}^2$ una quantità che dipende dal momento del muone, λ_j densità di *scattering* del materiale di cui è composto il voxel j e W_{ij} matrice dei pesi

$$W_{ij} = \begin{bmatrix} L_{ij} & L_{ij}^2/2 + L_{ij}T_{ij} \\ L_{ij}^2/2 + L_{ij}T_{ij} & L_{ij}^3/3 + L_{ij}^2T_{ij} + L_{ij}T_{ij}^2 \end{bmatrix}.$$

Allo stesso modo anche la distribuzione congiunta di $D_{y,i} = (A_{y,i}, U_{y,i})$ presenta la stessa forma di (2.1) ma relativamente alle misure dello *scattering* lungo la coordinata y . Si nota che Σ_i calcolata per lo *scattering* lungo y risulta identica a Σ_i ottenuta per la distribuzione secondo x .

La formula (2.2) per ricavare la varianza Σ_i evidenzia che lo *scattering* totale di ogni muone risulta dallo *scattering* verificatosi, o meno, all'interno di ogni voxel j , con $j = 1, \dots, M$.

La distribuzione dei dati osservati $d = (d_x, d_y)$ dunque si esprime come

$$\begin{aligned} p(d_x, d_y) &= p(d_x) \cdot p(d_y) = \prod_{i=1}^n p(d_{x,i}) \cdot p(d_{y,i}) \\ &= (2\pi)^{-2n} \prod_{i=1}^n |\Sigma_i|^{-1} \exp\left\{\sum_{i=1}^n \left[-\frac{1}{2}\left(d_{x,i}^T \Sigma_i^{-1} d_{x,i} + d_{y,i}^T \Sigma_i^{-1} d_{y,i}\right)\right]\right\}. \end{aligned}$$

2.1.2 Stima di massima verosimiglianza

È quindi di interesse stimare la densità di *scattering* del materiale λ_j che caratterizza il j -esimo voxel, per ogni $j = 1, \dots, M$. Il parametro M -dimensionale $\lambda = (\lambda_1, \dots, \lambda_M)$ è relativo alla distribuzione congiunta (D_x, D_y) e come metodologia per la sua stima si utilizza la massima verosimiglianza $L(\lambda; (d_x, d_y))$. Si è enunciato in precedenza che D_x e D_y sono indipendenti e identicamente distribuite, dunque la funzione di verosimiglianza associata alla distribuzione $p(d_x, d_y)$ è

$$\begin{aligned} L(\lambda; (d_x, d_y)) &= L(\lambda; d_x) \cdot L(\lambda; d_y) = \prod_{i=1}^n L(\lambda; d_{x,i}) \cdot L(\lambda; d_{y,i}) \\ &\propto \prod_{i=1}^n |\Sigma_i|^{-1} \exp \left\{ \sum_{i=1}^n \left[-\frac{1}{2} \left(d_{x,i}^T \Sigma_i^{-1} d_{x,i} + d_{y,i}^T \Sigma_i^{-1} d_{y,i} \right) \right] \right\}. \end{aligned} \quad (2.3)$$

I dati osservati (d_x, d_y) sono il risultato dello *scattering* del muone i all'interno di ognuno dei voxel attraversati; questa struttura si nota sempre nel calcolo della matrice Σ_i in (2.2), dove appare esplicitamente che la sua costruzione deriva dall'aggregazione dei singoli j -esimi eventi di *scattering*.

È però di interesse studiare lo *scattering* particolare all'interno di ogni voxel, ciascuno caratterizzato da una specifica densità di *scattering* λ_j . Poichè non è possibile osservare quello che si verifica all'interno del volume, le misurazioni di deviazione angolare e dislocamento all'interno di ogni voxel sono dati mancanti. Si definiscono dunque due vettori di dati mancanti

$$\begin{aligned} h_x &= \left\{ h_{x,ij} = \begin{bmatrix} a_{x,ij} \\ u_{x,ij} \end{bmatrix} : 1 \leq i \leq n \text{ e } 1 \leq j \leq M \right\} \\ h_y &= \left\{ h_{y,ij} = \begin{bmatrix} a_{y,ij} \\ u_{y,ij} \end{bmatrix} : 1 \leq i \leq n \text{ e } 1 \leq j \leq M \right\}, \end{aligned}$$

dove $a_{x,ij}$, $a_{y,ij}$, $u_{x,ij}$ e $u_{y,ij}$ sono rispettivamente le misure dell'angolo di *scattering* e del dislocamento dell' i -esimo muone nel j -esimo voxel secondo le due coordinate.

Ricordando che lo *scattering* lungo x e lungo y sono indipendenti, la distribuzione congiunta dei dati (D_x, D_y, H_x, H_y) può essere espressa nel seguente

modo

$$\begin{aligned}
p(d_x, d_y, h_x, h_y) &= p(d_x, h_x) \cdot p(d_y, h_y) = p(d_x|h_x)p(h_x) \cdot p(d_x|h_y)p(h_y) \\
&= \prod_{i=1}^n p(d_{x,i}|h_{x,i1}, \dots, h_{x,iM})p(h_{x,i1}) \dots p(h_{x,iM}) \cdot \\
&\quad \cdot p(d_{y,i}|h_{y,i1}, \dots, h_{y,iM})p(h_{y,i1}) \dots p(h_{y,iM}),
\end{aligned} \tag{2.4}$$

dove l'ultima uguaglianza segue dall'assunzione che lo *scattering* in ogni voxel è indipendente rispetto a quello negli altri voxel, come enunciato al termine del paragrafo 1.3.

Si consideri la distribuzione condizionata $p(d_{x,i}|h_{x,i1}, \dots, h_{x,iM})$: questa è degenera perchè i dati mancanti $h_{x,ij}$, con $j = 1, \dots, M$, determinano univocamente i dati osservati $d_{x,i}$, in quanto l'unione degli M eventi di *scattering* $h_{x,ij}$ costituiscono l'intera deviazione $d_{x,i}$. Lo stesso vale per la stessa quantità calcolata per lo *scattering* lungo y . Di conseguenza la distribuzione congiunta in (2.4) si può esprimere come

$$\begin{aligned}
p(d_x, d_y, h_x, h_y) &\propto p(h_x, h_y) = \prod_{i=1}^n p(h_{x,i1}) \dots p(h_{x,iM}) \cdot p(h_{y,i1}) \dots p(h_{y,iM}) \\
&\propto \prod_{i=1}^n \prod_{j=1}^M p(h_{x,ij}) \cdot p(h_{y,ij}).
\end{aligned} \tag{2.5}$$

In base a quanto esposto nel paragrafo 1.3, le distribuzioni $p(h_{x,ij})$ e $p(h_{y,ij})$ riferite allo *scattering* in un voxel sono note e sono delle Gaussiane bivariate di media 0 e matrice di covarianza Σ_{ij} pari a

$$\Sigma_{ij} = \lambda_j B_{ij} p_{r,i}^2 \tag{2.6}$$

con

$$B_{ij} = \begin{bmatrix} L_{ij} & L_{ij}^2/2 \\ L_{ij}^2/2 & L_{ij}^3/3 \end{bmatrix}. \tag{2.7}$$

Entrambe le equazioni sono state ottenute generalizzando (1.14) e (1.15).

La funzione di log-verosimiglianza per il parametro vettoriale di interesse $\lambda = (\lambda_1, \dots, \lambda_M)$ associata alla distribuzione congiunta $p(d_x, d_y, h_x, h_y)$ specificata in (2.5) è

$$\ell(\lambda|d_x, d_y, h_x, h_y) \propto \ell(\lambda|h_x, h_y) = \ell(\lambda|h_x) + \ell(\lambda|h_y). \tag{2.8}$$

Per la stima di massima verosimiglianza risulta particolarmente adatta la scelta dell'*Expectation-Maximization Algorithm* (EM) come strumento operativo, in quanto è stato elaborato opportunamente per la stima di massima verosimiglianza in presenza di dati mancanti.

L'algoritmo EM, formulato da Dempster, Laird e Rubin (1977), prevede la definizione della funzione di verosimiglianza completa, relativa alla distribuzione congiunta dei dati osservati e mancanti. Questa è proprio quella riportata in (2.8) e si indica come $\ell^C(\lambda|d_x, d_y, h_x, h_y)$.

2.2 Algoritmo EM

2.2.1 Definizione

L'idea alla base dell'algoritmo EM consiste nel trattare i dati osservati come una versione incompleta del campione ideale di dati, il quale consente un'analisi più semplice. Questo approccio massimizza non la log-verosimiglianza completa in (2.8), ma il suo valore atteso condizionato ai dati osservati, noto come funzione ausiliaria Q . Nel dettaglio, l'Algoritmo EM prevede l'alternanza iterativa di due passi, l'*Expectation* (E)-step ed il *Maximization* (M)-step. Sia $\lambda^{(q)}$ la stima del parametro di interesse all'iterazione (q): all'E-step dell'iterazione successiva ($q + 1$) viene calcolata la funzione ausiliaria

$$Q(\lambda, \lambda^{(q)}; d_x, d_y) = \mathbb{E}_{H|d_x, d_y, \lambda^{(q)}}[\ell^C(\lambda|d_x, d_y, H_x, H_y)]$$

e all'M-step quest'ultima viene massimizzata rispetto a λ per aggiornare la sua stima in $\lambda^{(q+1)}$

$$\lambda^{(q+1)} = \arg \max_{\lambda} Q(\lambda, \lambda^{(q)}; d_x, d_y).$$

I due passi vengono iterati fino a convergenza.

2.2.2 Passi E e M

In particolare:

1. Nel passo E si determina

$$\begin{aligned}
Q(\lambda, \lambda^{(q)}; d_x, d_y) &= \mathbb{E}_{H_x, H_y | d_x, d_y, \lambda^{(q)}} [\ell^C(\lambda | d_x, d_y, H_x, H_y)] \\
&= \mathbb{E}_{H_x, H_y | d_x, d_y, \lambda^{(q)}} [\ell^C(\lambda | d_x, H_x) + \ell^C(\lambda | d_y, H_y)] \\
&= \mathbb{E}_{H_x | d_x, \lambda^{(q)}} [\ell^C(\lambda | d_x, H_x)] + \mathbb{E}_{H_y | d_y, \lambda^{(q)}} [\ell^C(\lambda | d_y, H_y)].
\end{aligned} \tag{2.9}$$

I due valori attesi ottenuti nell'ultimo passaggio possono essere svolti in maniera separata. Si decide dunque di sviluppare i calcoli e i passaggi per un singolo valore atteso, omettendo i pedici x e y , a indicare che tale procedimento va svolto in maniera analoga per lo *scattering* lungo l'asse x e poi lo *scattering* lungo l'asse y .

Si calcola quindi in modo dettagliato il valore atteso

$$\begin{aligned}
\mathbb{E}_{H | d, \lambda^{(q)}} [\ell^C(\lambda | d, H)] &= \\
&= \mathbb{E}_{H | d, \lambda^{(q)}} \left\{ -\frac{1}{2} \left[\sum_{i=1}^n \sum_{j=1}^M \left(\log(|\Sigma_{ij}|) + H_{ij}^T \Sigma_{ij}^{-1} H_{ij} \right) \right] \right\} \\
&= \mathbb{E}_{H | d, \lambda^{(q)}} \left\{ -\frac{1}{2} \left[\sum_{j=1}^M \sum_{i: L_{ij} \neq 0} \left(\log \left(\lambda_j^2 p_{r,i}^4 \frac{L_{ij}^4}{12} \right) + H_{ij}^T \frac{B_{ij}^{-1}}{\lambda_j p_{r,i}^2} H_{ij} \right) \right] \right\} \\
&\propto \sum_{j=1}^M \left\{ -n_j \log(\lambda_j) - \frac{1}{2\lambda_j} \left[\sum_{i: L_{ij} \neq 0} \left(\frac{1}{p_{r,i}^2} \mathbb{E} \left(H_{ij}^T B_{ij}^{-1} H_{ij} | d_i, \lambda^{(q)} \right) \right) \right] \right\},
\end{aligned} \tag{2.10}$$

dove n_j indica il numero dei muoni che colpiscono il j -esimo voxel, ovvero per cui $L_{ij} \neq 0$. Il risultato in (2.10) si ottiene esplicitando il calcolo del determinante di Σ_{ij} e il valore di Σ_{ij}^{-1} , che si ottengono facendo riferimento alle equazioni (2.6) e (2.7)

$$|\Sigma_{ij}| = \lambda_j^2 p_{r,i}^4 \left(\frac{L_{ij}^4}{3} - \frac{L_{ij}^4}{4} \right) = \lambda_j^2 p_{r,i}^4 \frac{L_{ij}^4}{12}, \tag{2.11}$$

$$\Sigma_{ij}^{-1} = (\lambda_j p_{r,i}^2 B_{ij})^{-1} = \frac{B_{ij}^{-1}}{\lambda_j p_{r,i}^2}. \tag{2.12}$$

Per determinare il valore atteso $\mathbb{E} \left(H_{ij}^T B_{ij}^{-1} H_{ij} | d_i, \lambda^{(q)} \right)$ si consideri la variabile casuale $H_{ij} | d_i$: poichè sia H_{ij} che D_i sono Gaussiane, anche la

condizionata $H_{ij}|d_i$ è Gaussiana; in particolare

$$\begin{aligned} H_{ij}|d_i &\sim N_2(\mathbb{E}(H_{ij}) + \Sigma_{D_i H_{ij}}^T \Sigma_{D_i}^{-1}(d_i - \mathbb{E}(D_i)), \Sigma_{H_{ij}} - \Sigma_{D_i H_{ij}}^T \Sigma_{D_i}^{-1} \Sigma_{D_i H_{ij}}) \\ &\sim N_2(\Sigma_{D_i H_{ij}}^T \Sigma_i^{-1} d_i, \Sigma_{ij} - \Sigma_{D_i H_{ij}}^T \Sigma_i^{-1} \Sigma_{D_i H_{ij}}), \end{aligned} \quad (2.13)$$

dove $\Sigma_{D_i H_{ij}}$ è la matrice di covarianza della distribuzione congiunta (D_i, H_{ij}) , anch'essa Gaussiana (Johnson e Wichern (2007)).

Per il calcolo del valore atteso si tiene inoltre conto del fatto che $H_{ij}^T B_{ij}^{-1} H_{ij}|d_i$ è una forma quadratica; in particolare siano $C = B_{ij}^{-1}$ e $X = H_{ij}|d_i$, allora il valore atteso della forma quadratica $X^T C X$ è

$$\mathbb{E}_{X^T C X} = \text{tr}(C \Sigma_X) + \mu_X^T C \mu_X, \quad (2.14)$$

dove μ_X e Σ_X sono rispettivamente la media e la varianza di X , esplicitate in (2.13).

Il valore atteso in esame si può quindi esprimere come

$$\begin{aligned} \mathbb{E}\left(H_{ij}^T B_{ij}^{-1} H_{ij} | d_i, \lambda^{(q)}\right) &= \\ &= \text{tr}\left[B_{ij}^{-1}(\Sigma_{ij} - \Sigma_{D_i H_{ij}}^T \Sigma_i^{-1} \Sigma_{D_i H_{ij}})\right] + (\Sigma_{D_i H_{ij}}^T \Sigma_i^{-1} d_i)^T B_{ij}^{-1} (\Sigma_{D_i H_{ij}}^T \Sigma_i^{-1} d_i) \\ &= \text{tr}(B_{ij}^{-1} \Sigma_{ij} - B_{ij}^{-1} \Sigma_{D_i H_{ij}}^T \Sigma_i^{-1} \Sigma_{D_i H_{ij}}) + d_i^T \Sigma_i^{-1} (\Sigma_{D_i H_{ij}} B_{ij}^{-1} \Sigma_{D_i H_{ij}}^T) \Sigma_i^{-1} d_i. \end{aligned} \quad (2.15)$$

Anzichè svolgere esplicitamente i prodotti matriciali si può ricorrere alla seguente uguaglianza

$$\Sigma_{D_i H_{ij}} B_{ij}^{-1} \Sigma_{D_i H_{ij}}^T = W_{ij} (p_{r,i}^2 \lambda_j)^2,$$

la cui derivazione passa per il calcolo della matrice di covarianza $\Sigma_{D_i H_{ij}}$, riportato in Appendice A.2. Dunque, in riferimento a questi ultimi risultati

$$\begin{aligned} \Sigma_{D_i H_{ij}} (\Sigma_{ij}^{-1}) \Sigma_{D_i H_{ij}}^T &= \lambda_j p_{r,i}^2 W_{ij} \\ \rightarrow \Sigma_{D_i H_{ij}} \left(\frac{1}{\lambda_j p_{r,i}^2} B_{ij}^{-1} \right) \Sigma_{D_i H_{ij}}^T &= \lambda_j p_{r,i}^2 W_{ij} \\ \rightarrow \Sigma_{D_i H_{ij}} B_{ij}^{-1} \Sigma_{D_i H_{ij}}^T &= W_{ij} (\lambda_j p_{r,i}^2)^2. \end{aligned} \quad (2.16)$$

In Appendice A.2 sono inoltre esplicitati i pochi prodotti matriciali essenziali per esprimere la precedente uguaglianza.

Riconoscendo che B_{ij} e Σ_i sono matrici simmetriche e che $\text{tr}(AB) = \text{tr}(BA)$, il calcolo del valore atteso espresso in (2.15) può così procedere

$$\begin{aligned}
\mathbb{E}\left(H_{ij}^T B_{ij}^{-1} H_{ij} \mid d_i, \lambda^{(q)}\right) &= \\
&= \text{tr}\left[B_{ij}^{-1} \lambda_j^{(q)} p_{r,i}^2 B_{ij} - (\Sigma_{D_i H_{ij}} B_{ij}^{-1} \Sigma_{D_i H_{ij}}^T) \Sigma_i^{-1}\right] + \\
&\quad + d_i^T \Sigma_i^{-1} (W_{ij} (\lambda_j^{(q)} p_{r,i}^2)^2) \Sigma_i^{-1} d_i \\
&= \text{tr}\left[\lambda_j^{(q)} p_{r,i}^2 - W_{ij} (\lambda_j^{(q)} p_{r,i}^2)^2 \Sigma_i^{-1}\right] + d_i^T \Sigma_i^{-1} W_{ij} (\lambda_j^{(q)} p_{r,i}^2)^2 \Sigma_i^{-1} d_i \\
&= 2\lambda_j^{(q)} p_{r,i}^2 - (\lambda_j^{(q)} p_{r,i}^2)^2 \text{tr}(W_{ij} \Sigma_i^{-1}) + (\lambda_j^{(q)} p_{r,i}^2)^2 (d_i^T \Sigma_i^{-1} W_{ij} \Sigma_i^{-1} d_i) \\
&= p_{r,i}^2 \left\{ 2\lambda_j^{(q)} + (\lambda_j^{(q)})^2 p_{r,i}^2 [d_i^T \Sigma_i^{-1} W_{ij} \Sigma_i^{-1} d_i - \text{tr}(\Sigma_i^{-1} W_{ij})] \right\}.
\end{aligned}$$

Per i prossimi sviluppi si decide di introdurre la quantità $G_{ij}(\lambda^{(q)})$ per semplificare la lettura

$$\begin{aligned}
G_{ij}(\lambda^{(q)}) &= \frac{1}{p_{r,i}^2} \mathbb{E}\left(H_{ij}^T B_{ij}^{-1} H_{ij} \mid d_i, \lambda^{(q)}\right) \\
&= 2\lambda_j^{(q)} + (\lambda_j^{(q)})^2 p_{r,i}^2 [d_i^T \Sigma_i^{-1} W_{ij} \Sigma_i^{-1} d_i - \text{tr}(\Sigma_i^{-1} W_{ij})].
\end{aligned}$$

Il valore atteso in (2.10) può essere espresso infine come

$$\begin{aligned}
\mathbb{E}_{H \mid d, \lambda^{(q)}}[\ell^C(\lambda \mid d, H)] &\propto \\
&\propto \sum_{j=1}^M \left\{ -n_j \log(\lambda_j) - \frac{1}{2\lambda_j} \left[\sum_{i: L_{ij} \neq 0} \left(\frac{1}{p_{r,i}^2} \mathbb{E}\left(H_{x,ij}^T B_{ij}^{-1} H_{x,ij} \mid d_{x,i}, \lambda^{(q)}\right) \right) \right] \right\} \\
&\propto \sum_{j=1}^M \left(-n_j \log(\lambda_j) - \frac{1}{2\lambda_j} \sum_{i: L_{ij} \neq 0} G_{ij}(\lambda^{(q)}) \right).
\end{aligned}$$

Si ritorna infine alla formulazione della funzione ausiliaria completa

$Q(\lambda, \lambda^{(q)}; d_x, d_y)$ in (2.9): questa risulta essere pari a

$$\begin{aligned}
Q(\lambda, \lambda^{(q)}; d_x, d_y) &= \mathbb{E}_{H_x|d_x, \lambda^{(q)}}[\ell^C(\lambda|d_x, H_x)] + \mathbb{E}_{H_y|d_y, \lambda^{(q)}}[\ell^C(\lambda|d_y, H_y)] \propto \\
&\propto \sum_{j=1}^M \left[-n_j \log(\lambda_j) - \frac{1}{2\lambda_j} \sum_{i:L_{ij} \neq 0} G_{x,ij}(\lambda^{(q)}) + \right. \\
&\qquad \qquad \qquad \left. - n_j \log(\lambda_j) - \frac{1}{2\lambda_j} \sum_{i:L_{ij} \neq 0} G_{y,ij}(\lambda^{(q)}) \right] \\
&\propto \sum_{j=1}^M Q_j(\lambda_j, \lambda^{(q)}; d_x, d_y).
\end{aligned} \tag{2.17}$$

2. Il passo M prevede la massimizzazione rispetto a λ della funzione ausiliaria $Q(\lambda; \lambda^{(q)}; d_x, d_y)$. In particolare, all'iterazione $(q+1)$ dell'algoritmo EM si calcola la stima $\lambda^{(q+1)}$ tramite la derivazione della funzione ausiliaria rispetto a ogni λ_j , per $j = 1, \dots, M$.

Si sottolinea che nello sviluppo descritto in (2.17), per un indice specifico j fissato dalla sommatoria, la funzione $Q_j(\lambda_j, \lambda^{(q)}; d_x, d_y)$ include termini che coinvolgono esclusivamente il parametro λ_j corrispondente a quel particolare indice j . Per questo motivo, derivare $Q(\lambda, \lambda^{(q)}; d_x, d_y)$ rispetto a λ_j equivale a derivare direttamente $Q_j(\lambda_j, \lambda^{(q)}; d_x, d_y)$.

Dunque, per ogni $j = 1, \dots, M$ si effettua il calcolo della derivata di $Q_j(\lambda_j, \lambda^{(q)}; d_x, d_y)$ rispetto a λ_j , si eguaglia il risultato a 0 e si trova la seguente formula iterativa per il valore di λ_j all'iterazione $q+1$ dell'algoritmo

$$\lambda_j^{(q+1)} = \frac{1}{2n_j} \sum_{i:L_{ij} \neq 0} \frac{G_{x,ij}(\lambda^{(q)}) + G_{y,ij}(\lambda^{(q)})}{2}.$$

Capitolo 3

Il modello statistico proposto

In questo capitolo si propone una modifica alla distribuzione degli angoli di *scattering* utilizzata da Schultz et al. (2007) nella sua metodologia. In particolare si specifica un modello mistura a due componenti Gaussiane di nota proporzione e si sviluppa il procedimento di stima per il nuovo modello. Per la teoria sulle misture si fa riferimento a Bouveyron et al. (2019, §2.1–2.3) e McLachlan, Lee e Rathnayake (2019).

3.1 Il modello mistura

3.1.1 Formalizzazione statistica della proposta

Nel capitolo 2 è stato specificato il modello per descrivere lo *scattering* dei muoni, teorizzato in Rossi (1952), e sviluppato il procedimento di stima del parametro da cui dipende, la densità di *scattering* dei materiali $(\lambda_1, \dots, \lambda_M)$, dunque utilizzando una Gaussiana di media 0 e varianza σ_a^2 in (1.9).

In Lynch e Dahl (1990) si afferma che la vera variabile generatrice degli angoli di *scattering* presenta tuttavia code più pesanti di una Gaussiana. In particolare, è il 2% degli angoli più grandi in valore assoluto a non essere ben approssimato dalla distribuzione espressa in (1.7).

Per attribuire maggiore massa di probabilità alle code, la distribuzione degli angoli di *scattering* viene nel seguito descritta da una mistura di due Gaussiane, in cui la prima componente coincide con la distribuzione (1.7) e la seconda ha pari media ma varianza maggiore.

Nello specifico, tale mistura si esprime nel seguente modo

$$p_A(a) = \tau p_1(a) + (1 - \tau)p_2(a), \quad (3.1)$$

dove

$$p_1(a) = \frac{1}{\sqrt{2\pi\sigma_{a_1}^2}} \exp\left\{-\frac{a^2}{2\sigma_{a_1}^2}\right\}$$

$$p_2(a) = \frac{1}{\sqrt{2\pi\sigma_{a_2}^2}} \exp\left\{-\frac{a^2}{2\sigma_{a_2}^2}\right\},$$

e $\tau \in [0, 1]$ è la probabilità che un'osservazione sia generata dalla prima componente. È ragionevole considerare τ come noto e nello specifico pari a 0.98 a indicare che la prima componente della mistura modella il 98% degli angoli di *scattering*, mentre la seconda genera il rimanente 2% responsabile delle code più pesanti nella distribuzione Gaussiana specificata in (1.7).

La varianza della prima componente $\sigma_{a_1}^2$ coincide dunque con σ_a^2 indicata in (1.9). Poichè non esiste letteratura a noi nota in ambito fisico intorno a questo nuovo modello proposto, non è possibile determinare la varianza $\sigma_{a_2}^2$ della seconda componente in una funzione precisa di termini legati alle caratteristiche fisiche del muone e del materiale. Una possibile proposta è esprimere $\sigma_{a_2}^2$ in funzione di $\sigma_{a_1}^2$. In questo lavoro si assume che $\sigma_{a_2}^2$ sia proporzionale a $\sigma_{a_1}^2$, dunque

$$\sigma_{a_2}^2 = K\sigma_{a_1}^2 = K\sigma_a^2, \quad K > 1.$$

A questo punto si prosegue derivando la distribuzione del dislocamento associato allo *scattering* angolare, in base a quale componente della mistura segua quest'ultimo. In entrambi i casi il dislocamento è anch'esso modellato da una Gaussiana di media 0. Riprendendo i legami tra a e u esposti nel Paragrafo 1.3, nel caso in cui l'angolo di *scattering* segua la prima componente della mistura, la varianza del dislocamento è pari a (1.10)

$$\sigma_{u_1}^2 = \frac{L^2}{3}\sigma_{a_1}^2 = \frac{L^2}{3}\sigma_a^2 = \frac{L^2}{3}\lambda p_r^2,$$

altrimenti

$$\sigma_{u_2}^2 = \frac{L^2}{3}\sigma_{a_2}^2 = K\frac{L^2}{3}\sigma_a^2 = K\frac{L^3}{3}\lambda p_r^2,$$

dove L è la lunghezza della traiettoria tridimensionale del muone all'interno del voxel, λ è la densità di *scattering* del materiale e p_r^2 è una misura relativa

al momento del muone. Anche in questo caso si nota che la varianza $\sigma_{u_2}^2$ è proporzionale a $\sigma_{u_1}^2$ con un costante di proporzionalità K . Inoltre, poichè si lavora con la distribuzione congiunta dello *scattering* (A, U) , bisogna calcolare la covarianza tra le due quantità. Per la prima componente della mistura questa risulta coincidente con (1.11)

$$\sigma_{au_1} = \frac{L}{2}\sigma_{a_1}^2 = \frac{L}{2}\sigma_a^2 = \frac{L^2}{2}\lambda p_r^2,$$

mentre per la seconda componente

$$\sigma_{au_2} = \frac{L}{2}\sigma_{a_2}^2 = K\frac{L}{2}\sigma_a^2 = K\frac{L^2}{2}\lambda p_r^2.$$

Dunque anche la covarianza σ_{au_2} è pari a σ_{au_1} moltiplicata per la costante K .

Si generalizzano infine questi risultati per un generico muone i che attraversa un voxel j di materiale omogeneo.

La distribuzione congiunta (A_{ij}, U_{ij}) è una Gaussiana bidimensionale di media 0 e matrice di covarianza

$$\Sigma_{ij1} = \lambda_j B_{ij} p_{r,i}^2 = \Sigma_{ij} \quad (3.2)$$

se lo *scattering* si distribuisce secondo la prima componente della mistura, altrimenti

$$\Sigma_{ij2} = K\lambda_j B_{ij} p_{r,i}^2 = K\Sigma_{ij1} = K\Sigma_{ij}, \quad (3.3)$$

dove Σ_{ij} è espressa in (2.6).

Come sottolineato in precedenza questi passaggi sono analoghi per lo *scattering* rispetto alla coordinata y . È però rilevante evidenziare come non sia strettamente necessario che lo *scattering* di un muone in un voxel lungo l'asse y segua la distribuzione specifica della stessa componente della mistura che modella il corrispondente *scattering* lungo l'asse x .

L'obiettivo è sempre trovare una stima per il parametro $\lambda = (\lambda_1, \dots, \lambda_M)$, coinvolto nella specificazione delle matrici di covarianza $\Sigma_{x,ij1}$, $\Sigma_{x,ij2}$, $\Sigma_{y,ij1}$ e $\Sigma_{y,ij2}$, con $j = 1, \dots, M$. Anche in questo caso si scelgono la massima verosimiglianza come metodo di stima per $\lambda = (\lambda_1, \dots, \lambda_M)$, il parametro che misura la densità di *scattering* del materiale che compone il voxel, e l'algoritmo EM come strumento operativo.

La specificazione del modello relativo alla nuova proposta per la distribuzione dello *scattering*, si ottiene in maniera corrispondente a quanto esposto nel paragrafo 2.1.

3.1.2 Specificazione del modello

Si consideri un volume da ispezionare suddiviso in M voxel cubici di lato s e si considerino le misure dell'angolo di incidenza e di posizione di n muoni che attraversano tale volume. Si definiscono di conseguenza due campioni di dati osservati $d_x = (d_{x,1}, \dots, d_{x,n})$ e $d_y = (d_{y,1}, \dots, d_{y,n})$, che misurano lo *scattering* lungo le due coordinate x e y . Ogni elemento è un vettore bidimensionale, in particolare

$$d_x = \left\{ d_{x,i} = \begin{bmatrix} a_{x,i} \\ u_{x,i} \end{bmatrix} : 1 \leq i \leq n \right\}, \quad d_y = \left\{ d_{y,i} = \begin{bmatrix} a_{y,i} \\ u_{y,i} \end{bmatrix} : 1 \leq i \leq n \right\},$$

dove $a_{.,i}$ e $u_{.,i}$ quantificano lo *scattering* angolare e il dislocamento del muone i lungo l'asse specifico.

Nella specificazione del modello nel capitolo 2 era risultato conveniente esprimere i dati osservati d_x e d_y come unione dei dati mancanti h_x e h_y , dunque come risultato della somma dello *scattering* di uno specifico muone tra i voxel che compongono il volume complessivo. (Si riprendano le equazioni (1.4) e (1.5)).

Si consideri per il momento lo *scattering* lungo l'asse x (di cui si omette la specificazione al pedice); i dati mancanti h

$$h = \left\{ h_{ij} = \begin{bmatrix} a_{ij} \\ u_{ij} \end{bmatrix} : 1 \leq i \leq n \text{ e } 1 \leq j \leq M \right\}$$

misurano lo *scattering* angolare a_{ij} e il dislocamento u_{ij} del muone i all'interno di ogni voxel $j = 1, \dots, M$, ognuno costituito di materiale omogeneo. La distribuzione dello *scattering* H_{ij} è indipendente tra gli n muoni e tra gli M voxel.

La distribuzione generatrice dei vettori che costituiscono il campione di dati nascosti h è dunque nota ed è proprio la mistura delle due Gaussiane

esplicitata in precedenza. Facendo riferimento a (3.2) e (3.3), si ottiene più precisamente la seguente formulazione

$$p(h_{ij}) = \tau p_1(h_{ij}) + (1 - \tau) p_2(h_{ij}),$$

con $\tau = 0.98$ e

$$p_1(h_{ij}) = \frac{1}{2\pi|\Sigma_{ij}|^{1/2}} \exp\left(-\frac{1}{2}h_{ij}^T \Sigma_{ij}^{-1} h_{ij}\right)$$

$$p_2(h_{ij}) = \frac{1}{2\pi|K\Sigma_{ij}|^{1/2}} \exp\left(-\frac{1}{2}h_{ij}^T (K\Sigma_{ij})^{-1} h_{ij}\right).$$

A questo punto si può determinare la distribuzione dei dati osservati D ; questa tuttavia risulta di complicata espressione perchè lo *scattering* in ognuno dei voxel è modellato da una diversa mistura a due componenti.

Si introduce dunque la variabile Z_{ij} . Essa assume valore 1 quando h_{ij} segue la distribuzione della prima componente della mistura, 0 altrimenti. La variabile Z_{ij} è dunque una Bernoulliana di parametro τ ; riassumendo

$$Z_{ij} \sim Be(\tau), \quad z_{ij} = \begin{cases} 1, & h_{ij} \in 1^\circ \text{ componente della mistura} \\ 0, & \text{altrimenti} \end{cases}.$$

Si assume che le z_{ij} , per $i = 1, \dots, n$ e $j = 1, \dots, M$, siano indipendenti e identicamente distribuite. Ad esempio, la densità congiunta (H_{ij}, Z_{ij}) si può esprimere come

$$p(h_{ij}, z_{ij}) = [\tau p_1(h_{ij})]^{z_{ij}} + [(1 - \tau) p_2(h_{ij})]^{1-z_{ij}},$$

mentre la condizionata $H_{ij}|z_{ij}$ è una Gaussiana, pari a $p_1(h_{ij})$ se $z_{ij} = 1$ e pari a $p_2(h_{ij})$ altrimenti.

Nota una esplicita configurazione di valori z_{i1}, \dots, z_{iM} per un muone i , è immediato conoscere quale componente della mistura caratterizza i corrispondenti h_{i1}, \dots, h_{iM} . Di conseguenza, la distribuzione dei dati osservati d_i è facilmente esprimibile come in (2.1) se condizionata ai valori z_i

$$p(d_i|z_i) = p(d_i|z_{i1}, \dots, z_{iM})$$

$$= \frac{1}{2\pi|\tilde{\Sigma}_i|^{1/2}} \exp\left(-\frac{1}{2}d_i^T \tilde{\Sigma}_i^{-1} d_i\right).$$

La matrice di covarianza $\tilde{\Sigma}_i$ è pari a

$$\tilde{\Sigma}_i = p_{r,i}^2 \sum_{j=1}^M K^{1-z_{ij}} \lambda_j W_{ij}, \quad (3.4)$$

dove W_{ij} è la matrice dei pesi

$$W_{ij} = \begin{bmatrix} L_{ij} & L_{ij}^2/2 + L_{ij}T_{ij} \\ L_{ij}^2/2 + L_{ij}T_{ij} & L_{ij}^3/3 + L_{ij}^2T_{ij} + L_{ij}T_{ij}^2 \end{bmatrix}.$$

I calcoli per ottenere questa forma sono esplicitati in Appendice A.3.

Questi stessi ragionamenti valgono per lo *scattering* secondo y , ma è importante notare che la matrice $\tilde{\Sigma}_{y,i}$ può essere diversa rispetto a $\tilde{\Sigma}_{x,i}$, in quanto le variabili $z_{x,i}$ e $z_{y,i}$ possono assumere valori diversi.

3.2 Stima di massima verosimiglianza

3.2.1 Verosimiglianza completa e algoritmo EM

Per la stima di massima verosimiglianza di $\lambda = (\lambda_1, \dots, \lambda_M)$ il punto di partenza è ancora l'algoritmo EM come strumento operativo. All'iterazione $(q+1)$ dell'algoritmo, l'E-step prevede il calcolo della funzione ausiliaria

$$Q(\lambda, \lambda^{(q)}; d_x, d_y) = \mathbb{E}_{H_x, H_y, Z_x, Z_y | d_x, d_y, \lambda^{(q)}} [\ell^C(\lambda | d_x, d_y, H_x, H_y, Z_x, Z_y)]$$

e durante l'M-step si massimizza tale funzione per ottenere una stima di $\lambda^{(q+1)}$:

$$\lambda^{(q+1)} = \arg \max_{\lambda} Q(\lambda, \lambda^{(q)}; d_x, d_y).$$

Si determina quindi innanzitutto la distribuzione dei dati completi, ovvero la congiunta tra i dati osservati (D_x, D_y) e i dati mancanti, che in questo caso sono sia (H_x, H_y) che (Z_x, Z_y) . Poichè lo *scattering* lungo x è indipendente dallo *scattering* lungo y , la distribuzione congiunta sarà

$$p(d_x, d_y, h_x, h_y, z_x, z_y) = p(d_x, h_x, z_x) \cdot p(d_y, h_y, z_y).$$

La verosimiglianza associata ai dati completi risulta pari a

$$\ell^C(\lambda | d_x, d_y, h_x, h_y, z_x, z_y) = \ell^C(\lambda | d_x, h_x, z_x) + \ell^C(\lambda | d_y, h_y, z_y).$$

Si calcola infine il valore atteso della log-verosimiglianza completa per ottenere la funzione ausiliaria

$$\begin{aligned} Q(\lambda, \lambda^{(q)}; d_x, d_y) &= \mathbb{E}_{H_x, H_y, Z_x, Z_y | d_x, d_y, \lambda^{(q)}} [\ell^C(\lambda | d_x, H_x, Z_x) + \ell^C(\lambda | d_y, H_y, Z_y)] \\ &= \mathbb{E}_{H_x, Z_x | d_x, \lambda^{(q)}} [\ell^C(\lambda | d_x, H_x, Z_x)] + \mathbb{E}_{H_y, Z_y | d_y, \lambda^{(q)}} [\ell^C(\lambda | d_y, H_y, Z_y)]. \end{aligned} \quad (3.5)$$

È immediato riconoscere come sia possibile svolgere separatamente i valori attesi nell'ultimo passaggio. Di conseguenza, come è stato effettuato nel capitolo 2, si elude dal pedice l'indice che riguarda la coordinata e si procede con la descrizione del procedimento, sapendo che deve essere svolto una volta per lo *scattering* lungo l'asse x e una per quello lungo l'asse y .

Si ritorni alla distribuzione dei dati completi e se ne approfondisce la formulazione

$$\begin{aligned} p(d, h, z) &= p(d|h, z)p(h, z) = \prod_{i=1}^n p(d_i | h_{i1}, \dots, h_{iM}, z_{i1}, \dots, z_{iM}) p(h_i, z_i) \\ &\propto \prod_{i=1}^n \prod_{j=1}^M p(h_{ij}, z_{ij}) = \prod_{i=1}^n \prod_{j=1}^M \left\{ [\tau p_1(h_{ij})]^{z_{ij}} + [(1 - \tau) p_2(h_{ij})]^{1-z_{ij}} \right\}. \end{aligned}$$

Questa è stata ottenuta considerando il fatto che la distribuzione condizionata $D_i | h_i, z_i$ è degenere per ogni $i = 1, \dots, n$, in quanto i dati nascosti h_i sono di nota distribuzione per via della conoscenza degli z_i e nel loro insieme determinano univocamente la misura d_i .

La funzione di log-verosimiglianza per λ associata alla distribuzione dei dati completi diventa pari a

$$\begin{aligned} \ell^C(\lambda | h, z) &= \log \left(\prod_{i=1}^n \prod_{j=1}^M \left\{ [\tau p_1(h_{ij})]^{z_{ij}} + [(1 - \tau) p_2(h_{ij})]^{1-z_{ij}} \right\} \right) \\ &= \sum_{i=1}^n \sum_{j=1}^M \left\{ z_{ij} \left[\log \tau + \log p_1(h_{ij}) \right] + (1 - z_{ij}) \left[\log(1 - \tau) + \log p_2(h_{ij}) \right] \right\}. \end{aligned}$$

Si procede con l'E-step applicando alla verosimiglianza $\ell^C(\lambda | h, z)$ il valore atteso rispetto alla distribuzione dei dati mancanti condizionata ai dati

osservati $H, Z|d, \lambda^{(a)}$

$$\begin{aligned}
& \mathbb{E}_{H, Z|d, \lambda^{(a)}}[\ell^C(\lambda|d, H, Z)] = \\
& = \mathbb{E}_{H, Z|d, \lambda^{(a)}} \left\{ \sum_{i=1}^n \sum_{j=1}^M \left\{ Z_{ij} \left[\log \tau + \log p_1(H_{ij}) \right] + \right. \right. \\
& \qquad \qquad \qquad \left. \left. + (1 - Z_{ij}) \left[\log(1 - \tau) + \log p_2(H_{ij}) \right] \right\} \right\} \\
& = \sum_{j=1}^M \sum_{i: L_{ij} \neq 0} \mathbb{E}_{H_{ij}, Z_{ij}|d_i, \lambda^{(a)}} \left\{ \left\{ Z_{ij} \left[\log \tau - \frac{1}{2} \log |\Sigma_{ij}| - \frac{1}{2} H_{ij}^T \Sigma_{ij}^{-1} H_{ij} \right] + \right. \right. \\
& \qquad \qquad \qquad \left. \left. + (1 - Z_{ij}) \left[\log(1 - \tau) - \frac{1}{2} \log |K \Sigma_{ij}| - \frac{1}{2} H_{ij}^T (K \Sigma_{ij})^{-1} H_{ij} \right] \right\} \right\}.
\end{aligned} \tag{3.6}$$

Gli specifici valori attesi per la coppia i, j devono essere svolti rispetto a una distribuzione condizionata di forma non nota. Volendo semplificare, si calcola il valore atteso separatamente per i due addendi che compongono l'argomento della sommatoria. Nei seguenti passaggi si considera il primo addendo; gli svolgimenti per il secondo sono analoghi.

Risulta conveniente portare all'interno del valore atteso la sommatoria legata a j e si sfrutta la proprietà dello stesso

$$\begin{aligned}
& \mathbb{E}_{H_i, Z_i|d_i, \lambda^{(a)}} \left[\sum_{j=1}^M Z_{ij} \left(\log \tau - \frac{1}{2} \log |\Sigma_{ij}| - \frac{1}{2} H_{ij}^T \Sigma_{ij}^{-1} H_{ij} \right) \right] = \\
& = \mathbb{E}_{Z_i|d_i, \lambda^{(a)}} \left\{ \mathbb{E}_{H_i|Z_i, d_i, \lambda^{(a)}} \left[\sum_{j=1}^M Z_{ij} \left(\log \tau - \frac{1}{2} \log |\Sigma_{ij}| - \frac{1}{2} H_{ij}^T \Sigma_{ij}^{-1} H_{ij} \right) \right] \right\}.
\end{aligned}$$

Il valore atteso interno è calcolato rispetto alla distribuzione $(H_i|Z_i, d_i)$, una Gaussiana bidimensionale della stessa forma di (2.13). Infatti, poichè è data l'intera configurazione $z_i = (z_{i1}, \dots, z_{iM})$, è possibile conoscere sia la componente della miscela che caratterizza H_{ij} per ogni $j = 1, \dots, M$, che l'intera composizione del volume grazie alla quale si ottiene facilmente D_i . In particolare, noti i valori z_i , sia H_{ij} per ogni $j = 1, \dots, M$ che D_i sono Gaussiane, quindi la loro condizionata è

$$H_{ij}|z_i, d_i \sim N_2(\tilde{\Sigma}_{D_i H_{ij}}^T \tilde{\Sigma}_i^{-1} d_i, \Sigma_{ijg} - \tilde{\Sigma}_{D_i H_{ij}}^T \tilde{\Sigma}_i^{-1} \tilde{\Sigma}_{D_i H_{ij}}),$$

dove $\tilde{\Sigma}_{D_i H_{ij}}$ è la matrice di covarianza della distribuzione $H_{ij}, D_i | z_i$, $\tilde{\Sigma}_i$ è la matrice di covarianza di $D_i | z_i$ e Σ_{ijg} è la matrice di covarianza di $H_{ij} | z_i$ con $g = 1$ se $z_{ij} = 1$, $g = 2$ altrimenti. I primi due termini dell'argomento non dipendono da h_{ij} , dunque è sufficiente fare riferimento alla formula (2.14) per svolgere il valore atteso della forma quadratica $H_{ij}^T \Sigma_{ij}^{-1} H_{ij}$.

Si chiami $f(z_i)$ il risultato del valore atteso interno, che è essenzialmente una somma di M valori attesi, ognuno pre-moltiplicato per z_{ij} con $j = 1, \dots, M$. Ci si soffermi sul valore atteso esterno: esso è in funzione della configurazione $z_i = (z_{i1}, \dots, z_{iM})$

$$\begin{aligned} \mathbb{E}_{Z_i | d_i, \lambda^{(g)}}(f(z_i)) &= \sum_{\tilde{z}_i} f(\tilde{z}_i) p(\tilde{z}_i | d_i) = \sum_{\tilde{z}_i} f(\tilde{z}_i) \frac{p(d_i | Z_i = \tilde{z}_i) p(\tilde{z}_i)}{\sum_{\tilde{z}_i} p(d_i | Z_i = \tilde{z}_i) p(\tilde{z}_i)} \\ &= \sum_{\tilde{z}_i} f(\tilde{z}_i) \frac{p(d_i | Z_i = \tilde{z}_i) \prod_{j=1}^M \tau^{\tilde{z}_{ij}} (1 - \tau)^{1 - \tilde{z}_{ij}}}{\sum_{z_i} p(d_i | Z_i = z_i) \prod_{j=1}^M \tau^{\tilde{z}_{ij}} (1 - \tau)^{1 - \tilde{z}_{ij}}}, \end{aligned} \quad (3.7)$$

dove $p(d_i | Z_i = \tilde{z}_i)$ è una Gaussiana di media 0 e varianza $\tilde{\Sigma}_i$ in (3.4); sommare per ogni \tilde{z}_i significa aggiungere per ogni possibile configurazione di z_i .

Nel caso di un volume composto da M voxel, il numero di combinazioni è 2^M ; nel caso in cui un muone attraversasse $M = 30$ voxel, il valore atteso in (3.7) e il suo denominatore sommano più di 1 miliardo di addendi. Per questo motivo risulta impraticabile proseguire il calcolo della funzione ausiliaria mediante l'algoritmo EM.

3.3 Varianti dell'algoritmo EM

3.3.1 Lo *Stochastic* EM

Due varianti dell'algoritmo EM, adatte quando la funzione ausiliaria è difficile da calcolare o non ha soluzione in forma chiusa, sono gli algoritmi: *Classification* EM (CEM) e *Stochastic* EM (SEM), introdotti rispettivamente da Celeux e Govaert (1992) e Celeux e Diebolt (1985). Questi sono stati sviluppati come metodologie alternative per ottimizzare la verosimiglianza associata a una modello mistura, ovvero quando i dati mancanti sono gli indicatori

che identificano la componente della mistura, che in questa applicazione sono

$$z = \left\{ z_{ij} : 1 \leq i \leq n \ \& \ 1 \leq j \leq M \right\}.$$

Entrambi gli algoritmi prevedono uno step che assegna un valore ai dati mancanti z_1, \dots, z_M , per così poi proseguire semplicemente con i classici E-step e M-step dell'algoritmo EM.

Nell'algoritmo CEM questo ulteriore passo è chiamato *Classification*-step (C): all'iterazione (q) dell'algoritmo, il C-step consiste nell'assegnare l'osservazione d_i , per ogni $i = 1, \dots, n$, alla configurazione individuata da $z_i^{(q)}$ che produce la maggiore probabilità a posteriori

$$p(Z_i = z_i | d_i) = \frac{p(d_i | Z_i = z_i)p(Z_i = z_i)}{p(d_i)}, \quad (3.8)$$

dove per "assegnare" si intende calcolare la distribuzione D_i considerando la configurazione $z_i = z_i^{(q)}$.

L'algoritmo SEM è una versione stocastica del CEM: il C-step viene sostituito da uno *Stochastic*-step (S) che prevede la simulazione dei dati mancanti z_1, \dots, z_n . Nel dettaglio, all'iterazione (q) dell'algoritmo, durante l'S-step vengono generati casualmente gli $z_i^{(q)}$ per ogni $i = 1, \dots, n$ secondo la distribuzione condizionata $p(Z_i = z_i | d_i)$ e si assegna d_i alla configurazione simulata $z_i^{(q)}$.

Per approfondimenti sul SEM, si rimanda a Celeux, Chauveau e Diebolt (1992), Govaert e Nadif (2014, §2.4.3) e Biscarat, Celeux e Diebolt (1992).

In questo contesto l'algoritmo CEM risulta essere un'alternativa appropriata e valida, ma difficilmente praticabile. Infatti per ottenere al C-step la configurazione $z_i^{(q)}$ che produce la maggiore probabilità a posteriori in (3.8), bisogna calcolare tale probabilità per ogni configurazione degli M voxel, dunque 2^M volte. Questo numero è molto grande e comporta un notevole rallentamento nel processo di stima. Nonostante il problema del numero di combinazioni sia lo stesso segnalato per l'algoritmo EM, il CEM comunque semplifica molto il procedimento di stima in quanto, una volta assegnato un valore agli z_i per $i = 1, \dots, n$, i passi E e M risultano molto simili all'algoritmo EM esposto nel paragrafo 2.2.2. Al contrario, lo svolgimento dell'algoritmo EM a partire da (3.7) comporta lo sviluppo di una sommatoria comprendente 2^M termini.

Si decide per questo motivo di esporre nel dettaglio unicamente il processo di stima di massima verosimiglianza tramite l'algoritmo SEM.

3.3.2 I passi dell'algoritmo

La principale caratteristica che contraddistingue l'algoritmo SEM rispetto a un EM consiste nell'introduzione di un passo stocastico, in aggiunta ai passi EM.

- All'iterazione $(q + 1)$, al passo S dell'algoritmo si generano gli z_i per ogni $i = 1, \dots, n$ dalla loro distribuzione condizionata

$$p(Z_i = z_i | D_i) = \frac{p(d_i | Z_i = z_i)p(Z_i = z_i)}{p(d_i)}.$$

Poichè risulta complesso simulare una configurazione z_i da $Z_i | D_i$, si ricorre all'utilizzo di un algoritmo di simulazione. Alcune proposte sono riportate nel paragrafo 3.4 del capitolo successivo.

- Una volta generati i valori di $z_i^{(q+1)}$ per $i = 1, \dots, n$ mediante il passo S, è possibile procedere con il passo E come se fossero noti. Il valore atteso in (3.6), un addendo della funzione ausiliaria totale in (3.5), diventa allora

$$\begin{aligned} & \mathbb{E}_{H|d, z_i^{(q+1)}, \lambda^{(q)}} [\ell^C(\lambda | d, H, z)] \\ &= \sum_{j=1}^M \sum_{i: L_{ij} \neq 0} \mathbb{E}_{H_{ij}|d_i, z_i^{(q+1)}, \lambda^{(q)}} \left\{ \left\{ z_{ij}^{(q+1)} \left[\log \tau - \frac{1}{2} \log |\Sigma_{ij}| - \frac{1}{2} H_{ij}^T \Sigma_{ij}^{-1} H_{ij} \right] + \right. \right. \\ & \quad \left. \left. + (1 - z_{ij}^{(q+1)}) \left[\log(1 - \tau) - \frac{1}{2} \log |K \Sigma_{ij}| - \frac{1}{2} H_{ij}^T (K \Sigma_{ij})^{-1} H_{ij} \right] \right\} \right\} \\ &= \sum_{j=1}^M \sum_{i: L_{ij} \neq 0} \mathbb{E}_{H_{ij}|d_i, z_i^{(q+1)}, \lambda^{(q)}} \left\{ \left\{ z_{ij}^{(q+1)} l_{ij1} + (1 - z_{ij}^{(q+1)}) l_{ij2} \right\} \right\}. \end{aligned} \tag{3.9}$$

In base alla configurazione ottenuta per ogni coppia (i, j) con $i = 1, \dots, n$ e $j = 1, \dots, M$, si conserva solo un termine tra l_{ij1} e l_{ij2} .

Si consideri inizialmente il caso in cui $z_i^{(q+1)} = 1$; per la specifica coppia (i, j) si calcola

$$\begin{aligned}
\mathbb{E}_{H_{ij}|d_i, z_i^{(q+1)}, \lambda^{(q)}}(l_{ij1}) &= \mathbb{E}_{H_{ij}|d_i, z_i^{(q+1)}, \lambda^{(q)}} \left(\log \tau - \frac{1}{2} \log |\Sigma_{ij}| - \frac{1}{2} H_{ij}^T \Sigma_{ij}^{-1} H_{ij} \right) \\
&= -\frac{1}{2} \log |\Sigma_{ij}| - \frac{1}{2} \mathbb{E}_{H_{ij}|d_i, z_i^{(q+1)}, \lambda^{(q)}} (H_{ij}^T \Sigma_{ij}^{-1} H_{ij}) \\
&= -\frac{1}{2} \log \left(\lambda_j^2 p_{r,i}^4 \frac{L_{ij}^4}{12} \right) - \frac{1}{2} \mathbb{E}_{H_{ij}|d_i, z_i^{(q+1)}, \lambda^{(q)}} \left(H_{ij}^T \frac{B_{ij}^{-1}}{\lambda_j p_{r,i}^2} H_{ij} \right) \\
&\propto -\log \lambda_j - \frac{1}{2\lambda_j p_{r,i}^2} \mathbb{E}(H_{ij}^T B_{ij}^{-1} H_{ij} | d_i, z_i^{(q+1)}, \lambda^{(q)}).
\end{aligned} \tag{3.10}$$

Allo stesso modo si svolgono i calcoli nel caso in cui $z_{ij} = 0$ e si ottiene

$$\begin{aligned}
\mathbb{E}_{H_{ij}|d_i, z_i^{(q+1)}, \lambda^{(q)}}(l_{ij2}) &= \\
&= \mathbb{E}_{H_{ij}|d_i, \lambda^{(q)}, z_i^{(q+1)}} \left[\log(1 - \tau) - \frac{1}{2} \log |K\Sigma_{ij}| - \frac{1}{2} H_{ij}^T (K\Sigma_{ij})^{-1} H_{ij} \right] \\
&= -\frac{1}{2} \log |K\Sigma_{ij}| - \frac{1}{2} \mathbb{E}_{H_{ij}|d_i, z_i^{(q+1)}, \lambda^{(q)}} (H_{ij}^T (K\Sigma_{ij})^{-1} H_{ij}) \\
&= -\frac{1}{2} \log \left(K^2 \lambda_j^2 p_{r,i}^4 \frac{L_{ij}^4}{12} \right) - \frac{1}{2} \mathbb{E}_{H_{ij}|d_i, \lambda^{(q)}, z_i^{(q+1)}} \left(H_{ij}^T \frac{B_{ij}^{-1}}{K\lambda_j p_{r,i}^2} H_{ij} \right) \\
&\propto -\log \lambda_j - \frac{1}{2K\lambda_j p_{r,i}^2} \mathbb{E}(H_{ij}^T B_{ij}^{-1} H_{ij} | d_i, z_i^{(q+1)}, \lambda^{(q)}).
\end{aligned} \tag{3.11}$$

Per il calcolo di queste due casistiche, si è fatto riferimento ai seguenti risultati conseguiti a partire da (3.2) e (3.3)

$$\begin{aligned}
|\Sigma_{ij}| &= \lambda_j^2 p_{r,i}^4 \frac{L_{ij}^4}{12} \\
|K\Sigma_{ij}| &= K^2 \lambda_j^2 p_{r,i}^4 \frac{L_{ij}^4}{12} \\
\Sigma_{ij}^{-1} &= (\lambda_j p_{r,i}^2 B_{ij})^{-1} = \frac{1}{\lambda_j p_{r,i}^2} B_{ij}^{-1} \\
(K\Sigma_{ij})^{-1} &= (K\lambda_j p_{r,i}^2 B_{ij})^{-1} = \frac{1}{K\lambda_j p_{r,i}^2} B_{ij}^{-1}.
\end{aligned}$$

I valori attesi calcolati per $z_i^{(q+1)} = 1$ e $z_i^{(q+1)} = 0$ sono approssimativamente identici tranne per la presenza o meno di K al denominatore

del secondo termine. In entrambi i casi, il risultato ottenuto può procedere come nello sviluppo dell'algoritmo EM descritto nel paragrafo 2.2.2, ovvero senza considerare una mistura per la distribuzione dello *scattering*.

Allo scopo di esporre i seguenti passaggi senza distinguere le due casistiche, ci si serve della scrittura $K^{1-z_i^{(q+1)}}$ per identificare quando h_{ij} segue la distribuzione della prima o seconda componente della mistura, ovvero rispettivamente quando $z_i^{(q+1)} = 1$ e $z_i^{(q+1)} = 0$.

Innanzitutto si calcola $\mathbb{E}(H_{ij}^T B_{ij}^{-1} H_{ij} | d_i, z_i^{(q+1)}, \lambda^{(q)})$: si era considerato che la variabile casuale $H_{ij} | d_i, z_i^{(q+1)}, \lambda^{(q)}$ fosse Gaussiana, in particolare

$$H_{ij} | d_i, \lambda^{(q)}, z_i^{(q+1)} \sim N_2 \left(\tilde{\Sigma}_{D_i H_{ij}}^T \tilde{\Sigma}_i^{-1} d_i, (K^{1-z_i^{(q+1)}} \Sigma_{ij}) - \tilde{\Sigma}_{D_i H_{ij}}^T \tilde{\Sigma}_i^{-1} \tilde{\Sigma}_{D_i H_{ij}} \right),$$

dove $\tilde{\Sigma}_{D_i H_{ij}}$ è la matrice di covarianza di $H_{ij} | d_i, \lambda^{(q)}, z_i^{(q+1)}$.

Per il calcolo del valore atteso sopra esplicitato ci si ricollega al valore atteso della forma quadratica $H_{ij}^T B_{ij}^{-1} H_{ij} | d_i, \lambda^{(q)}, z_i^{(q+1)}$, giungendo così al risultato precedentemente trovato

$$\begin{aligned} \mathbb{E} \left(H_{ij}^T B_{ij}^{-1} H_{ij} | d_i, \lambda^{(q)}, z_i^{(q+1)} \right) &= \\ &= Tr \left(B_{ij}^{-1} (K^{1-z_i^{(q+1)}} \Sigma_{ij}) - B_{ij}^{-1} \tilde{\Sigma}_{D_i H_{ij}}^T \tilde{\Sigma}_i^{-1} \tilde{\Sigma}_{D_i H_{ij}} \right) + \\ &\quad + d_i^T \Sigma_i^{-1} (\tilde{\Sigma}_{D_i H_{ij}} B_{ij}^{-1} \tilde{\Sigma}_{D_i H_{ij}}^T) \tilde{\Sigma}_i^{-1} d_i. \end{aligned} \tag{3.12}$$

In Appendice A.4 sono riportati i calcoli per ottenere la matrice $\tilde{\Sigma}_{D_i H_{ij}}$ e alcune quantità per derivare la seguente uguaglianza, utilizzata per semplificare i calcoli del valore atteso in (3.12). Questa uguaglianza

$$\tilde{\Sigma}_{D_i H_{ij}} B_{ij}^{-1} \tilde{\Sigma}_{D_i H_{ij}}^T = W_{ij} (K^{1-z_{ij}^{(q+1)}} p_{r,i}^2 \lambda_j)^2$$

risulta da

$$\begin{aligned} \tilde{\Sigma}_{D_i H_{ij}} (K^{1-z_{ij}^{(q+1)}} \Sigma_{ij})^{-1} \tilde{\Sigma}_{D_i H_{ij}}^T &= K^{1-z_{ij}^{(q+1)}} \lambda_j p_{r,i}^2 W_{ij} \\ \rightarrow \tilde{\Sigma}_{D_i H_{ij}} \left(\frac{1}{K^{1-z_{ij}^{(q+1)}} \lambda_j p_{r,i}^2} B_{ij}^{-1} \right) \tilde{\Sigma}_{D_i H_{ij}}^T &= K^{1-z_{ij}^{(q+1)}} \lambda_j p_{r,i}^2 W_{ij} \\ \rightarrow \tilde{\Sigma}_{D_i H_{ij}} B_{ij}^{-1} \tilde{\Sigma}_{D_i H_{ij}}^T &= W_{ij} (K^{1-z_{ij}^{(q+1)}} \lambda_j p_{r,i}^2)^2. \end{aligned}$$

Il valore atteso in (3.12) viene così riscritto

$$\begin{aligned}
\mathbb{E}\left(H_{ij}^T B_{ij}^{-1} H_{ij} \mid d_i, z_i^{(q+1)}, \lambda^{(q)}\right) &= \\
&= \text{tr}\left[B_{ij}^{-1} K^{1-z_{ij}^{(q+1)}} \lambda_j^{(q)} p_{r,i}^2 B_{ij} - (\tilde{\Sigma}_{D_i H_{ij}} B_{ij}^{-1} \tilde{\Sigma}_{D_i H_{ij}}^T) \tilde{\Sigma}_i^{-1}\right] + \\
&\quad + d_i^T \tilde{\Sigma}_i^{-1} (W_{ij} (K^{1-z_{ij}^{(q+1)}} \lambda_j^{(q)} p_{r,i}^2)^2) \tilde{\Sigma}_i^{-1} d_i \\
&= \text{tr}\left[K^{1-z_{ij}^{(q+1)}} \lambda_j^{(q)} p_{r,i}^2 - W_{ij} (K^{1-z_{ij}^{(q+1)}} \lambda_j^{(q)} p_{r,i}^2)^2 \tilde{\Sigma}_i^{-1}\right] + \\
&\quad + d_i^T \tilde{\Sigma}_i^{-1} W_{ij} (K^{1-z_{ij}^{(q+1)}} \lambda_j^{(q)} p_{r,i}^2)^2 \tilde{\Sigma}_i^{-1} d_i \\
&= 2K^{1-z_{ij}^{(q+1)}} \lambda_j^{(q)} p_{r,i}^2 - (K^{1-z_{ij}^{(q+1)}} \lambda_j^{(q)} p_{r,i}^2)^2 \text{tr}(W_{ij} \tilde{\Sigma}_i^{-1}) + \\
&\quad + (K^{1-z_{ij}^{(q+1)}} \lambda_j^{(q)} p_{r,i}^2)^2 (d_i^T \tilde{\Sigma}_i^{-1} W_{ij} \tilde{\Sigma}_i^{-1} d_i) \\
&= K^{1-z_{ij}^{(q+1)}} p_{r,i}^2 \cdot \\
&\quad \cdot \left\{ 2\lambda_j^{(q)} + (\lambda_j^{(q)})^2 K^{1-z_{ij}^{(q+1)}} p_{r,i}^2 [d_i^T \tilde{\Sigma}_i^{-1} W_{ij} \tilde{\Sigma}_i^{-1} d_i - \text{tr}(\tilde{\Sigma}_i^{-1} W_{ij})] \right\},
\end{aligned}$$

sfruttando il fatto che B_{ij} e $\tilde{\Sigma}_i$ sono matrici simmetriche e che $\text{tr}(AB) = \text{tr}(BA)$.

Tutti gli svolgimenti e considerazioni appena elaborati si integrano nell'espressione della funzione ausiliaria $Q(\lambda, \lambda^{(q)}; d)$ in (3.9). Per semplificare la lettura si decide di denominare la quantità presente nei secondi termini di (3.10) e di (3.11)

$$\begin{aligned}
\tilde{G}_{ij}(\lambda^{(q)}, z_{ij}^{(q+1)}) &= \frac{1}{K^{1-z_{ij}^{(q+1)}} p_{r,i}^2} \mathbb{E}\left(H_{ij}^T B_{ij}^{-1} H_{ij} \mid d_i, z_{ij}^{(q+1)}, \lambda^{(q)}\right) \\
&= 2\lambda_j^{(q)} + (\lambda_j^{(q)})^2 K^{1-z_{ij}^{(q+1)}} p_{r,i}^2 [d_i^T \tilde{\Sigma}_i^{-1} W_{ij} \tilde{\Sigma}_i^{-1} d_i - \text{tr}(\tilde{\Sigma}_i^{-1} W_{ij})].
\end{aligned}$$

Questa quantità presenta il termine K se $z_{ij}^{(q+1)} = 0$ e di conseguenza per la coppia di indici (i, j) nella funzione ausiliaria viene conservato il termine l_{ij2} , al contrario si conserva l_{ij1} .

Il valore atteso in (3.9) si può quindi riscrivere come

$$\begin{aligned}
Q(\lambda, \lambda^{(q)}; d) &= \sum_{j=1}^M \sum_{i: L_{ij} \neq 0} \mathbb{E}_{H_{ij}|d_i, z_i^{(q+1)}, \lambda^{(q)}} \left\{ \left\{ z_{ij}^{(q+1)} l_{ij1} + (1 - z_{ij}^{(q+1)}) l_{ij2} \right\} \right\} \\
&\propto \sum_{j=1}^M \sum_{i: L_{ij} \neq 0} \left[-\log \lambda_j - \frac{1}{2\lambda_j} \tilde{G}_{ij}(\lambda^{(q)}, z_{ij}^{(q+1)}) \right] \\
&\propto \sum_{j=1}^M \left[-n_j \log \lambda_j - \frac{1}{2\lambda_j} \sum_{i: L_{ij} \neq 0} \tilde{G}_{ij}(\lambda^{(q)}, z_{ij}^{(q+1)}) \right],
\end{aligned}$$

dove n_j è il numero di muoni che passano per il voxel j .

Si riprende infine la scrittura della funzione ausiliaria completa delle informazioni delle due coordinate in (3.5): questa può essere riscritta come

$$\begin{aligned}
Q(\lambda, \lambda^{(q)}; d_x, d_y) &= \\
&= \mathbb{E}_{H_x, Z_x | d_x, \lambda^{(q)}} [\ell^C(\lambda | d_x, H_x, Z_x)] + \mathbb{E}_{H_y, Z_y | d_y, \lambda^{(q)}} [\ell^C(\lambda | d_y, H_y, Z_y)] \\
&\propto \sum_{j=1}^M \left[-n_j \log \lambda_j - \frac{1}{2\lambda_j} \sum_{i: L_{ij} \neq 0} \tilde{G}_{x,ij}(\lambda^{(q)}, z_{x,ij}^{(q+1)}) + \right. \\
&\quad \left. -n_j \log \lambda_j - \frac{1}{2\lambda_j} \sum_{i: L_{ij} \neq 0} \tilde{G}_{y,ij}(\lambda^{(q)}, z_{y,ij}^{(q+1)}) \right] \\
&\propto \sum_{j=1}^M Q_j(\lambda_j, \lambda^{(q)}; d_x, d_y).
\end{aligned}$$

- Si procede con il passo M e dunque con la massimizzazione della funzione ausiliaria $Q(\lambda; \lambda^{(q)}; d_x, d_y)$

$$\lambda^{(q+1)} = \arg \max_{\lambda} Q(\lambda, \lambda^{(q)}; d_x, d_y).$$

Come è stato considerato nella stima del parametro λ nel paragrafo 2.2.2, i termini che coinvolgono λ_j sono contenuti solo nel termine $Q_j(\lambda_j, \lambda^{(q)}; d_x, d_y)$ della sommatoria quando l'indice è proprio j ; per la massimizzazione della funzione ausiliaria secondo questa specifica λ_j è sufficiente calcolare direttamente la derivata su $Q_j(\lambda_j, \lambda^{(q)}; d_x, d_y)$.

Si deriva dunque il termine $Q_j(\lambda_j, \lambda^{(q)}; d_x, d_y)$ per λ_j , si eguaglia il risultato a 0 e si ottiene la formula iterativa per la sua stima al passo $q + 1$

$$\lambda_j^{(q+1)} = \frac{1}{2n_j} \sum_{i:L_{ij} \neq 0} \frac{\tilde{G}_{x,ij}(\lambda^{(q)}, \hat{z}_{x,ij}^{(q+1)}) + \tilde{G}_{y,ij}(\lambda^{(q)}, \hat{z}_{y,ij}^{(q+1)})}{2}.$$

3.4 Alcuni algoritmi di simulazione

Nel Paragrafo 3.3.2 è stato evidenziato che per il passo stocastico S dell'algoritmo SEM è utile, e in questo contesto essenziale, ricorrere a un algoritmo di simulazione, in quanto risulta difficile simulare dei valori direttamente dalla distribuzione $p(Z_i = z_i | d_i)$.

Si propongono in seguito alcuni algoritmi, utilizzati spesso in letteratura per ovviare al problema della simulazione dalla distribuzione condizionata all'S-step di un algoritmo SEM.

3.4.1 Algoritmi MCMC

Si considerano due metodi *Markov chain Monte Carlo* (MCMC): il *Gibbs Sampler* e il *Metropolis-Hastings*. Alcune applicazioni di questi metodi all'algoritmo SEM si ritrovano in Zanghi et al. (2008) e Kuhn e Lavielle (2004), rispettivamente per i due metodi citati.

Gli algoritmi MCMC sono di ampio utilizzo in ambito Bayesiano quando non è semplice campionare dalla distribuzione a posteriori. Questi metodi simulano una catena di Markov ergodica la cui distribuzione stazionaria è la distribuzione obiettivo, in questo caso $p(z_i | d_i)$. A prescindere dal valore iniziale $z_i^{(0)}$ della catena, per un numero grande di iterazioni k dell'algoritmo, $Z_i^{(k)}$ sarà approssimativamente $p(z_i | d_i)$ e i valori $z_i^{(k+1)}, z_i^{(k+2)}, z_i^{(k+3)}, \dots$ saranno campione pseudo-casuale della distribuzione obiettivo $p(z_i | d_i)$. Per maggiori approfondimenti sulle catene di Markov e i metodi MCMC si consulti Robert e Casella (2004, §6–7–9).

Si entra ora maggiormente nei dettagli applicativi dei due algoritmi.

Gibbs Sampler: questo algoritmo è considerato particolarmente conveniente in una applicazione multivariata e quando sono facilmente calcolabili

le distribuzioni condizionate delle variabili la cui congiunta è la distribuzione obiettivo. È necessario però che le distribuzioni congiunta e condizionate siano completamente note, in quanto l'algoritmo non è definito a meno di costanti di normalizzazione. Al principio dell'algoritmo SEM si inizializza il valore di $z_i^{(0)}$. All'iterazione $(q + 1)$ dell'algoritmo SEM, un'iterazione del *Gibbs Sampler* si svolge nel seguente modo:

1. si randomizza l'ordine degli elementi del vettore degli indici $(1, \dots, M)$ secondo una distribuzione uniforme;
2. per ogni elemento j del vettore degli indici randomizzati si assegna a $z_{ij}^{(q+1)}$ il valore 1 o 0 secondo la Bernoulliana

$$Be(p(Z_{ij} = 1|D_i, Z_{i,\setminus j})),$$

con $z_{i,\setminus j}$ il vettore $(z_{i1}, \dots, z_{i,j-1}, z_{i,j+1}, \dots, z_{iM})$ e

$$\begin{aligned} p(Z_{ij} = 1|d_i, z_{i,\setminus j}) &= \frac{p(Z_{ij} = 1, d_i, z_{i,\setminus j})}{\sum_{\tilde{z}_{ij}=0}^1 p(d_i, z_{i,\setminus j}|Z_{ij} = \tilde{z}_{ij})p(Z_{ij} = \tilde{z}_{ij})} \\ &= \frac{p(d_i|Z_{ij} = 1, z_{i,\setminus j})p(Z_{ij} = 1)p(z_{i,\setminus j})}{\sum_{\tilde{z}_{ij}=0}^1 p(d_i|Z_{ij} = \tilde{z}_{ij}, z_{i,\setminus j})p(Z_{ij} = \tilde{z}_{ij})p(z_{i,\setminus j})} \\ &= \frac{p(d_i|Z_{ij} = 1, z_{i,\setminus j}) \tau \prod_{k \neq j} \tau^{z_{ik}} (1 - \tau)^{1-z_{ik}}}{\sum_{\tilde{z}_{ij}=0}^1 p(d_i|Z_{ij} = \tilde{z}_{ij}, z_{i,\setminus j}) \tau^{\tilde{z}_{ij}} (1 - \tau)^{1-\tilde{z}_{ij}} \prod_{k \neq j} \tau^{z_{ik}} (1 - \tau)^{1-z_{ik}}}. \end{aligned}$$

Si osservi che i precedenti passaggi sono stati svolti considerando che gli z_{ij} per ogni $j = 1, \dots, M$ si distribuiscono in modo indipendente e identicamente distribuito tra loro. Inoltre, si noti che la distribuzione $(D_i|Z_{ij} = 1, Z_{i,\setminus j})$ è esplicitabile in quanto per ogni z_{i1}, \dots, z_{iM} è assegnato un valore e dunque si può conoscere quale componente della mistura segue lo *scattering* in ogni voxel.

Independent Metropolis-Hastings - 1° variante: per simulare dalla distribuzione $p(z_i|d_i)$, che è sufficiente sia nota a meno di una costante di proporzionalità, è richiesta la definizione della cosiddetta distribuzione delle proposte. Questa distribuzione regola la probabilità che l'algoritmo presenti uno spostamento della catena dal valore attuale $z_i^{(k+1)}$ al nuovo valore $z_i^{(m+1)}$, al passo

(k) dell'algoritmo. L'algoritmo associato a un campionamento per importanza utilizza come distribuzione delle proposte una distribuzione indipendente dai valori di z_i .

Inizializzato il valore $\hat{z}_i^{(0)}$ al primo passo del SEM, al passo $(q+1)$ un'iterazione dell'algoritmo *Metropolis-Hastings* consiste nei seguenti passaggi:

1. si simula il vettore z_i^* da una $Bi(M, \tau)$ o equivalentemente si simula z_{ij}^* da una $Be(\tau)$ per ogni $j = 1, \dots, M$, in quanto tra loro sono i.i.d.;
2. si calcola

$$\begin{aligned} \alpha &= \min \left\{ 1, \frac{p(Z_i = z_i^{(q+1)}) p(Z_i = z_i^* | d_i)}{p(Z_i = z_i^*) p(Z_i = z_i^{(q+1)} | d_i)} \right\} \\ &= \min \left\{ 1, \frac{\frac{p(Z_i = z_i^{(q+1)}) p(d_i | Z_i = z_i^*) p(Z_i = z_i^*)}{p(d_i)}}{\frac{p(Z_i = z_i^*) p(d_i | Z_i = z_i^{(q+1)}) p(Z_i = z_i^{(q+1)})}{p(d_i)}} \right\} \\ &= \min \left\{ 1, \frac{p(d_i | Z_i = z_i^*)}{p(d_i | Z_i = z_i^{(q+1)})} \right\}; \end{aligned}$$

3. si assegna a $z_i^{(q+1)}$ il nuovo valore z_i^* con probabilità α .

Independent Metropolis-Hastings - 2° variante: questa versione dell'algoritmo dovrebbe convergere più velocemente qualora si lavorasse in uno spazio multidimensionale e fosse difficile scegliere una buona distribuzione delle proposte. A partire dai valori inizializzati $z_i^{(0)}$, per ogni $j = 1, \dots, M$

1. si simula z_{ij}^* da una $Be(\tau)$;
2. si pone $z_i^* = (z_{i1}^{(q+1)}, \dots, z_{i,j-1}^{(q+1)}, z_{ij}^*, z_{i,j+1}^{(q+1)}, \dots, z_{iM}^{(q+1)})$;
3. supponendo che la distribuzione proposta sia la medesima per ogni $j = 1, \dots, M$ si calcola

$$\begin{aligned} \alpha_j &= \min \left\{ 1, \frac{p_j(Z_i = z_i^{(q+1)} | Z_i = z_{ij}^*) p(Z_i = z_i^* | d_i)}{p_j(Z_i = z_{ij}^* | Z_i = z_i^{(q+1)}) p(Z_i = z_i^{(q+1)} | d_i)} \right\} \\ &= \min \left\{ 1, \frac{\frac{p_j(Z_i = z_i^{(q+1)}, Z_i = z_{ij}^*)}{p(Z_i = z_{ij}^*)} \cdot \frac{p(d_i | Z_i = z_i^*) p(Z_i = z_i^*)}{p(d_i)}}{\frac{p_j(Z_i = z_i^{(q+1)}, Z_i = z_{ij}^*)}{p(Z_i = z_i^{(q+1)})} \cdot \frac{p(d_i | Z_i = z_i^{(q+1)}) p(Z_i = z_i^{(q+1)})}{p(d_i)}} \right\} \\ &= \min \left\{ 1, \frac{p(d_i | Z_i = z_i^*) p(Z_i = z_i^*)}{p(d_i | Z_i = z_i^{(q+1)}) p(Z_i = z_{ij}^*)} \right\}; \end{aligned}$$

4. si assegna a $z_{ij}^{(q+1)}$ il nuovo valore z_{ij}^* con probabilità α_j .

3.4.2 Algoritmi ABC

L'algoritmo *Approximate Bayesian Computation* (ABC) è un metodo che non richiede la conoscenza della verosimiglianza; una sua applicazione al caso SEM è presente in Picchini e Samson (2018). Gli algoritmi *likelihood-free* si applicano quando una verosimiglianza è computazionalmente intrattabile o non disponibile e consistono nell'effettuare le classiche analisi basate su di essa senza una sua valutazione numerica. L'ABC, in particolare, è sviluppato per produrre un'approssimazione di una distribuzione obiettivo da cui si vuole simulare. Per una trattazione completa sui metodi senza verosimiglianza e l'ABC si rimanda a Sisson, Fan e Beaumont (2018).

Si propone un algoritmo ABC del tipo *accetto-rifiuto*.

Acceptance-Rejection ABC: si propone per questo contesto un campionamento accetto-rifiuto ABC, che consiste nell'iterazione dei seguenti tre passi, dopo aver inizializzato il valore di $z_i^{(0)}$ al primo passo del SEM. All'iterazione $(q + 1)$ del SEM, un'iterazione dell'ABC è così caratterizzata:

1. si simula z_i^* dalla distribuzione a priori $Bi(M, \tau)$ o equivalentemente si simula z_{ij}^* da una $Be(\tau)$ per ogni $j = 1, \dots, M$, in quanto tra loro sono i.i.d.;
2. si simula d_i^* dalla distribuzione $D_i|Z_i = z_i^*$;
3. si assegna a $z_i^{(q+1)}$ il nuovo valore z_i^* se $\|d_i - d_i^*\| < \varepsilon$.

A differenza dei metodi MCMC presentati prima, questo algoritmo coinvolge alcune quantità la cui definizione è importante per l'efficacia dell'algoritmo. Innanzitutto il parametro soglia ε è il tasso di accettazione dell'algoritmo e quando cresce fa allontanare la distribuzione obiettivo dalla sua vera forma. Esistono alcuni algoritmi, come quello APCM di Lenormand, Jabot e Defuant (2013) che adattano una soglia mobile. La distanza calcolata tra le osservazioni originali e simulate al passo (3.) può essere una qualsiasi funzione di distanza parametrica, come quella di Mahalanobis, oppure non parametrica, come ad esempio l'entropia o il *k-nearest neighbors*.

Inoltre per diminuire la dimensionalità del confronto $\|d_i - d_i^*\|$ al passo (3.), è possibile sostituirlo con quello tra $\|s_i - s_i^*\|$, dove $s_i = S(d_i)$ è una statistica sufficiente scelta con il vincolo che $\dim(s) \ll \dim(d_i)$.

Capitolo 4

Un'applicazione a dati simulati

In questo capitolo si presentano e discutono i risultati di un'applicazione delle metodologie descritte nei capitoli 2 e 3. I dati utilizzati sono relativi ad un esperimento condotto presso il Dipartimento di Fisica Moderna all'Università di Cantabria, volto a valutare il livello di usura nelle pareti interne di tubi di isolamento attraverso la ricostruzione dei materiali che compongono gli strati del tubo con i relativi spessori.

4.1 Descrizione dei dati

I dati a disposizione simulano un tratto di tubo isolante di raggio pari a 20 cm , costituito da un tubo in acciaio ricoperto esternamente da uno strato di lana di roccia, il tutto avvolto da una copertura in alluminio. Lo strato isolante esterno ha spessore di 2 cm . Si veda la Figura 4.1 per un'illustrazione del volume da ricostruire.

Per registrare il passaggio dei muoni all'interno del volume oggetto di studio sono stati predisposti due rilevatori, uno posto superiormente al tubo isolante e uno inferiormente, di superficie pari a $90 \times 90\text{ cm}^2$ e distanza pari a 78 cm . Assumendo che l'origine degli assi coincida con un vertice del rilevatore inferiore del volume oggetto di studio, il tubo isolante si trova esattamente centrato nella coordinata $(0, 0, 0)$. Ogni rilevatore è costituito da 4 strati, che a coppie registrano la posizione lungo la coordinata x e lungo la coordinata y

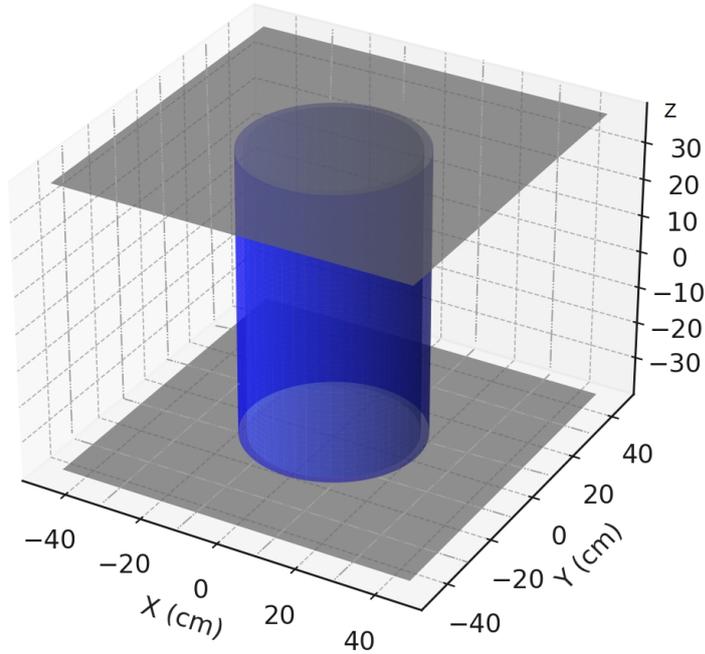


Figura 4.1: Schema illustrativo del volume da ricostruire

del muone che lo attraversa. La posizione z in entrata e uscita è invece data dalla posizione nello spazio dei due rivelatori.

Lo scopo dell'applicazione è di ricostruire il tubo isolante, distinguendo inoltre i tre distinti materiali che lo compongono, all'interno del volume da investigare.

Per ogni muone sono disponibili le seguenti misure:

- (x_{IN}, y_{IN}, z_{IN}) : posizione registrata al rivelatore superiore, in cm ;
- $(x_{OUT}, y_{OUT}, z_{OUT})$: posizione registrata al rivelatore inferiore, in cm ;
- $(v_{x_{IN}}, v_{y_{IN}}, v_{z_{IN}})$: vettore direzionale che descrive la direzione della traiettoria al passaggio del rivelatore superiore;
- $(v_{x_{OUT}}, v_{y_{OUT}}, v_{z_{OUT}})$: vettore direzionale che descrive la direzione della traiettoria al passaggio del rivelatore inferiore;
- e_{IN} : energia registrata al rivelatore superiore, in MeV ;

- e_{OUT} : energia registrata al rivelatore inferiore, in MeV .

Si evidenzia il fatto che la misura dell'energia in questo caso è nota in quanto i dati sono stati generati tramite simulazioni. Un rivelatore nel caso reale però non può misurare l'energia di un muone, a meno che non sia in grado di rallentarne e arrestarne la corsa. Ciò che viene misurato è in genere la perdita di energia per ionizzazione che il muone produce nell'attraversare un materiale.

L'insieme dei dati comprende due tipologie di misurazioni. Le prime sono registrate da un particolare rivelatore, chiamato camera proporzionale a multifili (*MultiWire Proportional Chamber* - MWPC), costituito da una serie di fili ad alta tensione positiva disposti ogni 4 mm tra due piani metallici. Si riesce a misurare la posizione del muone in base ai fili che vengono attivati al passaggio della particella. La risoluzione intrinseca dell'MWPC, che misura di quanto precisamente il sistema può determinare la posizione del passaggio del muone, è pari a $4/\sqrt{12}$.

Le seconde misurazioni raccolte invece sono "ideali" perchè simulate da un rivelatore perfetto, ovvero caratterizzato da infinita precisione. Le analisi che seguono fanno riferimento alle misure ideali, in quanto le metodologie espone nei capitoli 2 e 3 sono state sviluppate assumendo che i rivelatori non commettano errore di misurazione. Si rimanda a Schultz et al. (2007) per una discussione sulla possibilità di tener conto di errori di misurazione nei modelli considerati.

4.2 Alcune analisi preliminari

Per la stima del parametro di interesse λ , bisogna calcolare innanzitutto le quantità necessarie che entrano nella specificazione del modello; per ogni muone:

- angolo di *scattering* (a_x, a_y) ;
- dislocamento (u_x, u_y) ;
- p_r^2 ;

- L_j, T_j per ogni $j = 1, \dots, M$.

L'angolo di *scattering* lungo ogni coordinata è dato dalla differenza tra gli angoli di entrata e uscita. (Si vedano le formule in (1.2)). A partire dai vettori che misurano la direzione della traiettoria del muone al passaggio di un rivelatore, è immediato calcolare l'angolazione di tale traiettoria, ad esempio per l'angolo in entrata rispetto all'asse x , tramite

$$\theta_{x,IN} = \arccos \left(\frac{v_{zIN}}{\sqrt{v_{zIN}^2 + v_{xIN}^2}} \right).$$

Il dislocamento si ottiene sempre distinguendolo per le coordinate x e y , tramite le formule in (1.3). Il calcolo dipende dalla quantità definita in corrispondenza di (1.1), $l_{xy} = \sqrt{1 + \tan^2_{\theta_{x,IN}} + \tan^2_{\theta_{y,IN}}}$, e dal punto (x_p, y_p, z_p) , il punto di uscita del muone se la sua traiettoria non venisse deviata. Le coordinate del punto (x_p, y_p, z_p) si ottengono nel seguente modo

$$\begin{cases} x_p &= x_{IN} + \frac{v_{xIN} * (z_{OUT} - z_{IN})}{v_{zIN}} \\ y_p &= y_{IN} \frac{v_{yIN} * (z_{OUT} - z_{IN})}{v_{zIN}} \\ z_p &= z_{OUT} \end{cases} .$$

Il valore di $p_r = p/p_0$, dove il momento p e il momento nominale p_0 del muone sono ottenuti rispettivamente come

$$p_0 = \sqrt{e_{IN}^2 - m^2} \quad p = \sqrt{e_{OUT}^2 - m^2},$$

e $m \approx 106 \text{ MeV}$ è la massa del muone.

Infine, L_j e T_j vengono calcolati per ogni $j = 1, \dots, M$, tramite il procedimento esposto nel paragrafo 1.4 e, rispettivamente, mediante la (1.6).

L'insieme dei dati simulato contiene informazioni per 306984 muoni. La distribuzione marginale degli angoli di *scattering* secondo le due coordinate è raffigurata in Figura 4.2 e la distribuzione marginale del dislocamento distinto per le due coordinate è rappresentata in Figura 4.3.

Si nota che entrambe le distribuzioni sono simmetriche intorno allo 0. Gli angoli di *scattering* rispetto all'asse x variano tra $(-0.40^\circ, 0.47^\circ)$ e quelli rispetto all'asse y variano tra $(-0.38^\circ, 0.52^\circ)$. I dislocamenti secondo x e secondo y variano invece rispettivamente tra $(-26.07 \text{ cm}, 18.63 \text{ cm})$ e tra $(-18.81 \text{ cm}, 19.09 \text{ cm})$.

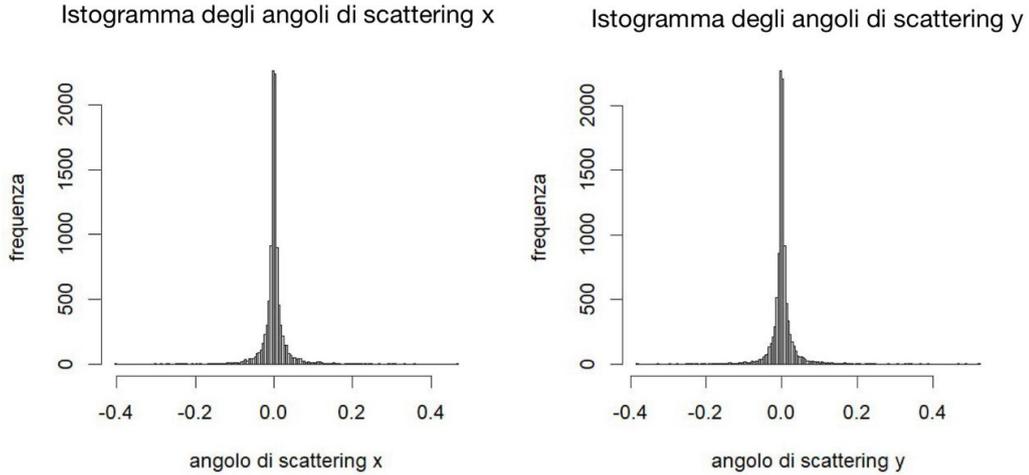


Figura 4.2: Istogramma di frequenza degli angoli di *scattering* relativi alla coordinata x a sinistra e y a destra.

4.3 Applicazione dei modelli

4.3.1 Alcuni dettagli di implementazione

La ricostruzione del tubo isolante è stata effettuata attraverso l'applicazione dei modelli discussi nei Capitoli 2 e 3. Gli algoritmi di stima associati ai due modelli sono stati implementati nel linguaggio R (R Core Team (2017)). Al fine di contenere i tempi computazionali, con riferimento al modello proposto e all'algoritmo di stima SEM, si è proceduto all'applicazione della sola variante che implementa il passo di simulazione mediante l'algoritmo *Metropolis-Hastings*, e in particolare la sua prima versione descritta in 3.4.1, perché computazionalmente meno oneroso rispetto alle altre varianti presentate.

Si discutono nel seguito alcune scelte per l'implementazione dei due metodi.

- *Inizializzazione degli algoritmi:* in via preliminare si è proceduto a valutare diverse inizializzazioni di entrambi gli algoritmi, utilizzando un medesimo valore di $\lambda_j^{(0)}$ per tutti i voxel $j = 1, \dots, M$, e valori differenti in corrispondenza di diverse regioni del volume. I risultati di tutte

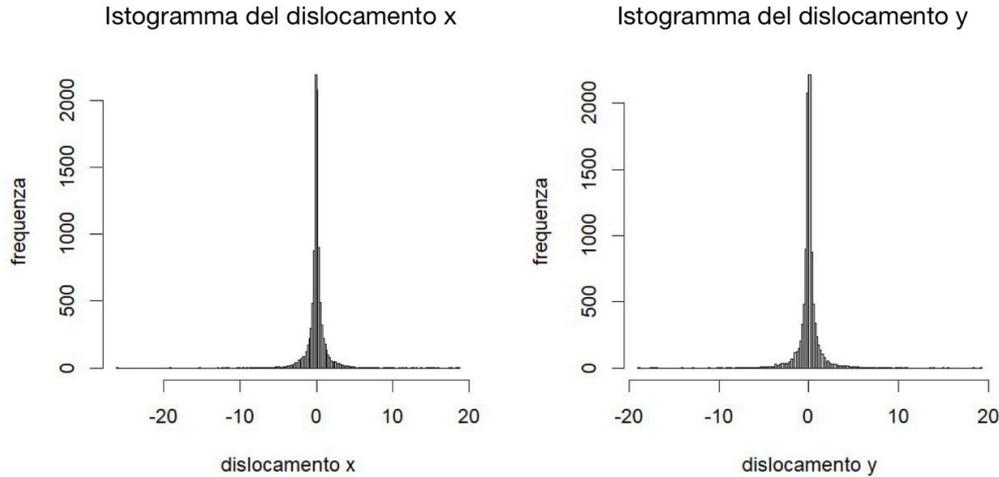


Figura 4.3: Istogramma di frequenza dei dislocamenti relativi alla coordinata x a sinistra e y a destra.

le prove hanno portato a conclusioni molto simili e si è pertanto proceduto ad inizializzare i due algoritmi ponendo $\lambda_j^{(0)} = 0.01$ per ogni $j = 1, \dots, M$, vale a dire assumendo che l'intero volume sia vuoto. Questa inizializzazione appare la più sensata anche in linea generale, quando non si conosce nulla degli ipotetici oggetti contenuti in un volume da investigare.

- *Tuning*: in seguito all'attuazione di diverse prove dell'algoritmo SEM al variare del valore di K , si è posto per la costante di proporzionalità un valore pari a 20, perchè abbastanza grande da produrre una differenza tra l'uso di una distribuzione Gaussiana e quello di un modello mistura.
- *Numero di iterazioni dell'algoritmo di simulazione al passo stocastico S nel SEM*: questo si fissa pari a 5. In alcuni esempi in letteratura, come Zanghi et al. (2008), viene effettuata un'unica iterazione dell'algoritmo di simulazione per ogni iterazione del SEM. Effettuando invece più di una iterazione, si garantisce che l'algoritmo di simulazione abbia sufficienti opportunità per convergere verso una stima più accurata, permettendo così al SEM di operare efficacemente.

- *Criterio di convergenza*: data la natura iterativa di entrambi gli algoritmi, si è imposto come criterio di convergenza che nelle ultime tre iterazioni la log-verosimiglianza non variasse di oltre lo 0.03%.

4.3.2 Ricostruzione dell'intero tubo isolante

In prima analisi gli algoritmi sono stati applicati al fine di ricostruire il tubo isolante nella sua interezza. Per contenere i tempi computazionali, tuttavia, e avendo il tubo isolante un diametro esterno pari a 40 cm , si è scelto di restringere l'area di base del volume ad un quadrato di lato 50 cm , il cui centro coincide con il centro del tubo in $(0, 0, 0)$, in modo da escludere regioni vuote (Figura 4.4). Inoltre, dei 77673 muoni la cui traiettoria attraversa questo volume ristretto, ne sono stati selezionati casualmente 10000.

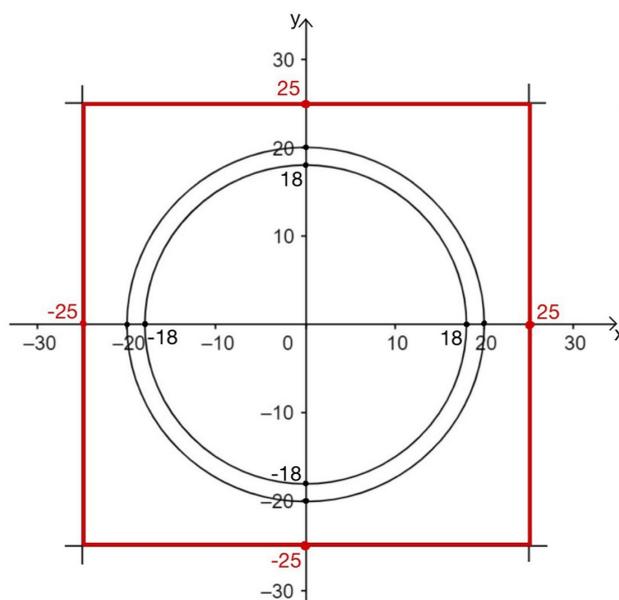


Figura 4.4: Area di base del volume ridotto.

Il volume in esame è stato suddiviso in 24375 voxel cubici di lato 2 cm , nello specifico ottenendo 25 voxel di base sull'asse x e sull'asse y e 39 voxel sull'asse z .

Tabella 4.1: Risultati degli algoritmi EM e SEM per la ricostruzione del volume ristretto.

	Algoritmo EM	Algoritmo SEM
iterazioni	966	984
range di $\hat{\lambda}$	(0, 0.048)	(0, 0.047)
$\hat{\lambda} \geq 20$	0	0
$3 \leq \hat{\lambda} < 20$	0	0
$0.25 \leq \hat{\lambda} < 3$	0	0

Entrambi gli algoritmi hanno raggiunto la convergenza in poco meno di 1000 iterazioni (Tabella 4.1) producendo risultati molto simili. La Figura 4.5 riporta le distribuzioni dei parametri λ_j stimati dai due algoritmi.

Si nota che per entrambi gli algoritmi, i valori stimati di λ sono molto bassi, ben oltre la soglia minima 0.25 che rileva un materiale a basso numero atomico Z . Infatti la distribuzione dei valori stimati λ_j per $j = 1, \dots, M$ raffigurata in Figura 4.5 è fortemente asimmetrica verso lo 0. In Figura 4.6 sono riportate alcune sezioni centrali di ogni piano parallelo alle facce del volume. Ogni sezione è un'immagine costituita da voxel di colore proporzionale all'intensità del valore di λ stimato, in una scala dal bianco al nero i cui gli estremi coincidono con il valore più basso e più alto di λ ottenuto.

Dalla sezione parallela al piano xy si nota che entrambi gli algoritmi riescono a cogliere la presenza di un oggetto di base circolare, seppur in maniera molto grezza, in quanto non sono nè distinguibili i tre diversi strati che compongono lo strato isolante del tubo, nè è stato colto il fatto che l'interno del tubo sia cavo.

Per quanto riguarda le sezioni parallele ai piani yz e xz si riscontrano le stesse criticità. Queste sezioni tagliano il tubo verticalmente al centro, dunque dovrebbero presentare anch'esse una zona vuota centrale e i tre strati di materiale ai due estremi opposti della sezione. Le sezioni parallele al piano yz , a differenza di quelle parallele al piano xz che appaiono molto confuse, lasciano agli estremi due parti bianche, a indicare correttamente che lì non è presente alcun materiale.

Si ritiene che la scarsa accuratezza dei risultati sia ascrivibile ad una serie

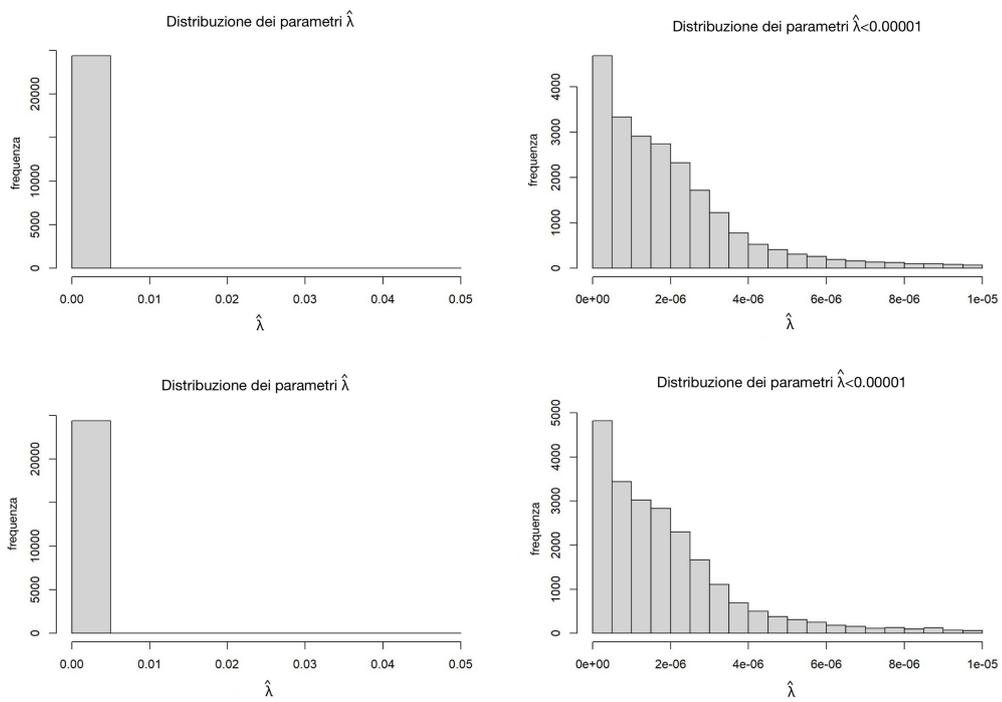


Figura 4.5: Istogramma di frequenza di λ stimato tramite l'algoritmo EM (in alto) e SEM (in basso), nel caso del volume ridotto.

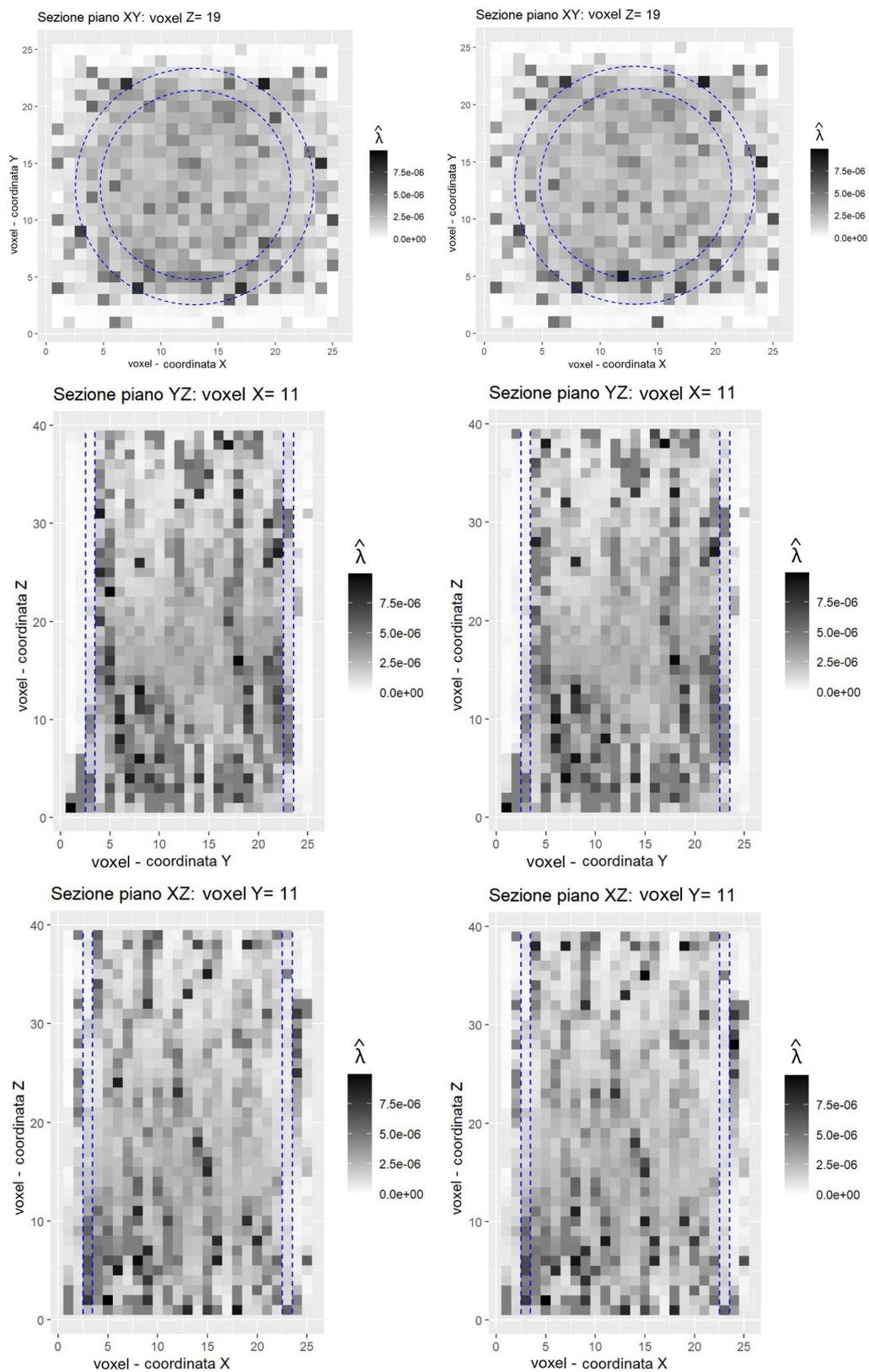


Figura 4.6: Sezioni del volume ricostruito, a sinistra tramite l'algorithmo EM e a destra tramite l'algorithmo SEM. In blu i confini del tubo isolante.

di ragioni di seguito discusse. Per quanto non siano note le vere densità di *scattering* λ , ci si aspetta di rilevare valori più elevati di λ per l'acciaio, a seguire per l'alluminio e infine i valori più bassi per la lana di roccia. Tuttavia, nessuno dei materiali che costituiscono il tubo isolante è caratterizzato da un alto numero atomico Z , e quindi non fa parte dei materiali più immediati da individuare attraverso le tecniche tomografiche (Borozdin et al. (2003)). Infatti, Schultz et al. (2004) propongono la seguente distinzione: se il valore di un λ_j stimato è maggiore o uguale a 20, allora questo è relativo a un materiale ad alto numero atomico Z ; se $3 \leq \lambda < 20$, questa densità di *scattering* caratterizza un materiale a medio Z ; se $0.25 \leq \lambda < 3$ allora si riferisce a un materiale a basso Z .

In secondo luogo, per far raggiungere agli algoritmi la convergenza in tempi contenuti, è stato necessario utilizzare voxel di lato non inferiore a 2 cm. Ciò significa, tuttavia, che ciascun voxel possa contenere in sé sia una parte di volume effettivamente vuota che piena di un certo materiale. Di conseguenza, gli algoritmi non hanno stimato alti valori λ perché ogni voxel è di composizione eterogenea e dunque la densità di *scattering* stimata risulta una media tra valori alti e valori quasi nulli. Infine, poiché lo strato di isolamento è spesso in totale solo 2 cm e i voxel sono di lato 2 cm, il motivo per cui non si riescono a distinguere i tre diversi materiali è il medesimo appena esposto legato alla problematica di eterogeneità.

4.3.3 Ricostruzione di una porzione del tubo isolante

Per i motivi ora discussi, gli algoritmi sono stati successivamente applicati ad una porzione di volume ulteriormente ristretta ma suddivisa in voxel più piccoli. Quale area di base di tale volume si è considerato un rettangolo di coordinata x tra $(-6, 6)$ e coordinata y tra $(-22, -14)$, come rappresentato nella Figura 4.7. La porzione di volume considerata è stata suddivisa in 59904 voxel cubici di lato 0.5 cm, dove si contano 24 voxel di base sull'asse x , 16 sull'asse y e 156 sull'asse z . Sono stati individuati 9113 muoni passanti per tale porzione di volume.

Rispetto all'applicazione precedente al volume ridotto, gli algoritmi impiegano molte più iterazioni per convergere, in particolar modo l'algoritmo

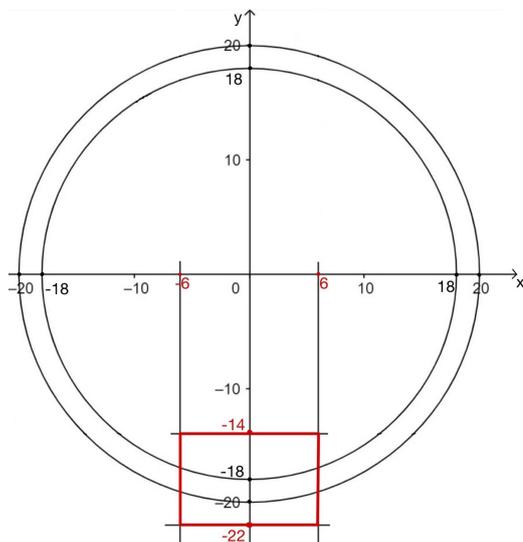


Figura 4.7: Area di base del volume ridotto.

Tabella 4.2: Risultati degli algoritmi EM e SEM per la ricostruzione della porzione di volume.

	Algoritmo EM	Algoritmo SEM
iterazioni	2066	1250
range di $\hat{\lambda}$	(0, 13.30)	(0, 3.37)
$\hat{\lambda} \geq 20$	0	0
$3 \leq \hat{\lambda} < 20$	1	1
$0.25 \leq \hat{\lambda} < 3$	14	13

EM raggiunge le 2000 iterazioni, mentre l'algoritmo SEM risulta più veloce (Tabella 4.2).

La distribuzione dei valori stimati λ_j per $j = 1, \dots, M$ raffigurata in Figura 4.8 è ancora asimmetrica verso lo 0. Tuttavia in questo caso si osserva che sono stati stimati alcuni valori di λ maggiori di 0.25. Questo risultato è positivo rispetto a quello relativo al volume ridotto, sebbene sulla totalità di 59904 parametri stimati, questi numeri siano ancora molto bassi nell'ottica che questi voxel rappresentino tre diversi strati di materiale.

Nelle Figure 4.9 e 4.10 sono riportate, rispettivamente per l'algoritmo EM e per il SEM, le sezioni centrali secondo ogni piano parallelo alle facce

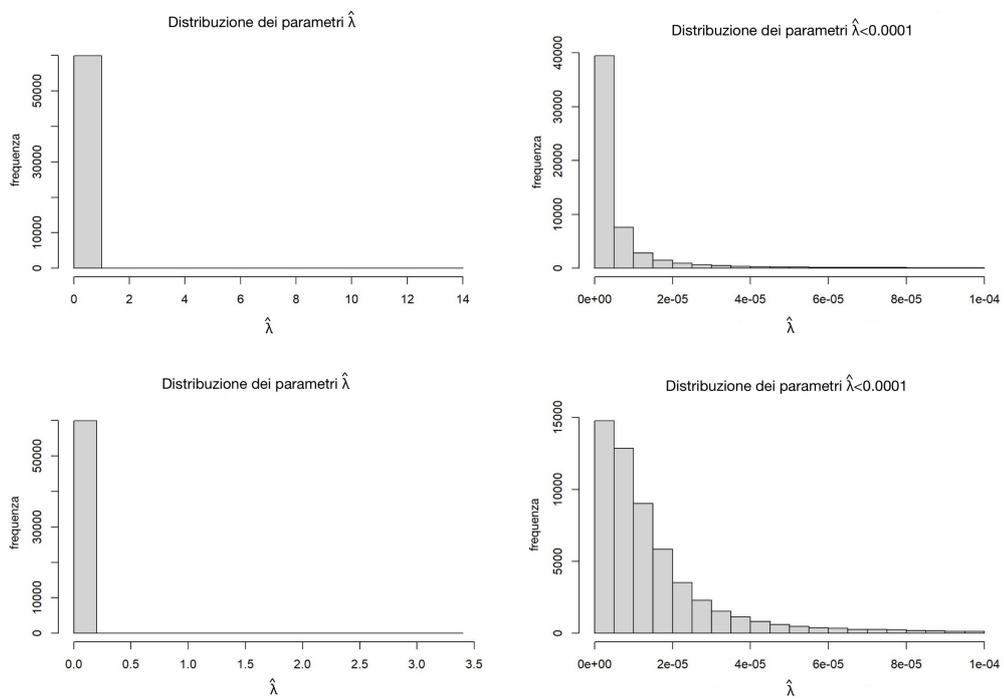


Figura 4.8: Istogramma di frequenza di λ stimato tramite l'algoritmo EM (in alto) e SEM (in basso), nel caso della porzione di volume.

del volume. I valori di λ stimati con l'algoritmo EM sono tendenzialmente più bassi rispetto ai corrispettivi ottenuti tramite l'algoritmo SEM. Con riferimento a quest'ultimo, nella sezione parallela al piano xy si può osservare una lieve curvatura scura in corrispondenza dell'area in cui si estende il tubo isolante. Per quanto riguarda la sezione parallela al piano yz , si nota che è stata colta la distinzione tra l'area del volume vuota e la presenza del tubo. Sono state presentate due sezioni parallele al piano xz perchè, per come è stata definita la porzione di volume, le sezioni iniziali dovrebbero risultare vuote, mentre quelle centrali dovrebbe cogliere la presenza di un oggetto. Si può osservare questo aspetto nelle due sezioni presentate; in particolare nella sezione individuata dal voxel y numero 6 si coglie una macchia più scura al centro, a significare che il tubo possa estendersi in altezza in quell'area. Le immagini ricostruite tramite l'utilizzo dell'algoritmo EM non colgono tali aspetti in maniera altrettanto evidente.

Nonostante sia riconoscibile una maggiore definizione nelle immagini delle sezioni che ricostruiscono la porzione di volume in questa seconda applicazione, è indubbio che queste siano ancora molto grezze, ma è verosimile supporre che aumentando il numero di muoni e riducendo ulteriormente la dimensione dei voxel l'accuratezza possa migliorare ulteriormente, seppur con un inevitabile aumento dello sforzo computazionale.

4.4 Discussione

A seguito dell'implementazione dei due algoritmi per la ricostruzione di questo volume parziale, si può affermare che in questa applicazione è l'algoritmo SEM legato alla nuova metodologia proposta ad aver colto maggiormente la presenza del tubo isolante all'interno del volume da investigare. Inoltre questo risultato è stato ottenuto con quasi la metà delle iterazioni che ha impiegato l'algoritmo EM a soddisfare il medesimo criterio di convergenza. Per dichiarare con certezza che il metodo proposto ottenga migliori risultati rispetto a quello di riferimento, sarebbe però necessario approfondire le applicazioni di entrambi i metodi in un contesto di lavoro molto dettagliato. Nelle applicazioni del metodo di Schultz et al. (2007), per esempio, si contano non meno di 100000 muoni, in un contesto di voxel cubici di lato 5 cm. Dunque

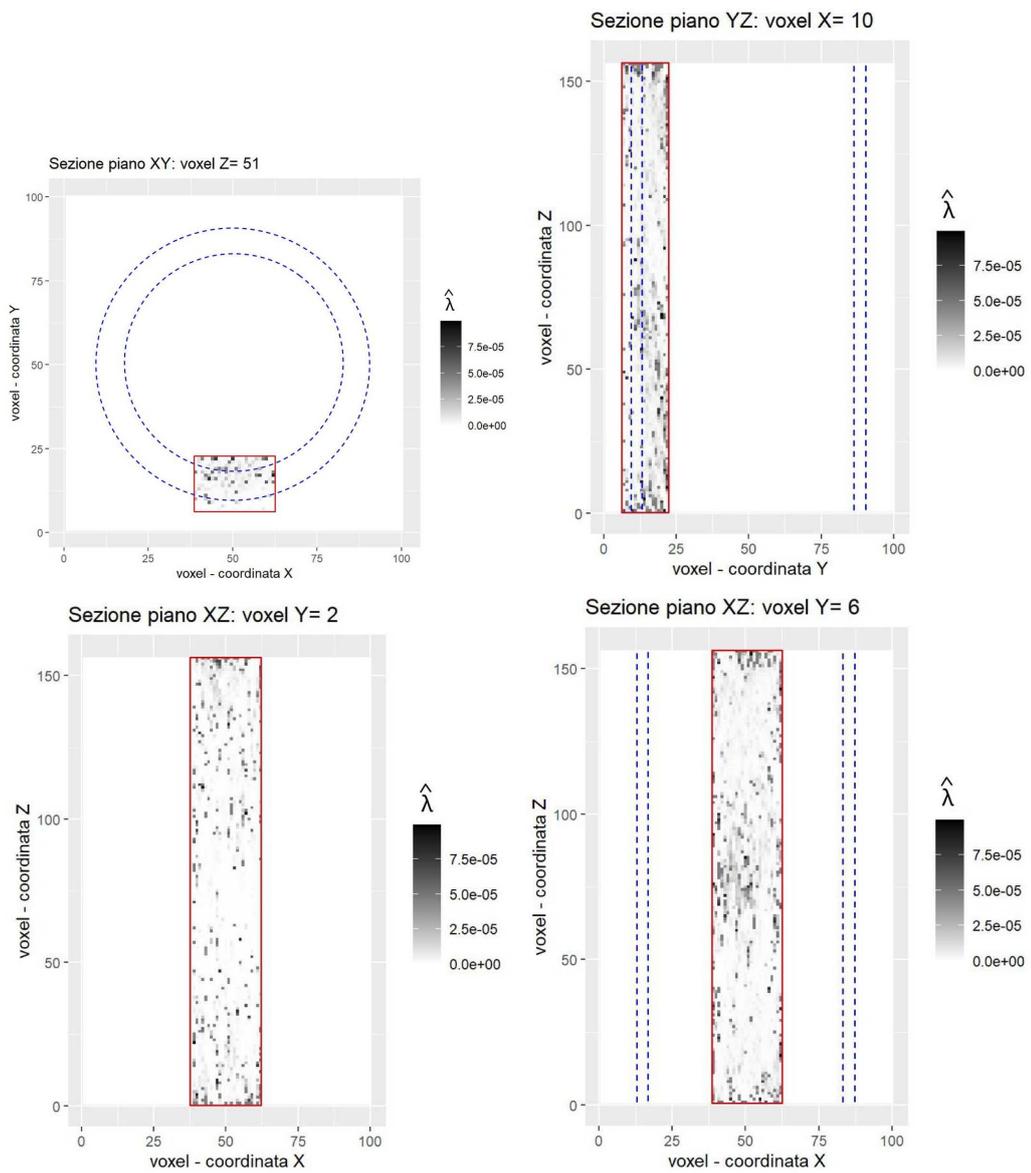


Figura 4.9: Sezioni della porzione di volume ricostruita tramite l'algoritmo EM. In blu i confini del tubo isolante.

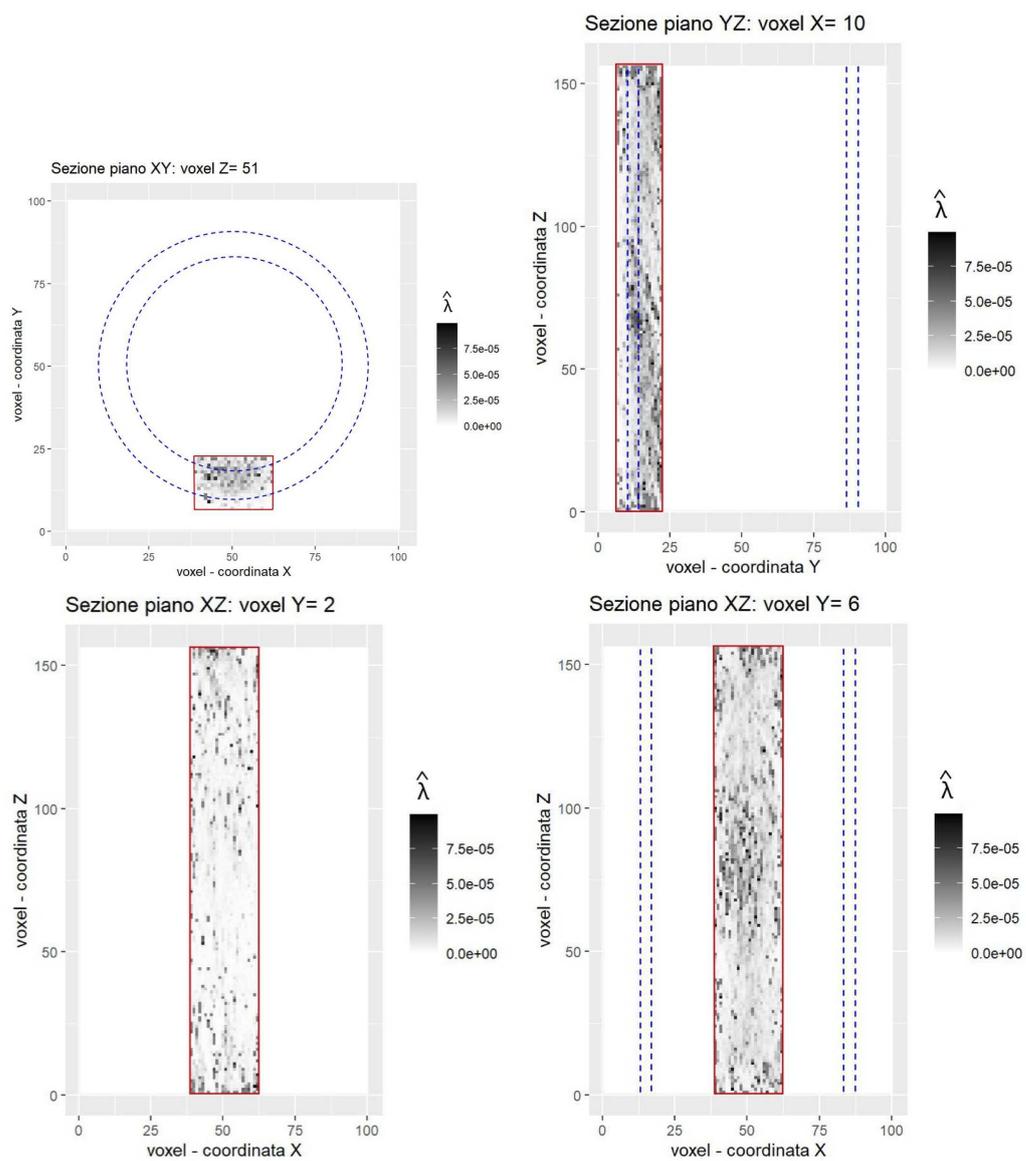


Figura 4.10: Sezioni della porzione di volume ricostruita tramite l'algoritmo SEM. In blu i confini del tubo isolante.

è immediato riconoscere che nel caso di voxel di lato 0.5 cm sia necessaria una quantità molto superiore ai 9113 muoni di cui si ha informazione nell'applicazione alla porzione di volume, perché al diminuire della dimensione dei voxel è sempre meno probabile che ognuno di essi sia attraversato da uno o più muoni.

Il costo computazionale degli algoritmi EM e SEM aumenta notevolmente sia al crescere del numero di muoni che del numero di voxel. L'algoritmo SEM in particolare subisce un forte rallentamento a seconda dell'algoritmo di simulazione impiegato al passo stocastico. Gli algoritmi come il *Gibbs sampler* e la seconda versione del *Metropolis-Hastings* proposti nel paragrafo 3.4.1 comprendono infatti dei passi che devono essere svolti per ogni singolo voxel attraversato da uno specifico muone. Dopo aver effettuato diverse prove con questi algoritmi, si è riscontrato un importante incremento del tempo necessario allo svolgimento di una iterazione dell'algoritmo SEM. Al contrario, la prima versione dell'algoritmo *Metropolis-Hastings* non prevede dei passi iterativi da svolgere per ogni voxel e dunque il costo computazionale si avvicina a quello dell'algoritmo EM.

Ad esempio, in questo caso applicativo, un'iterazione del SEM che utilizza la prima versione dell'algoritmo *Metropolis-Hastings* impiega circa 35 secondi, mentre se si fa uso del *Gibbs sampler* un'iterazione del SEM viene svolta in circa 20 minuti.

In generale, per quanto riguarda la ricostruzione della porzione di volume, nel caso cioè in cui dovrebbe essere ridotto il problema di eterogeneità del materiale di ogni voxel, si è valutato in base alle molteplici prove effettuate per i due algoritmi, che tendenzialmente l'algoritmo EM raggiunge la convergenza in quasi il doppio delle iterazioni rispetto all'algoritmo SEM. Il tempo di un'iterazione del SEM è comunque più elevato di quello impiegato da un'iterazione dell'EM, soprattutto a seconda dell'algoritmo di simulazione utilizzato. Infatti questo tempo ulteriore necessario per la convergenza dell'EM viene raggiunto e superato dall'algoritmo SEM se questo si serve di un algoritmo di simulazione che prevede un procedimento iterativo per ogni voxel.

In conclusione si afferma che non è stato possibile effettuare prove degli algoritmi più particolareggiate di quelle proposte a causa delle limitazioni

tecniche e delle risorse disponibili degli strumenti a disposizione.

4.5 Direzioni future della ricerca

Il metodo proposto in questa tesi con la sua applicazione si può ritenere un primo approccio esplorativo alla risoluzione del problema per cui la distribuzione Gaussiana descritta in (1.7) non approssima bene la totalità degli angoli di *scattering*.

In primo luogo, anche a fronte dei problemi riscontrati ed esposti nel precedente paragrafo, si può innanzitutto prendere in considerazione l'impiego di altri algoritmi di simulazione al passo S. Per esempio si può implementare l'algoritmo *acceptance-rejection* ABC e sue varianti, studiando in maniera preliminare come determinare il parametro di soglia e la funzione distanza migliori ed efficienti per la specifica applicazione.

In secondo luogo, si può applicare il metodo del PoCA descritto in 1.4 che, come descritto in Schultz et al. (2007), dovrebbe condurre a ricostruzioni d'immagini dal contorno meno confuso.

A questo proposito si potrebbe inoltre introdurre nel metodo proposto una correzione per l'errore di misurazione dei rilevatori, sia per poter permettere l'utilizzo dei metodi con misure reali affette da errore, sia per evitare distorsioni dovute alla precisione numerica finita indotta dai calcoli durante l'applicazione pratica degli algoritmi.

Tornando alla specificazione della nuova proposta per il modello che descrive lo *scattering* angolare, innanzitutto si può stimare la costante di proporzionalità K come parametro durante il processo di stima di λ .

Inoltre si potrebbe ipotizzare un diverso legame tra le varianze delle due componenti della mistura, valutando se questo può dipendere anche dal valore di λ .

Un'altra proposta riguarda la scelta dell'algoritmo SEM: si potrebbe far riferimento invece all'algoritmo *Stochastic Approximation* EM (SAEM), che risulta essere un misto tra il SEM e l'EM e consiste nel calcolare la stima del parametro di interesse pesando quella ottenuta con i due diversi algoritmi.

Si potrebbe inoltre ipotizzare un'altra modalità per ben approssimare la vera distribuzione degli angoli di *scattering*; al posto del modello mistura, si

può considerare una distribuzione t -Student, che presenta anch'essa code più pesanti di una Gaussiana.

Il metodo di Schultz et al. (2007) su cui è stata basata la proposta esposta nel capitolo 3, presenta una principale contraddizione. Nella ricostruzione topografica di un volume di materiale omogeneo, si calcola la lunghezza della traiettoria tridimensionale del muone all'interno del volume come distanza tra il punto di entrata del muone nel volume (x_{IN}, y_{IN}, z_{IN}) e il punto (x_p, y_p, z_p) da dove dovrebbe uscire il muone se non venisse deviato.

Invece nel caso di un volume eterogeneo suddiviso in voxel, utilizzato poi per la specificazione dei modelli di riferimento e proposto, la traiettoria tridimensionale si ottiene calcolando la distanza tra (x_{IN}, y_{IN}, z_{IN}) e $(x_{OUT}, y_{OUT}, z_{OUT})$, il punto di uscita del muone dal volume.

Inoltre, si pone tra le assunzioni principali il fatto che gli angoli di *scattering* siano di piccola entità, in quanto alcune quantità che entrano nella specificazione del modello, L_j e T_j per ogni $j = 1, \dots, M$, si calcolano in base alla traiettoria tridimensionale del muone all'interno del voxel, assunta una linea retta tra il punto di entrata e di uscita del muone dal volume.

Per evitare il problema della definizione della traiettoria tridimensionale del muone, si può proporre un nuovo metodo in cui la traiettoria non viene determinata a priori, ma viene costruita strato per strato paralleli a xy , incrociando le informazioni sull'angolo e posizione di entrata e uscita dei muoni dal volume, senza quindi assumere che tale traiettoria sia quasi approssimabile a una retta o subisca un unico evento di *scattering* a livello del punto PoCA.

Conclusioni

L'obiettivo di questa tesi era esplorare le basi statistiche e gli aspetti computazionali della tomografia muonica per *scattering*. Questa metodologia sfrutta la peculiarità dei muoni di penetrare materiali molto densi, perdendo energia molto lentamente. A seconda della densità di *scattering* del materiale attraversato dal muone, quest'ultimo devia la sua traiettoria, con deflessioni maggiori per materiali più densi. La tomografia muonica mira a ricostruire la composizione materiale del volume ignoto tramite immagini tridimensionali, stimando la densità di *scattering* in ciascun voxel, le unità tridimensionali in cui è suddiviso il volume.

In primo luogo è stato descritto il metodo di riferimento di Schultz et al. (2007), che utilizza una distribuzione Gaussiana per gli angoli di *scattering* dei muoni e l'algoritmo EM come strumento operativo per la stima di massima verosimiglianza. Successivamente è stata proposta una metodologia alternativa che prevede un modello mistura di due Gaussiane per la distribuzione degli angoli di *scattering*, in quanto la loro vera distribuzione generatrice presenta code più pesanti di una Gaussiana. La verosimiglianza per la densità di *scattering* viene massimizzata tramite l'algoritmo SEM.

I due metodi sono stati applicati a dati simulati in un esperimento condotto presso il Dipartimento di Fisica Moderna dell'Università di Cantabria, mirato a stimare l'usura delle pareti di tubi isolanti, ricostruendo lo spessore dei singoli materiali che compongono lo strato isolante. Sono stati analizzati i risultati degli algoritmi EM e SEM in due configurazioni: l'intero volume circostante al tubo e una sua piccola porzione. I risultati riflettono l'importanza di definire voxel di dimensioni ridotte per evitare stime fuorvianti della densità di *scattering* dovute alla composizione eterogenea di ogni voxel: ciò porta a stime molto basse della densità di *scattering*, in quanto risultanti

dalla media di valori medio-alti e quasi nulli. A questo proposito, è essenziale disporre di informazioni sullo *scattering* di almeno decine di migliaia di muoni, in modo tale che ogni voxel sia attraversato da più di un muone. Tuttavia, l'aumento del numero di muoni e voxel comporta un incremento significativo del costo computazionale.

A causa delle limitazioni tecniche degli strumenti con cui sono stati implementati gli algoritmi, è risultato infattibile risolvere con successo il problema di eterogeneità. L'algoritmo EM converge in un numero di iterazioni quasi doppio rispetto al SEM, ma il SEM richiede più tempo per iterazione a causa del passo stocastico aggiuntivo, il cui onere computazionale può variare a seconda dell'algoritmo di simulazione impiegato. Entrambi gli algoritmi forniscono risultati ancora grezzi nella ricostruzione del volume, non rappresentando precisamente né i confini del tubo né i materiali che lo compongono; si evidenzia però che il SEM coglie meglio la presenza del tubo nella porzione di volume analizzata, seppur in modo molto vago.

Per la ricerca futura si propone innanzitutto la modifica del modello proposto per la distribuzione degli angoli di *scattering*, nei suoi singoli aspetti o totalmente, per trovare la soluzione che modelli al meglio le code pesanti della vera distribuzione. Inoltre, si possono sperimentare varianti degli algoritmi di simulazione per il passo S del SEM, individuando il metodo più adatto anche dalla prospettiva computazionale. Si possono infine valutare nuove metodologie per la tomografia muonica libere dalle assunzioni restrittive descritte nel metodo di riferimento, ovvero che l'angolo di *scattering* debba essere di piccola entità e che le misure registrate dai rilevatori non siano affette da errore di misurazione.

Bibliografia

- Alvarez, L. W. et al. (1970). «Search for Hidden Chambers in the Pyramids: The structure of the Second Pyramid of Giza is determined by cosmic-ray absorption.» In: *Science* 167.3919, pp. 832–839.
- Biscarat, J. C., G. Celeux e J. Diebolt (1992). «Stochastic Versions of the EM Algorithm». In: *Institut national de recherche en informatique et en automatique*.
- Borozdin, K. N. et al. (2003). «Radiographic imaging with cosmic-ray muons». In: *Nature* 422.6929, p. 277.
- Bouveyron, C. et al. (2019). *Model-Based Clustering and Classification for Data Science: With Applications in R*. 1^a ed. Cambridge University Press.
- Celeux, G., D. Chauveau e J. Diebolt (1992). «On Stochastic Versions of the EM Algorithm». In: *Institut national de recherche en informatique et en automatique* 37.1, pp. 55–57.
- Celeux, G. e J. Diebolt (1985). «The SEM algorithm: a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem». In: *Computational statistics quarterly* 2, pp. 73–82.
- Celeux, G. e G. Govaert (1992). «A classification EM algorithm for clustering and two stochastic versions». In: *Computational Statistics & Data Analysis* 14.3, pp. 315–332.
- Dempster, A. P., N. M. Laird e D. B. Rubin (1977). «Maximum Likelihood from Incomplete Data Via the EM Algorithm». In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 39.1, pp. 1–22.

- Eidelman, S. (1952). «Review of particle physics». In: *Phys. Lett.* B592, p. 1.
- George, E.P. (1955). «Cosmic rays measure overburden of tunnel». In: *Commonwealth Engineer* 455.
- Govaert, G. e M. Nadif (2014). *Co-clustering: models, algorithms and applications*. London, Hoboken: ISTE, Ltd. ; John Wiley & Sons.
- Green, J.A. et al. (2006). «Optimizing the Tracking Efficiency for Cosmic Ray Muon Tomography». In: *2006 IEEE Nuclear Science Symposium Conference Record*. San Diego, CA, USA: IEEE, pp. 285–288.
- Hogan, G. E. et al. (2004). «Detection of High-Z Objects using Multiple Scattering of Cosmic Ray Muons». In: vol. 698. American Institute of Physics, pp. 755–758.
- Johnson, R. A. e D. W. Wichern (2007). *Applied multivariate statistical analysis*. 6th ed. Upper Saddle River, N.J: Pearson Prentice Hall.
- Kuhn, E. e M. Lavielle (2004). «Coupling a stochastic approximation version of EM with an MCMC procedure». In: *ESAIM: Probability and Statistics* 8, pp. 115–131.
- Lenormand, M., F. Jabot e G. Deffuant (2013). «Adaptive approximate Bayesian computation for complex models». In: *Computational Statistics* 28.6, pp. 2777–2796.
- Lynch, G. R. e O. I. Dahl (1990). «Approximations to Multiple Coulomb Scattering». In: *Nuclear instruments and methods in physics research B*.
- McLachlan, G. J., S. X. Lee e S. I. Rathnayake (2019). «Finite Mixture Models». In: *Annual Review of Statistics and Its Application*.
- Morris, C. L. et al. (2008). «Tomographic Imaging with Cosmic Ray Muons». In: *Science & Global Security* 16.1-2, pp. 37–53.
- Nagamine, K. et al. (1995). «Method of probing inner-structure of geophysical substance with the horizontal cosmic-ray muons and possible application to volcanic eruption prediction». In: *Nuclear Instruments and*

- Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 356.2-3, pp. 585–595.
- Picchini, U. e A. Samson (2018). «Coupling stochastic EM and approximate Bayesian computation for parameter inference in state-space models». In: *Computational Statistics* 33.1, pp. 179–212.
- R Core Team (2017). *R: A Language and Environment for Statistical Computing*. Vienna, Austria.
- Robert, C. P. e G. Casella (2004). *Monte Carlo statistical methods*. 2nd ed. Springer texts in statistics. New York: Springer.
- Roentgen, W.C. (1989). In: *Nature* 53, p. 274.
- Rossi, B. (1952). «High-Energy Particles». In: *Englewood Cliffs, NJ: Prentice-Hall*.
- Schultz, L. J. (2003). «Cosmic ray muon radiography». Ph.D. dissertation. Portland, OR: Portland State University.
- Schultz, L. J. et al. (2007). «Statistical Reconstruction for Cosmic Ray Muon Tomography». In: *IEEE Transactions on Image Processing* 16.8, pp. 1985–1993.
- Schultz, L.J. et al. (2004). «Image reconstruction and material Z discrimination via cosmic ray muon radiography». In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 519.3, pp. 687–694. (Visitato il 16/06/2024).
- Siddon, R. L. (1985). «Fast calculation of the exact radiological path for a three-dimensional CT array». In: *Medical Physics* 12.2, pp. 252–255.
- Sisson, S. A., Y. Fan e M. Beaumont (2018). *Handbook of Approximate Bayesian Computation*. First edition. New York: CRC Press.
- Strong, G. C. et al. (2023). «TomOpt: Differential optimisation for task- and constraint-aware design of particle detectors in the context of muon tomography». In: *Machine Learning: Science and Technology*.

Zanghi, H. et al. (2008). «Strategies for Online Inference of Network Mixture». In: *Statistics for Systems Biology Group* Research publication.14.

Appendice A

Approfondimenti algebrici

A.1 Calcolo della matrice Σ

Si sviluppano i calcoli per ottenere la matrice di covarianza Σ della distribuzione congiunta dell'angolo di scattering e del dislocamento (A_x, U_x) , indipendente e identicamente distribuita ad (A_y, U_y) , in funzione della lunghezza delle traiettorie tridimensionale L_j, T_j e della densità di scattering del materiale λ_j , con $j = 1, \dots, M$.

Nello svolgimento si è ricorso all'assunzione che $\mathbb{E}(U_x) = \mathbb{E}(A_x) = 0$, così come $\mathbb{E}(U_{x,j}) = \mathbb{E}(A_{x,j}) = 0$ per $j = 1, \dots, M$, e inoltre che il dislocamento e lo scattering angolare in un voxel sono indipendenti rispetto alle medesime misure negli altri voxel. Inoltre si specifica che l'indice k è sempre riferito ai voxel. Nello specifico, si esprime innanzitutto la varianza dell'angolo di scattering

$$\begin{aligned}\sigma_{a_x}^2 &= Var(A_x) = Var\left(\sum_{j=1}^M A_{x,j}\right) \\ &= \sum_{j=1}^M Var(A_{x,j}) \\ &= \sum_{j=1}^M (L_j \lambda_j p_r^2).\end{aligned}$$

La varianza del dislocamento u_x si ricava nel seguente modo

$$\begin{aligned}
\sigma_{u_x}^2 &= Var(U_x) = Var\left[\sum_{j=1}^M (U_{x,j} + T_j A_{x,j})\right] \\
&= \sum_{j=1}^M Var(U_{x,j} + T_j A_{x,j}) \\
&= \sum_{j=1}^M \left[Var(U_{x,j}) + T_j^2 Var(A_{x,j}) + 2T_j cov(U_{x,j}, A_{x,j})\right] \\
&= \sum_{j=1}^M \left[\frac{L_j^3}{3} \lambda_j p_r^2 + T_j^2 L_j \lambda_j p_r^2 + 2T_j \frac{L_j^2}{2} \lambda_j p_r^2\right] \\
&= \sum_{j=1}^M \left[\left(\frac{L_j^3}{3} + T_j L_j^2 + T_j^2 L_j\right) \lambda_j p_r^2\right].
\end{aligned}$$

Infine, la covarianza tra il dislocamento e lo scattering angolare è calcolata come

$$\begin{aligned}
\sigma_{au_x} &= cov(A_x, U_x) = cov\left(\sum_{j=1}^M A_{x,j}, \sum_{j=1}^M (U_{x,j} + T_j A_{x,j})\right) \\
&= \sum_{j=1}^M cov[A_{x,j}, U_{x,j} + T_j A_{x,j}] \\
&= \sum_{j=1}^M \left[cov(A_{x,j}, U_{x,j}) + T_j Var(A_{x,j})\right] \\
&= \sum_{j=1}^M \left[\frac{L_j^2}{2} \lambda_j p_r^2 + T_j L_j \lambda_j p_r^2\right] \\
&= \sum_{j=1}^M \left[\left(\frac{L_j^2}{2} + T_j L_j\right) \lambda_j p_r^2\right].
\end{aligned}$$

A questo punto è immediato riconoscere la forma della matrice di covarianza in (1.16), in particolare la matrice dei pesi in (1.17) raccoglie i termini di $\sigma_{a_x}^2$, $\sigma_{u_x}^2$ e σ_{au_x} relativi alla lunghezza della traiettoria.

Si nota che le espressioni finali di $\sigma_{a_x}^2$, $\sigma_{u_x}^2$ e σ_{au_x} non presentano riferimenti alla coordinata specifica x , a confermare che i medesimi calcoli svolti per la distribuzione dello scattering lungo y portano a una matrice di covarianza esattamente identica.

A.2 Calcolo della matrice di covarianza $\Sigma_{D_i H_{ij}}$

La distribuzione congiunta (D_i, H_{ij}) è una Gaussiana con matrice di covarianza $\Sigma_{D_i H_{ij}}$, pari a

$$\begin{aligned}
 \Sigma_{D_i H_{ij}} &= \mathbb{E}(D_i H_{ij}^T) - \mathbb{E}(D_i)\mathbb{E}(H_{ij}) \\
 &= \mathbb{E}(D_i H_{ij}^T) \\
 &= \mathbb{E}\left(\begin{bmatrix} A_i \\ U_i \end{bmatrix} \begin{bmatrix} A_{ij} & U_{ij} \end{bmatrix}\right) \\
 &= \mathbb{E}\begin{bmatrix} A_i A_{ij} & A_i U_{ij} \\ U_i A_{ij} & U_i U_{ij} \end{bmatrix} \\
 &= \lambda_j p_{r,i}^2 \begin{bmatrix} L_{ij} & \frac{L_{ij}^2}{2} \\ \frac{L_{ij}^2}{2} + T_j L_{ij} & \frac{L_{ij}^3}{3} + T_j \frac{L_{ij}^2}{2} \end{bmatrix}.
 \end{aligned}$$

I singoli elementi della matrice $\Sigma_{D_i H_{ij}}$ sono ottenuti nel modo descritto in seguito: nei vari passaggi si è considerato il fatto che lo scattering è indipendente nei vari voxel, ovvero sono diversi da zero solo gli elementi con lo stesso indice j , che $\mathbb{E}(A_{ij}) = \mathbb{E}(U_{ij}) = 0$ e infine si è fatto riferimento alle formulazioni in (1.4) e (1.5) relative al relazione tra il dislocamento e l'angolo di scattering.

- $\mathbb{E}(A_i A_{ij}) = \mathbb{E}\left[\left(\sum_{k=1}^M A_{ik}\right) A_{ij}\right] = \mathbb{E}(A_{i1} A_{ij} + \dots + A_{iM} A_{ij})$
 $= \mathbb{E}(A_{ij} A_{ij}) = \text{Var}(A_{ij}) = \lambda_j L_{ij} p_{r,i}^2;$
- $\mathbb{E}(U_i U_{ij}) = \mathbb{E}\left[\left(\sum_{k=1}^M U_{ik} + T_k A_{ik}\right) U_{ij}\right]$
 $= \sum_{k=1}^M [\mathbb{E}(U_{ik} U_{ij}) + T_k \mathbb{E}(A_{ik} U_{ij})] = \mathbb{E}(U_{ij} U_{ij}) + T_j \mathbb{E}(A_{ij} U_{ij})$
 $= \text{Var}(U_{ij}) + T_j \mathbb{E}(A_{ij} U_{ij}) = \lambda_j \frac{L_{ij}^3}{3} p_{r,i}^2 + T_j \lambda_j \frac{L_{ij}^2}{2} p_{r,i}^2;$
- $\mathbb{E}(A_i U_{ij}) = \mathbb{E}\left[\left(\sum_{k=1}^M A_{ik}\right) U_{ij}\right] = \mathbb{E}(A_{ij} U_{ij}) = \lambda_j \frac{L_{ij}^2}{2} p_{r,i}^2;$

$$\begin{aligned}
\bullet \mathbb{E}(U_i A_{ij}) &= \mathbb{E} \left[\left(\sum_{k=1}^M U_{ik} + T_k A_{ik} \right) A_{ij} \right] \\
&= \sum_{k=1}^N [\mathbb{E}(U_{ik} A_{ij}) + T_k \mathbb{E}(A_{ik} A_{ij})] = \mathbb{E}(U_{ij} A_{ij}) + T_j \mathbb{E}(A_{ij} A_{ij}) \\
&= \mathbb{E}(U_{ij} A_{ij}) + T_j \text{Var}(A_j) = \lambda_j \frac{L_{ij}^2}{2} p_{r,i}^2 + T_j \lambda_j L_{ij} p_{r,i}^2.
\end{aligned}$$

Dopodichè si calcolano alcune prodotti matriciali utili per l'espressione dell'uguaglianza in (2.16), facendo riferimento alle equazioni (2.11) e (2.12)

$$\begin{aligned}
\bullet \Sigma_{D_i H_{ij}} \Sigma_{ij}^{-1} &= \lambda_j p_{r,i}^2 \begin{bmatrix} L_{ij} & \frac{L_{ij}^2}{2} \\ \frac{L_{ij}^2}{2} + T_j L_{ij} & \frac{L_{ij}^3}{3} + T_j \frac{L_{ij}^2}{2} \end{bmatrix} \frac{12}{\lambda_j p_{r,i}^2 L_{ij}^4} \begin{bmatrix} \frac{L_{ij}^3}{3} & -\frac{L_{ij}^2}{2} \\ -\frac{L_{ij}^2}{2} & L_{ij} \end{bmatrix} \\
&= \frac{12}{L_{ij}^4} \begin{bmatrix} \frac{L_{ij}^4}{6} + T_j \frac{L_{ij}^4}{3} - \frac{L_{ij}^5}{6} - T_j \frac{L_{ij}^4}{4} & -\frac{L_{ij}^3}{4} - T_j \frac{L_{ij}^3}{2} + \frac{L_{ij}^4}{3} + T_j \frac{L_{ij}^3}{2} \\ \frac{L_{ij}^4}{12} & 0 \end{bmatrix} \\
&= \frac{12}{L_{ij}^4} \begin{bmatrix} \frac{L_{ij}^4}{12} & 0 \\ T_j \frac{L_{ij}^4}{12} & \frac{L_{ij}^4}{12} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ T_j & 1 \end{bmatrix};
\end{aligned}$$

$$\begin{aligned}
\bullet \Sigma_{D_i H_{ij}} \Sigma_{ij}^{-1} \Sigma_{D_i H_{ij}}^T &= \begin{bmatrix} 1 & 0 \\ T_j & 1 \end{bmatrix} \lambda_j p_{r,i}^2 \begin{bmatrix} L_{ij} & \frac{L_{ij}^2}{2} + T_j L_{ij} \\ \frac{L_{ij}^2}{2} & \frac{L_{ij}^3}{3} + T_j \frac{L_{ij}^2}{2} \end{bmatrix} \\
&= \lambda_j p_{r,i}^2 \begin{bmatrix} L_{ij} & \frac{L_{ij}^2}{2} + T_j L_{ij} \\ \frac{L_{ij}^2}{2} + T_j L_{ij} & \frac{L_{ij}^3}{3} + T_j \frac{L_{ij}^2}{2} + T_j^2 L_{ij} \end{bmatrix} = \lambda_j p_{r,i}^2 W_{ij}.
\end{aligned}$$

A.3 Calcolo della matrice $\tilde{\Sigma}_i$

Si sviluppano i calcoli per ottenere la matrice di covarianza $\tilde{\Sigma}_i$ della distribuzione $(D_i | Z_i)$. Nello svolgimento si è ricorso all'assunzione che $\mathbb{E}(A_i) = \mathbb{E}(U_i) = \mathbb{E}(A_{ij}) = \mathbb{E}(U_{ij}) = 0$ per $j = 1, \dots, M$, e inoltre che il dislocamento e lo scattering angolare in un voxel sono indipendenti rispetto alle medesime misure negli altri voxel. Poichè il procedimento è molto simile a quello proposto per il calcolo della matrice Σ in Appendice A.1, si evidenziano solo i passaggi essenziali.

La varianza dell'angolo di scattering a_i è pari a

$$\begin{aligned}\sigma_{a_i}^2 &= Var\left(\sum_{j=1}^M A_{ij}\right) = \sum_{j=1}^M Var(A_{ij}) \\ &= \sum_{j=1}^M (K^{1-z_{ij}} \sigma_{a_{ij}}^2) \\ &= \sum_{j=1}^M (K^{1-z_{ij}} L_{ij} \lambda_j p_{r,i}^2).\end{aligned}$$

La varianza del dislocamento u_i si ricava nel seguente modo

$$\begin{aligned}\sigma_{u_i}^2 &= Var\left[\sum_{j=1}^M (U_{ij} + T_{ij} A_{ij})\right] = \sum_{j=1}^M Var(U_{ij} + T_{ij} A_{ij}) \\ &= \sum_{j=1}^M K^{1-z_{ij}} \left[\sigma_{u_{ij}}^2 + T_{ij}^2 \sigma_{a_{ij}}^2 + 2T_{ij} \sigma_{au_{ij}}\right] \\ &= \sum_{j=1}^M K^{1-z_{ij}} \left[\left(\frac{L_{ij}^3}{3} + T_{ij} L_{ij}^2 + T_{ij}^2 L_{ij}\right) \lambda_j p_{r,i}^2\right].\end{aligned}$$

La covarianza tra il dislocamento e lo scattering angolare è calcolata come

$$\begin{aligned}\sigma_{au_i} &= cov\left(\sum_{j=1}^M A_{ij}, \sum_{j=1}^M (U_{ij} + T_{ij} A_{ij})\right) = \sum_{j=1}^M cov[A_{ij}, U_{ij} + T_{ij} A_{ij}] \\ &= \sum_{j=1}^M K^{1-z_{ij}} \left[\sigma_{au_{ij}} + T_{ij} \sigma_{a_{ij}}^2\right] \\ &= \sum_{j=1}^M K^{1-z_{ij}} \left[\left(\frac{L_{ij}^2}{2} + T_{ij} L_{ij}\right) \lambda_j p_{r,i}^2\right].\end{aligned}$$

A.4 Calcolo della matrice di covarianza $\tilde{\Sigma}_{D_i H_{ij}}$

Per ottenere la matrice di covarianza della distribuzione congiunta $(D_i, H_{ij}|Z_i)$ si sviluppa uno svolgimento molto simile a quello per il calcolo della matrice $\Sigma_{D_i H_{ij}}$ in Appendice A.2, dunque si riportano solo i passaggi fondamentali.

Dunque sia $\hat{z}_{ij}^{(q+1)}$ il valore stimato di z_{ij} all'iterazioni $(q+1)$ dell'algoritmo

$$\begin{aligned}
\tilde{\Sigma}_{D_i H_{ij}} &= \mathbb{E}(D_i H_{ij}^T) - \mathbb{E}(D_i)\mathbb{E}(H_{ij}) = \mathbb{E}(D_i H_{ij}^T) \\
&= \mathbb{E}\left(\begin{bmatrix} A_i \\ U_i \end{bmatrix} \begin{bmatrix} A_{ij} & U_{ij} \end{bmatrix}\right) = \mathbb{E}\begin{bmatrix} A_i A_{ij} & A_i U_{ij} \\ U_i A_{ij} & U_i U_{ij} \end{bmatrix} \\
&= K^{1-\hat{z}_{ij}^{(q+1)}} \lambda_j p_{r,i}^2 \begin{bmatrix} L_{ij} & \frac{L_{ij}^2}{2} \\ \frac{L_{ij}^2}{2} + T_j L_{ij} & \frac{L_{ij}^3}{3} + T_j \frac{L_{ij}^2}{2} \end{bmatrix} \\
&= K^{1-\hat{z}_{ij}^{(q+1)}} \Sigma_{D_i H_{ij}}.
\end{aligned}$$

Rispetto al calcolo della matrice $\tilde{\Sigma}_{D_i H_{ij}}$ bisogna appunto distinguere la componente della mistura che descrive lo scattering in ogni voxel. Si calcola dunque ogni elemento della matrice

- $\mathbb{E}(A_i A_{ij}) = \text{Var}(A_{ij}) = K^{1-\hat{z}_{ij}^{(q+1)}} \lambda_j L_{ij} p_{r,i}^2$;
- $\mathbb{E}(U_i U_{ij}) = \text{Var}(U_{ij}) + T_j \mathbb{E}(A_{ij} U_{ij}) = K^{1-\hat{z}_{ij}^{(q+1)}} \left(\lambda_j \frac{L_{ij}^3}{3} p_{r,i}^2 + T_j \lambda_j \frac{L_{ij}^2}{2} p_{r,i}^2 \right)$;
- $\mathbb{E}(A_i U_{ij}) = \mathbb{E}(A_{ij} U_{ij}) = K^{1-\hat{z}_{ij}^{(q+1)}} \lambda_j \frac{L_{ij}^2}{2} p_{r,i}^2$;
- $\mathbb{E}(U_i A_{ij}) = \mathbb{E}(U_{ij} A_{ij}) + T_j \text{Var}(A_{ij}) = K^{1-\hat{z}_{ij}^{(q+1)}} \left(\lambda_j \frac{L_{ij}^2}{2} p_{r,i}^2 + T_j \lambda_j L_{ij} p_{r,i}^2 \right)$.

Si calcolano inoltre alcune quantità utili per ai passaggi successivi dell'E-step

$$\begin{aligned}
&\bullet \tilde{\Sigma}_{D_i H_{ij}} (K^{1-\hat{z}_{ij}^{(q+1)}} \Sigma_{ij})^{-1} = \\
&= K^{1-\hat{z}_{ij}^{(q+1)}} \lambda_j p_{r,i}^2 \begin{bmatrix} L_{ij} & \frac{L_{ij}^2}{2} \\ \frac{L_{ij}^2}{2} + T_j L_{ij} & \frac{L_{ij}^3}{3} + T_j \frac{L_{ij}^2}{2} \end{bmatrix} \frac{12}{K^{1-\hat{z}_{ij}^{(q+1)}} \lambda_j p_{r,i}^2 L_{ij}^4} \begin{bmatrix} \frac{L_{ij}^3}{3} & -\frac{L_{ij}^2}{2} \\ -\frac{L_{ij}^2}{2} & L_{ij} \end{bmatrix} \\
&= \frac{12}{L_{ij}^4} \begin{bmatrix} \frac{L_{ij}^4}{3} - \frac{L_{ij}^4}{4} & -\frac{L_{ij}^3}{2} + \frac{L_{ij}^3}{2} \\ \frac{L_{ij}^5}{6} + T_j \frac{L_{ij}^4}{3} - \frac{L_{ij}^5}{6} - T_j \frac{L_{ij}^4}{4} & -\frac{L_{ij}^4}{4} - T_j \frac{L_{ij}^3}{2} + \frac{L_{ij}^4}{3} + T_j \frac{L_{ij}^3}{2} \end{bmatrix} \\
&= \frac{12}{L_{ij}^4} \begin{bmatrix} \frac{L_{ij}^4}{12} & 0 \\ T_j \frac{L_{ij}^4}{12} & \frac{L_{ij}^4}{12} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ T_j & 1 \end{bmatrix};
\end{aligned}$$

$$\begin{aligned}
\bullet \tilde{\Sigma}_{D_i H_{ij}} (K^{1-\hat{z}_{ij}^{(q+1)}} \Sigma_{ij})^{-1} \tilde{\Sigma}_{D_i H_{ij}}^T &= \\
&= \begin{bmatrix} 1 & 0 \\ T_j & 1 \end{bmatrix} K^{1-\hat{z}_{ij}^{(q+1)}} \lambda_j p_{r,i}^2 \begin{bmatrix} L_{ij} & \frac{L_{ij}^2}{2} + T_j L_{ij} \\ \frac{L_{ij}^2}{2} & \frac{L_{ij}^3}{3} + T_j \frac{L_{ij}^2}{2} \end{bmatrix} \\
&= K^{1-\hat{z}_{ij}^{(q+1)}} \lambda_j p_{r,i}^2 \begin{bmatrix} L_{ij} & \frac{L_{ij}^2}{2} + T_j L_{ij} \\ \frac{L_{ij}^2}{2} + T_j L_{ij} & \frac{L_{ij}^3}{3} + T_j L_{ij}^2 + T_j^2 L_{ij} \end{bmatrix} \\
&= K^{1-\hat{z}_{ij}^{(q+1)}} \lambda_j p_{r,i}^2 W_{ij}.
\end{aligned}$$

Appendice B

Listato R

B.1 Preparazione dei dati

Calcolo delle quantità necessarie al procedimento di stima della densità di *scattering*.

```
# PREPARAZIONE DEI DATI

n = dim(dati)[1] # numero di muoni
s # lato dei voxel
Nx-1 # numero di voxel di base lungo l'asse x
Ny-1 # numero di voxel di base lungo l'asse y
Nz-1 # numero di voxel di base lungo l'asse z
M = (Nx-1)*(Ny-1)*(Nz-1) # numero di voxel

dati$px1 # X_IN
dati$px2 # X_OUT
dati$py1 # Y_IN
dati$py2 # Y_OUT
dati$pz1 # Z_IN
dati$pz2 # Z_OUT

dati$pvx1 # v_x_IN
dati$pvx2 # v_x_OUT
dati$py1 # v_y_IN
```

```

dati$pv2 # v_y_OUT
dati$pvz1 # v_z_IN
dati$pvz2 # v_z_OUT

dati$ax1 = acos(dati$pvz1/(sqrt(dati$pvz1^2 + dati$pvx1^2))) # theta_x_IN
dati$ax2 = acos(dati$pvz2/(sqrt(dati$pvz2^2 + dati$pvx2^2))) # theta_x_OUT
dati$ay1 = acos(dati$pvz1/(sqrt(dati$pvz1^2 + dati$pvx1^2))) # theta_y_IN
dati$ay2 = acos(dati$pvz2/(sqrt(dati$pvz2^2 + dati$pvx2^2))) # theta_y_OUT
dati$ax = dati$ax2-dati$ax1 # a_x
dati$ay = dati$ay2-dati$ay1 # a_y

dati$xp = dati$px1+(dati$pvx1*((dati$pz2-dati$pz1)/dati$pvz1)) # x_p
dati$yp = dati$py1+(dati$pvx1*((dati$pz2-dati$pz1)/dati$pvz1)) # y_p
dati$zp = dati$pz2 # z_p

lxy = sqrt(1+tan(dati$ax1)^2+tan(dati$ay1)^2)

dati$ux = (dati$px2-dati$xp)*cos(dati$ax1)*lxy*
  (cos(dati$ax+dati$ax1)/cos(dati$ax)) # u_x
dati$uy = (dati$py2-dati$yp)*cos(dati$ay1)*lxy*
  (cos(dati$ay+dati$ay1)/cos(dati$ay)) # u_y

e1 # energia in entrata del muone
e2 # energia in uscita del muone
m = rep(106, nrow(dati)) # massa del muone
p0 = sqrt((dati$e1^2 - m^2)) # momento nominale del muone
p = sqrt((dati$e2^2 - m^2)) # momento del muone
pr2 = (p0/p)^2 # p_r^2

d_x = cbind(dati$ax, dati$ux) # d_x=(a_x,u_x)
d_y = cbind(dati$ay, dati$uy) # d_y=(a_y,u_y)

# ALGORITMO DI SIDDON
# per il calcolo della traiettoria tridimensionale
# si ottengono: L_ij e T_ij, per i=1,...,n e j=1,...,M

```

```

alpha_coord = function(coord_in, coord_out, coord_0, s, ind){
  a1 = (coord_0-coord_in)/(coord_out-coord_in)
  ifelse(ind==1, a1, a1+(ind-1)*s/(coord_out-coord_in))
}
p_coord = function(coord_in, coord_out, alpha){
  coord_in + alpha*(coord_out-coord_in)
}
geometria_siddon = function(xin, xout, yin, yout, zin, zout,
                           x0, y0, z0, Nx, Ny, Nz, s){
  # misura traiettoria tridimensionale
  traj_tot = sqrt((xout-xin)^2+(yout-yin)^2+(zout-zin)^2)

  alpha_x_min = min(alpha_coord(xin, xout, x0, s, 1),
                    alpha_coord(xin, xout, x0, s, Nx))
  alpha_x_max = max(alpha_coord(xin, xout, x0, s, 1),
                    alpha_coord(xin, xout, x0, s, Nx))
  alpha_y_min = min(alpha_coord(yin, yout, y0, s, 1),
                    alpha_coord(yin, yout, y0, s, Ny))
  alpha_y_max = max(alpha_coord(yin, yout, y0, s, 1),
                    alpha_coord(yin, yout, y0, s, Ny))
  alpha_z_min = min(alpha_coord(zin, zout, z0, s, 1),
                    alpha_coord(zin, zout, z0, s, Nz))
  alpha_z_max = max(alpha_coord(zin, zout, z0, s, 1),
                    alpha_coord(zin, zout, z0, s, Nz))

  # ATT: caso in cui xin=xout o yin=yout => alpha corrispondenti non
  definiti
  elementi_a_min = c(0, alpha_z_min)
  elementi_a_max = c(1, alpha_z_max)
  if(!is.na(alpha_x_min)) elementi_a_min = c(elementi_a_min, alpha_x_min)
  if(!is.na(alpha_y_min)) elementi_a_min = c(elementi_a_min, alpha_y_min)
  if(!is.na(alpha_x_max)) elementi_a_max = c(elementi_a_max, alpha_x_max)
  if(!is.na(alpha_y_max)) elementi_a_max = c(elementi_a_max, alpha_y_max)
  alpha_min = max(elementi_a_min)
  alpha_max = min(elementi_a_max)
}

```

```

# indici rette estreme che incrocia la traiettoria
i_min = ifelse(xout>xin,
               ceiling(1+(p_coord(xin, xout, alpha_min)-x0)/s),
               ceiling(1+(p_coord(xin, xout, alpha_max)-x0)/s))
i_max = ifelse(xout>xin,
               floor(1+(p_coord(xin, xout, alpha_max)-x0)/s),
               floor(1+(p_coord(xin, xout, alpha_min)-x0)/s))
j_min = ifelse(yout>yin,
               ceiling(1+(p_coord(yin, yout, alpha_min)-y0)/s),
               ceiling(1+(p_coord(yin, yout, alpha_max)-y0)/s))
j_max = ifelse(yout>yin,
               floor(1+(p_coord(yin, yout, alpha_max)-y0)/s),
               floor(1+(p_coord(yin, yout, alpha_min)-y0)/s))
k_min = ifelse(zout>zin,
               ceiling(1+(p_coord(zin, zout, alpha_min)-z0)/s),
               ceiling(1+(p_coord(zin, zout, alpha_max)-z0)/s))
k_max = ifelse(zout>zin,
               floor(1+(p_coord(zin, zout, alpha_max)-z0)/s),
               floor(1+(p_coord(zin, zout, alpha_min)-z0)/s))

alpha_x_ins = alpha_y_ins = alpha_z_ins = NULL
for(index in i_min:i_max){
  alpha_x_ins = c(alpha_x_ins, alpha_coord(xin, xout, x0, s, index))
}
if(xout<xin) rev(alpha_x_ins)
for(index in j_min:j_max){
  alpha_y_ins = c(alpha_y_ins, alpha_coord(yin, yout, y0, s, index))
}
if(yout<yin) rev(alpha_y_ins)
for(index in k_min:k_max){
  alpha_z_ins = c(alpha_z_ins, alpha_coord(zin, zout, z0, s, index))
}
if(zout<zin) rev(alpha_z_ins)

alpha_xyz = sort(unique(c(
  alpha_min, alpha_x_ins, alpha_y_ins, alpha_z_ins, alpha_max)))

```

```

# ATT: caso in cui la traiettoria secondo una coordinata
# non esce mai dallo stesso voxel OPPURE coord_in=coord_out
condiz_x = condiz_y = FALSE
if(sum(alpha_xyz>=0 & alpha_xyz<=1)!=length(alpha_xyz)){
  alpha_xyz = alpha_xyz[alpha_xyz>=0 & alpha_xyz<=1]
  if(abs(xin-xout)<s & (is.na(alpha_x_min) & is.na(alpha_x_max)) |
      (length(alpha_x_ins)==2 & sum(alpha_x_ins>=0 & alpha_x_ins<=1)==0))
    {
      i = i_max
      condiz_x = TRUE
    }
  else i=NULL
  if(abs(yin-yout)<s & (is.na(alpha_y_min) & is.na(alpha_y_max)) |
      (length(alpha_y_ins)==2 & sum(alpha_y_ins>=0 & alpha_y_ins<=1)==0))
    {
      j = j_max
      condiz_y = TRUE
    }
  else j=NULL
} else{
  i=NULL
  j=NULL
}

# indici dei voxel incrociati dalla traiettoria +
# lunghezza traiettoria nel voxel identificato
k = l = NULL
for (index in 2:length(alpha_xyz)){
  alpha_mid = (alpha_xyz[index]+alpha_xyz[index-1])/2
  i = c(i, floor(1+(p_coord(xin, xout, alpha_mid)-x0)/s))
  j = c(j, floor(1+(p_coord(yin, yout, alpha_mid)-y0)/s))
  k = c(k, floor(1+(p_coord(zin, zout, alpha_mid)-z0)/s))

  l = c(l, traj_tot*(alpha_xyz[index]-alpha_xyz[index-1])) # L_j j
  =1,...,M
}

```

```

}
if(condiz_x) i = i[-1]
if(condiz_y) j = j[-1]

t = NULL
for(ll in 1:length(l)){
  t = c(t, sum(l[-c(1:ll)])) # T_j j=1,...,M
}

list(x_vox=i, y_vox=j, z_vox=k, l_values=l, t_values=t)
}

# Indicizzo i voxel da 1 a M
ordine = cbind(voxel=1:M, expand.grid(ix = 1:(Nx-1), iy = 1:(Ny-1),
                                     iz = (Nz-1):1), n_muon=0)

# Calcolo le matrici B_{ij} e W_{ij}
B = vector("list", n)
W = vector("list", n)

# lista che contiene l'indice dei voxel che attraversa ogni muone
i_vox = vector("list", n)

for(muon in 1:n){
  a = geometria_siddon(xin=dati$px1[muon], xout=dati$px2[muon],
                      yin=dati$py1[muon], yout=dati$py2[muon],
                      zin=dati$pz1[muon], zout=dati$pz2[muon],
                      x0=0, y0=0, z0=0, Nx=Nx, Ny=Ny, Nz=Nz, s=s)
  for(b in 1:length(a$x_vox)){
    i_vox[[muon]] = c(i_vox[[muon]], ordine$voxel[ordine$ix==a$x_vox[b] &
                                                             ordine$iy==a$y_vox[b] &
                                                             ordine$iz==a$z_vox[b]])
  }
  nvoxmuon = length(i_vox[[muon]])

```

```

B[[muon]] = vector("list", nvoxmuon)
W[[muon]] = vector("list", nvoxmuon)

for(vox in 1:nvoxmuon){

  # aggiorno il numero di muoni che passa per tale voxel (serve dopo)
  ordine$n_muon[ i_vox[[muon]][vox] ] = ordine$n_muon[ i_vox[[muon]][vox]
    ] +1

  B[[muon]][[ vox ]] = matrix(c(
    a$_values[vox], (a$_values[vox]^2)/2,
    (a$_values[vox]^2)/2, (a$_values[vox]^3)/3),
    ncol=2, byrow=T)

  W[[muon]][[ vox ]] = matrix(c(
    a$_values[vox], ((a$_values[vox]^2)/2)+a$_values[vox]*a$t_values[
      vox],
    ((a$_values[vox]^2)/2)+ a$_values[vox]*a$t_values[vox],
    ((a$_values[vox]^3)/3)+(a$_values[vox]^2)*a$t_values[vox]+
      a$_values[vox]*(a$t_values[vox])^2),
    ncol=2, byrow=T)
}
if(muon%%10==0) print(muon)
}

```

B.2 Algoritmo EM

```

# ALGORITMO EM

require(mvtnorm)

em.fun = function(lambda_start=rep(0.01,M), d_x, d_y, eps=1e-5, max_iter
  =3000){
  err = 1

```

```

iter = 0
lambda_est = lambda_start
lambda_new = rep(0,M)
lverosD = NULL

while(iter<max_iter){
  iter = iter+1
  G_tot = rep(0, M)
  lv_D = 0 # log-verosimiglianza

  for(muon in 1:n){
    # PASSO E:
    # calcolo inverso matrice di covarianza dei dati d
    wl = W[[muon]]
    sum_wl = 0
    sigma_d_i_inv = matrix(NA, nrow=2, ncol=2)
    for(j in 1:length(i_vox[[muon]])){
      sum_wl = sum_wl + (wl[[j]] * lambda_est[ i_vox[[muon]][j] ])
    }
    sigma_d_i = pr2[muon]*sum_wl
    sigma_d_i_inv = solve(sigma_d_i)

    lv_D = lv_D +
      dmvnorm(d_x[muon,], mean=c(0,0), sigma=sigma_d_i, log=T) +
      dmvnorm(d_y[muon,], mean=c(0,0), sigma=sigma_d_i, log=T)

    for(j in 1:length(i_vox[[muon]])){
      G_tot[ i_vox[[muon]][j] ] = G_tot[ i_vox[[muon]][j] ] +
        ( ( 2*lambda_est[ i_vox[[muon]][j] ] +
          (lambda_est[ i_vox[[muon]][j] ]^2)*pr2[muon]*
          (t(d_x[muon,]) %*% sigma_d_i_inv %*% W[[muon]][[j]] %*%
            sigma_d_i_inv %*% d_x[muon,] -
            sum(diag(sigma_d_i_inv %*% W[[muon]][[j]]))) )
        +
          ( 2*lambda_est[ i_vox[[muon]][j] ] +
            (lambda_est[ i_vox[[muon]][j] ]^2)*pr2[muon]*

```

```

        (t(d_y[muon,]) %**% sigma_d_i_inv %**% W[[muon]][[j]] %*
         %
         sigma_d_i_inv %**% d_y[muon,] -
         sum(diag(sigma_d_i_inv %**% W[[muon]][[j]]))) ) ) /
         2)
    }
}
lverosD = c(lverosD, lv_D)

# PASSO M:
# calcolo formula iterativa per ogni lambda_j
for(j in 1:M){
  lambda_new[j] = (2*ordine$n_muon[j])^(-1) * G_tot[j]
}
err = max(abs(lambda_new-lambda_est), na.rm=T)
lambda_est = lambda_new
lambda_est[is.na(lambda_est)] = 0

# criterio di convergenza
if (iter>4) {
  cond = diff(lverosD[iter:(iter-3)])
  l03 = 0.0003*mean(abs(lverosD[iter:(iter-3)]))
  if (sum(abs(cond)<l005) == 3) break
}

if(iter%%10==0)
  print(iter)
}
if(iter==max_iter) warning("Maximum number of iterations reached")
list(lambda_est=lambda_est, n_iter=iter, lverosD=lverosD)
}

```

B.3 Algoritmo SEM e algoritmi di simulazione per il passo S

```
# ALGORITMO SEM

require(mvtnorm)

# funzione per calcolare la matrice di covarianza di d_i
sigma_tilde_i = function(imuon, indici0, lambda_val, K){
  wij = W[[imuon]]
  sum_wij = 0

  jvox1 = which(!(i_vox[[imuon]] %in% indici0))
  jvox0 = which(i_vox[[imuon]] %in% indici0)

  for(jvox in jvox1){
    sum_wij = sum_wij + (wij[[jvox]] * lambda_val[ i_vox[[imuon]][jvox] ])
  }
  for(jvox in jvox0){
    sum_wij = sum_wij + (K * wij[[jvox]] * lambda_val[ i_vox[[imuon]][jvox]
    ])
  }
  pr2[imuon]*sum_wij
}

# ALGORITMI DI SIMULAZIONE per il passo S:

# GIBBS SAMPLING
gs.sim = function(i_muon, ddd, ind_vox0, tau, K, lambda, R=5){
  for (r in 1:R){
    order_jvox = if(length(i_vox[[i_muon]])!=1){
      sample(x=i_vox[[i_muon]], size=length(i_vox[[i_muon]]), replace=F)
    }else{ i_vox[[i_muon]]}
  }
}
```

```

for (j_vox in order_jvox){
  if(!(j_vox %in% ind_vox0)){ # zij=1
    sigma_i1 = sigma_tilde_i(imuon=i_muon, indici0=ind_vox0, lambda_
      val=lambda, K=K)
    num = dmvnorm(ddd, mean=c(0,0), sigma=sigma_i1) *
      (tau^(length(i_vox[[i_muon]])-sum(i_vox[[i_muon]] %in% ind_vox0)
        ))*
      ((1-tau)^(sum(i_vox[[i_muon]] %in% ind_vox0)))

    ind_vox000 = c(ind_vox0, j_vox)
    sigma_i0 = sigma_tilde_i(imuon=i_muon, indici0=ind_vox000, lambda_
      val=lambda, K=K)
    denom = num + ( dmvnorm(ddd, mean=c(0,0), sigma=sigma_i0)*
      (tau^(length(i_vox[[i_muon]])-sum(i_vox[[i_muon]
        ] %in% ind_vox000))))*
      ((1-tau)^(sum(i_vox[[i_muon]] %in% ind_vox000)))
      )
  }else{ # zij=0
    sigma_i0 = sigma_tilde_i(imuon=i_muon, indici0=ind_vox0, lambda_
      val=lambda, K=K)
    denom_parz = dmvnorm(ddd, mean=c(0,0), sigma=sigma_i0)*
      (tau^(length(i_vox[[i_muon]])-sum(i_vox[[i_muon]] %in% ind_vox0)
        ))*
      ((1-tau)^(sum(i_vox[[i_muon]] %in% ind_vox0)))

    ind_vox011 = setdiff(ind_vox0, j_vox)
    sigma_i1 = sigma_tilde_i(imuon=i_muon, indici0=ind_vox011, lambda_
      val=lambda, K=K)
    num = dmvnorm(ddd, mean=c(0,0), sigma=sigma_i1) *
      (tau^(length(i_vox[[i_muon]])-sum(i_vox[[i_muon]] %in% ind_
        vox011))))*
      ((1-tau)^(sum(i_vox[[i_muon]] %in% ind_vox011)))
    denom = denom_parz + num
  }

  prob_zij1 = num / denom
}

```

```

zij_sim = rbinom(n=1, size=1, prob=prob_zij1)

if(zij_sim==1 && (j_vox %in% ind_vox0)){ # prima era zij=0, ora e'
  zij=1
  ind_vox0 = setdiff(ind_vox0, j_vox)
}
if(zij_sim==0 && !(j_vox %in% ind_vox0)){ # prima era zij=1, ora e'
  zij=0
  ind_vox0 = c(ind_vox0, j_vox)
}
}
}
ind_vox0
}

# METROPOLIS-HASTINGS 1
mh.is.sim = function(i_muon, ddd, ind_vox0, tau, K, lambda, R=5){
  ind0_sim = ind_vox0
  for (r in 1:R){
    ind0_star = sample(1:length(i_vox[[muon]]), size = round(length(i_vox
      [[muon]])*(1-tau)),
      replace = FALSE)

    sigma_i_old = sigma_tilde_i(imuon=i_muon, indici0=ind0_sim, lambda_val
      =lambda, K=K)
    d_old = dmvnorm(ddd, mean=c(0,0), sigma=sigma_i_old)

    sigma_i_star = sigma_tilde_i(imuon=i_muon, indici0=ind0_star, lambda_
      val=lambda, K=K)
    d_star = dmvnorm(ddd, mean=c(0,0), sigma=sigma_i_star)

    alpha = min(1, d_star/d_old)
    if (runif(1) < alpha){
      ind0_sim = ind0_star
    }
  }
}

```

```

ind0_sim
}

# METROPOLIS-HASTINGS 2
mh.is.j.sim = function(i_muon, ddd, ind_vox0, tau, K, lambda, R=5){
  ind0_sim = ind0_star = ind_vox0
  for (r in 1:R){
    for(j_vox in i_vox[[i_muon]]){
      zij_sim = rbinom(n=1, size=1, prob=tau)
      if(zij_sim==1 && (j_vox %in% ind0_sim)){ # prima era zij=0, ora e'
        zij=1
        ind0_star = setdiff(ind0_sim, j_vox)
      }
      if(zij_sim==0 && !(j_vox %in% ind0_sim)){ # prima era zij=1, ora e'
        zij=0
        ind0_star = c(ind0_sim, j_vox)
      }

      sigma_i_old = sigma_tilde_i(imuon=i_muon, indici0=ind0_sim, lambda_
        val=lambda, K=K)
      d_old = dmvnorm(ddd, mean=c(0,0), sigma=sigma_i_old)

      sigma_i_star = sigma_tilde_i(imuon=i_muon, indici0=ind0_star, lambda
        _val=lambda, K=K)
      d_star = dmvnorm(ddd, mean=c(0,0), sigma=sigma_i_star)

      n0 = sum(i_vox[[i_muon]] %in% ind0_star)

      alpha_j = min(1, (d_star* (tau^(length(i_vox[[i_muon]])-n0))*((1-tau
        )^n0)) /
        (d_old* (tau^zij_sim)*((1-tau)^zij_sim)) )

      if(runif(1) < alpha_j){
        ind0_sim = ind0_star
      }
    }
  }
}

```

```

}
  ind0_sim
}

# inizializzo casualmente i voxel in cui lo scattering si distribuisce
  secondo
# la seconda componente della mistura
ind0_start_zx = ind0_start_zy = vector("list", n)
for(ind0 in 1:n){
  set.seed(1+ind0)
  ind0_start_zx[[ind0]] = sample(1:M, size = round(M*(1-tau)), replace = F
  )
  set.seed(12*ind0)
  ind0_start_zy[[ind0]] = sample(1:M, size = round(M*(1-tau)), replace = F
  )
}

sem.fun = function(lambda_start=rep(0.01,M), d_x, d_y,
  indice0_zx=ind0_start_zx, indice0_zy=ind0_start_zy,
  tau=0.98, K=20, eps=1e-5, max_iter=3000, simulation="mh
  "){
  err = 1
  iter = 0
  lambda_est = lambda_start
  lambda_new = rep(0,M)
  lverosD = NULL # log-verosimiglianza

  while(iter<max_iter){
    iter = iter+1

    G_tot = rep(0, M)
    lv_D = 0

    for(muon in 1:n){
      # PASSO S:

```

```

if(simulation=="gibbs"){
  indice0_zx[[muon]] = gs.sim(i_muon=muon, ddd=d_x[muon,],
                              ind_vox0=indice0_zx[[muon]],
                              tau=tau, K=K, lambda=lambda_est)
  indice0_zy[[muon]] = gs.sim(i_muon=muon, ddd=d_y[muon,],
                              ind_vox0=indice0_zy[[muon]],
                              tau=tau, K=K, lambda=lambda_est)
}
if(simulation=="mh"){
  indice0_zx[[muon]] = mh.is.sim(i_muon=muon, ddd=d_x[muon,],
                                 ind_vox0=indice0_zx[[muon]],
                                 tau=tau, K=K, lambda=lambda_est)
  indice0_zy[[muon]] = mh.is.sim(i_muon=muon, ddd=d_y[muon,],
                                 ind_vox0=indice0_zy[[muon]],
                                 tau=tau, K=K, lambda=lambda_est)
}
if(simulation=="mhj"){
  indice0_zx[[muon]] = mh.is.j.sim(i_muon=muon, ddd=d_x[muon,],
                                   ind_vox0=indice0_zx[[muon]],
                                   tau=tau, K=K, lambda=lambda_est)
  indice0_zy[[muon]] = mh.is.j.sim(i_muon=muon, ddd=d_y[muon,],
                                   ind_vox0=indice0_zy[[muon]],
                                   tau=tau, K=K, lambda=lambda_est)
}
if(simulation=="abc"){
  z_hat_i_x = abc.is.sim(d_x[muon,], z_x_hat[muon,], tau=tau, K=K,
                        lambda=lambda_est)
  z_hat_i_y = abc.is.sim(d_y[muon,], z_y_hat[muon,], tau=tau, K=K,
                        lambda=lambda_est)
}

# PASSO E:
# calcolo inverso matrice di covarianza dei dati d
sigma_d_i_x = sigma_tilde_i(muon, indici0=indice0_zx[[muon]],
                            lambda_val=lambda_est, K=K)
sigma_d_i_y = sigma_tilde_i(muon, indici0=indice0_zy[[muon]],

```

```

                                lambda_val=lambda_est, K=K)
sigma_d_i_x_inv = solve(sigma_d_i_x)
sigma_d_i_y_inv = solve(sigma_d_i_y)

lv_D = lv_D +
  dmvnorm(d_x[muon,], mean=c(0,0), sigma=sigma_d_i_x, log=T) +
  dmvnorm(d_y[muon,], mean=c(0,0), sigma=sigma_d_i_y, log=T)

for(j in 1:length(i_vox[[muon]])){
  G_tot[ i_vox[[muon]][j] ] = G_tot[ i_vox[[muon]][j] ] +
    ( ( 2*lambda_est[ i_vox[[muon]][j] ] +
      (lambda_est[ i_vox[[muon]][j] ]^2)*pr2[muon]*
      K^(i_vox[[muon]][j] %in% indice0_zx[[muon]])*
      (t(d_x[muon,]) %**% sigma_d_i_x_inv %**% W[[muon]][[j]]) %*
        %
        sigma_d_i_x_inv %**% d_x[muon,] -
        sum(diag(sigma_d_i_x_inv %**% W[[muon]][[j]]))) ) )
    +
    ( 2*lambda_est[ i_vox[[muon]][j] ] +
      (lambda_est[ i_vox[[muon]][j] ]^2)*pr2[muon]*
      K^(i_vox[[muon]][j] %in% indice0_zy[[muon]])*
      (t(d_y[muon,]) %**% sigma_d_i_y_inv %**% W[[muon]][[j]])
        %**%
        sigma_d_i_y_inv %**% d_y[muon,] -
        sum(diag(sigma_d_i_y_inv %**% W[[muon]][[j]]))) ) ) /
      2)
}
}
lverosD = c(lverosD, lv_D)

# PASSO M:
# calcolo formula iterativa per ogni lambda_j
for(j in 1:M){
  lambda_new[j] = (2*ordine$n_muon[j])^(-1) * G_tot[j]
}
err = max(abs(lambda_new-lambda_est), na.rm=T)

```

```
lambda_est = lambda_new
lambda_est[is.na(lambda_est)] = 0

if (iter>4) {
  cond = diff(lverosD[iter:(iter-3)])
  l03 = 0.0003*mean(abs(lverosD[iter:(iter-3)]))
  if(sum(abs(cond)<l001) == 3) break
}

if(iter%%10==0)
  print(iter)
}
if(iter==max_iter) warning("Maximum number of iterations reached")
list(lambda_est=lambda_est, n_iter=iter, lverosD=lverosD)
}
```
