

POLITECNICO DI TORINO

I Facoltà di Ingegneria

Corso di Laurea in Matematica per le Scienze dell'Ingegneria

Prova finale

**Analisi e risoluzione numerica di
un problema inverso di
conduttività elettrica**

Stima del piano che separa due regioni di un solido con proprietà
elettriche diverse



Relatori:

prof. Giovanni MONEGATO

ing. Andrea BORSIC

Candidato:

Matteo ICARDI

Marzo 2006

Sommario

In questa tesi, sviluppata presso la SC-AIP S.A.S., un'azienda dell'Incubatore Imprese Innovative del Politecnico di Torino, viene affrontato il problema di ricostruire il fronte piano di separazione di due regioni di un solido con conduttività diverse. L'obiettivo finale è quello di ricostruire la conduttività del solido a partire dalle misure del potenziale elettrico che si instaura agli elettrodi in seguito a determinate correnti iniettate. Si tratta di un problema inverso non-lineare mal-posto quindi, per prima cosa verrà modellizzato, discretizzato e risolto il problema diretto ad esso associato. Poi, dopo aver affrontato le problematiche principali legate ai problemi inversi e alla loro regolarizzazione, sarà possibile, mediante un procedimento iterativo, trovare la giusta posizione del piano a partire da valori di potenziale e corrente noti.

Per risolvere il problema diretto si utilizzerà un modello matematico chiamato *complete electrode model*. Questo modello e tutti i metodi usati per invertire il problema sono frequentemente citati nella letteratura sulla ricostruzione di immagini attraverso la tomografia ad impedenza elettrica, una tecnica usata soprattutto in campo medico, industriale e geologico. Il modello che vedremo darà luogo ad un'equazione alle derivate parziali ellittica con condizioni al bordo di tipo misto che consente di risolvere il problema diretto.

Tutto questo sarà analizzato da un punto di vista matematico e poi risolto tramite l'utilizzo dell'ambiente di sviluppo MATLAB e di funzioni sviluppate in parte dal candidato ed in parte dalla SC-AIP.

Indice

Sommario	I
1 Introduzione	5
2 Problema diretto	9
2.1 Introduzione	9
2.2 Modellizzazione	10
2.2.1 Valori al bordo	12
2.3 Esistenza e unicità	15
2.4 Linearità di $\mathbf{I} \rightarrow \mathbf{U}$	20
2.5 Risoluzione numerica	21
2.5.1 Caso semplificato	23
2.6 Conclusioni	24
3 Problema inverso	25
3.1 Introduzione	25
3.2 Problemi inversi e minimi quadrati	26
3.3 Regolarizzazione tramite SVD	29
3.3.1 Altre tecniche di regolarizzazione	32
3.4 Inversione del problema	33
3.5 Conclusione	35
4 Sintesi del lavoro svolto e test eseguiti	37
4.1 Introduzione	37

4.2	Assemblaggio della matrice degli EF	38
4.2.1	Verifica del risultato	39
4.3	Calcolo dello Jacobiano	42
4.3.1	Verifica del risultato	44
4.4	Inversione del problema	45
4.5	Test conclusivo	49
5	Conclusioni e possibili sviluppi	53
A	Listati dei programmi	57
A.1	inters.m	58
A.2	FEMmatrix.m	58
A.3	measures.m	61
A.4	deriv.m	61
A.5	dFEMmatrix.m	66
A.6	jacobian.m	67
A.7	inverse_solve.m	67
A.8	disegna.m	68
	Bibliografia	71

Capitolo 1

Introduzione

Questa tesi di laurea nasce da una collaborazione con la società SC-AIP S.A.S. (*Scientific Computing & Applied Inverse Problems*) presso l'Incubatore Imprese Innovative del Politecnico di Torino. L'obiettivo della collaborazione era quello di elaborare alcune funzioni MATLAB che, insieme ad altri programmi già sviluppati dall'ing. Borsic, permettessero di risolvere il problema qui di seguito descritto.

Supponiamo di aver un solido di un materiale che può assumere, al suo interno, proprietà elettriche diverse. Nel nostro caso la proprietà che ci interessa è la conduttività elettrica e supponiamo inoltre che questa sia costante al di sopra e al di sotto di un piano che sarà quindi il fronte di separazione tra queste due parti del solido e una discontinuità per la funzione che rappresenta la conduttività. Siamo interessati ad ottenere una stima della posizione di questo piano utilizzando delle misure sperimentali sull'oggetto. L'idea è quella di applicare degli elettrodi alla superficie dell'oggetto sotto misura, di utilizzare coppie di elettrodi per iniettare correnti e di misurare i potenziali elettrici risultanti agli elettrodi stessi. Da questi dati dovremo poi dedurre la posizione del piano. Come vedremo, si tratta quindi di risolvere un problema inverso conoscendo i valori al bordo.

Questo problema si inserisce all'interno dei recenti studi sulla ricostruzione di immagini attraverso la tomografia elettrica (*Electrical Impedance Tomography*, EIT) che è la tecnica con cui si ricavano informazioni sulle proprietà elettriche di un corpo a partire da misurazioni non-invasive al bordo.

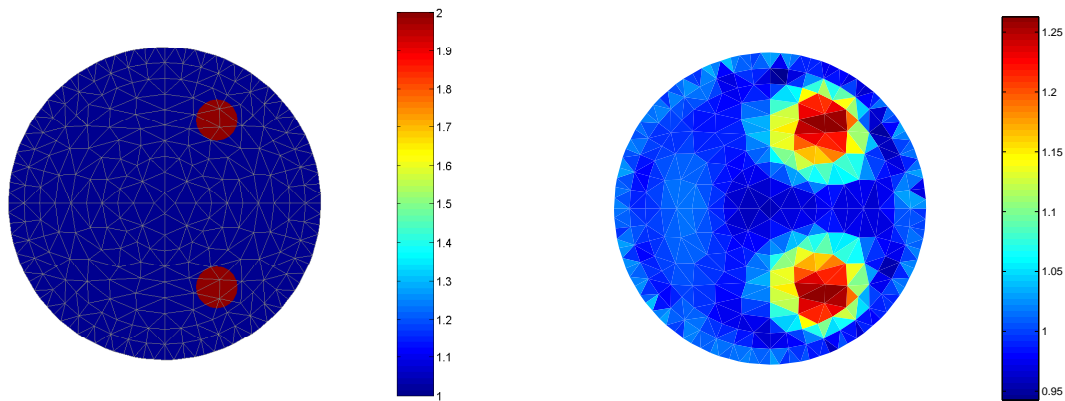


Figura 1.1. Esempio di ricostruzione di immagini tramite EIT che mostra le difficoltà nel ricostruire variazioni brusche di conduttività. A sinistra possiamo vedere il profilo reale di un oggetto che contiene al suo interno due regioni con conduttività diversa. A destra la sua ricostruzione.

La ricostruzione di immagini attraverso la tomografia elettrica risulta infatti difficile per oggetti ad elevata frequenza spaziale, cioè con variazioni elevate di conduttività interna. In figura 1.1 possiamo vedere come la tecnica di ricostruzione classica non è adatta per questo tipo di oggetti perché tende a smorzare queste variazioni e non è possibile quindi individuare in maniera precisa il contorno delle varie regioni a conduttività diversa. Per questo tipo di immagini è necessario utilizzare tecniche di ricostruzione diverse ([1]) che sfruttino delle conoscenze a priori. Nel nostro caso particolare, in cui l'oggetto da ricostruire è diviso in due regioni da un piano, non otterremo buoni risultati se cercassimo di stimare la conduttività su tutto il corpo punto per punto. Il nostro obiettivo è quindi di ricostruire l'immagine sfruttando le informazioni a priori cioè stimando solo la posizione del piano.

Durante questa collaborazione, cominciata nell'Ottobre 2004, trovando particolarmente stimolante ed istruttivo l'argomento in questione, ho deciso di analizzare più in dettaglio gli strumenti matematici alla base di questo lavoro. Quindi questa tesi di laurea, oltre a riportare i risultati di questa collaborazione, vuole anche analizzare e illustrare i vari metodi e modelli matematici utilizzati. Tuttavia alcuni di questi argomenti sono contenuto specifico di insegnamenti proposti dal corso di laurea specialistica di Ingegneria Matematica quindi questi ultimi non saranno approfonditi allo stesso modo degli altri argomenti.

Come abbiamo visto, il problema da risolvere è molto specifico ma per poterlo affrontare e comprendere bene è stato necessario porsi in una situazione più generale. Nei primi capitoli di questa tesi si tratteranno quindi temi di carattere più generale e si analizzeranno i metodi matematici utilizzati per poi applicarli al caso in esame.

Nel prossimo capitolo dedurremo e analizzeremo un modello matematico per risolvere il nostro problema, nel terzo verranno trattati tutti gli argomenti riguardanti l'inversione del problema. Nel capitolo 4 invece ci concentreremo sul nostro problema specifico e si darà una relazione di tutto il lavoro svolto durante la collaborazione con l'ing. Borsic sotto la supervisione dei relatori. Infine nel quinto capitolo si tornerà ad una visione più generale per trarre le conclusioni del lavoro e dare possibili ulteriori sviluppi e applicazioni dei software sviluppati.

Capitolo 2

Problema diretto

2.1 Introduzione

Tralasciando le semplificazioni a cui abbiamo accennato nel capitolo 1 e su cui ci soffermeremo più avanti, il nostro problema potrebbe essere così illustrato: *in un corpo sono iniettate delle correnti continue attraverso n elettrodi posti sulla superficie dello stesso. Dalla misura del potenziale elettrico che si instaura agli elettrodi vogliamo ricavare informazioni sulla conduttività del corpo.*

Per cercare di capire il tipo di problema che dobbiamo affrontare è necessario fare una premessa sui modelli matematici.

Nella modellizzazione matematica di un problema come il nostro si possono distinguere alcuni elementi:

- le *variabili indipendenti* che generalmente sono il tempo $t \in [0, T] \subseteq \mathbb{R}$ e lo spazio $\mathbf{x} = \{x, y, z\} \in \Omega \subseteq \mathbb{R}^3$;
- le *funzioni di stato* che sono le grandezze che di solito vogliamo studiare e che indichiamo con $\mathbf{u} = \mathbf{u}(t, \mathbf{x}) : [0, T] \times \Omega \rightarrow \mathbb{R}^n$ dove $\mathbf{u} = \{u_1, u_2, \dots, u_n\}$;
- i *parametri*, cioè le grandezze caratteristiche del singolo sistema fisico;
- i *valori al bordo*, cioè il valore della funzione \mathbf{u} o delle sue derivate per $t = 0$, $t = T$ e per $\mathbf{x} \in \partial\Omega$. Questi dati sono necessari quando, per trovare \mathbf{u} , dobbiamo risolvere un'equazione differenziale.

I problemi in cui, dati i parametri e i valori al bordo, bisogna trovare la funzione di stato su tutto o su parte del dominio sono i problemi più classici e più facili da affrontare. Questi sono chiamati *problemi diretti*. Tuttavia nel nostro caso non possiamo ricondurci a questo tipo di problemi. Sempre più spesso infatti, nella fisica e nelle scienze moderne, ci troviamo di fronte a problemi di natura diversa.

Nel nostro problema la conduttività, che è facilmente identificabile come un parametro, non è nota. Siamo quindi in presenza di un cosiddetto *problema inverso ai valori al bordo* in cui dati i valori al bordo dobbiamo trovare il valore del parametro.

Detto questo, possiamo ora analizzare le altre caratteristiche del nostro problema. Siamo infatti a conoscenza di alcune *informazioni a priori*, cioè sappiamo che la conduttività è costante all'interno di due regioni separate da un piano. A partire da queste informazioni, per poter conoscere la conduttività su tutto il corpo, basterà stimare la posizione di questo piano. Il nostro problema quindi consiste nello stimare il bordo (*boundary estimation*) delle due regioni a conduttività diverse. Vedremo più avanti come abbiamo rappresentato il piano e quali sono quindi i parametri da stimare.

Si può capire facilmente che, per risolvere qualunque problema inverso, è fondamentale conoscere e poter risolvere il relativo problema diretto. Sarà quindi necessario, prima di affrontare il problema inverso, dedurre un buon modello per il nostro sistema e analizzare e risolvere il problema diretto.

In 2.2 ci occuperemo di ottenere una formulazione matematica precisa del problema diretto per poter poi dedurre un modello generale. Quindi in 2.3 dimostreremo l'esistenza e unicità della soluzione del problema diretto. Nella sezione 2.5 vedremo come è possibile discretizzare e risolvere il problema diretto con il metodo degli elementi finiti e illustreremo come abbiamo utilizzato questo metodo.

2.2 Modellizzazione

In base a quanto detto nel paragrafo precedente risolvere il problema diretto nel nostro caso vuol dire trovare il valore del potenziale elettrico ai capi degli elettrodi

conoscendo la conduttività su tutto il dominio e le correnti iniettate. In questo paragrafo vedremo come sia possibile costruire un modello matematico del nostro sistema fisico e nel capitolo 3 potremo anche verificare la bontà del nostro modello perché si possono costruire casi particolari in cui risulta facile confrontare la predizione fatta dal modello con i risultati teorici.

Per prima cosa occorre definire le grandezze coinvolte. Sulla superficie $\partial\Omega$ del nostro corpo $\Omega \subset \mathbb{R}^3$ di conduttività $\sigma = \sigma(x,y,z)$ dobbiamo iniettare delle correnti attraverso n elettrodi posti sulla superficie del solido e calcolare i potenziali agli stessi elettrodi. Identifichiamo gli elettrodi con la porzione di superficie a cui sono affacciati. Li indicheremo quindi come $e_i \subset \partial\Omega$, $1 \leq i \leq n$. Avremo quindi un vettore $\mathbf{I} = (I_1, I_2, \dots, I_n)$, dove I_i rappresenta la corrente iniettata al i -esimo elettrodo, e un vettore $\mathbf{U} = (U_1, U_2, \dots, U_n)$, dove U_i rappresenta il valore del potenziale nel i -esimo elettrodo. \mathbf{U} sarà identificato a meno di una costante che possiamo eliminare facilmente, per esempio scegliendo a priori un nodo di terra a potenziale nullo.

É importante osservare che il nostro problema risulta *statico*, cioè tutte le grandezze considerate non dipendono dal tempo, in quanto le correnti iniettate sono correnti continue e le misure che facciamo non sono legate agli effetti transitori ma sono effettuate dopo il raggiungimento dell'equilibrio. Nella tomografia elettrica vengono spesso considerate delle correnti sinusoidali a frequenza fissa. In questi casi basta usare al posto delle grandezze reali, le relative grandezze complesse.

Sappiamo da dati sperimentali che la relazione che lega \mathbf{U} e \mathbf{I} è di tipo *lineare*. Possiamo quindi scriverla come

$$\mathbf{U} = R\mathbf{I} \tag{2.1}$$

dove $R \in \mathbb{R}^{n \times n}$ è la matrice simmetrica delle resistenze che dipende quindi da σ . Torneremo più avanti sull'importanza e sulle conseguenze della linearità del problema diretto.

Per definire il nostro modello potremmo partire dalle equazioni di Maxwell ma per semplicità possiamo trascurare gli effetti del campo magnetico perché lavoriamo con correnti continue. Possiamo supporre inoltre che il campo elettrico sia

conservativo¹. Quindi possiamo esprimere il campo elettrico come

$$\mathbf{E} = \nabla u \quad (2.2)$$

dove u è il potenziale scalare del campo elettrico. La densità di corrente, usando la versione continua della legge di Ohm, risulta

$$\mathbf{J} = \sigma \nabla u \quad (2.3)$$

Inoltre, in assenza di correnti interne, possiamo applicare la versione continua della legge di Kirchoff e otteniamo la seguente equazione per il potenziale u :

$$\nabla \cdot \sigma(\mathbf{x}) \nabla u = 0 \quad (2.4)$$

L'equazione da risolvere è quindi un *equazione alle derivate parziali di tipo ellittico*. Dato σ , per poter risolvere questo tipo di equazione è necessario aggiungere delle condizioni al contorno.

2.2.1 Valori al bordo

Sono possibili diverse scelte per definire i valori al bordo cioè nel nostro caso le correnti iniettate nel solido. La relazione che dà luogo ad una condizione di Neumann su $\partial\Omega$ la ricaviamo immediatamente dalla definizione di potenziale elettrico

$$\sigma \frac{\partial u}{\partial \mathbf{n}} = j \quad (2.5)$$

dove \mathbf{n} è la normale esterna unitaria e $j = \mathbf{J} \cdot \mathbf{n}$ è la densità di corrente sul bordo. Questa condizione ci permette anche di trovare $u|_{\partial\Omega}$ a meno di una costante.

Il modello più semplice si deduce considerando j come una funzione nota. Nel nostro esperimento consideriamo delle correnti iniettate solo da alcuni punti del solido quindi si può considerare j come

$$j = \begin{cases} \frac{I_i}{|e_i|} & \text{su } e_i, \quad \text{con } i = 1, 2, \dots, n \\ 0 & \text{altrimenti} \end{cases} \quad (2.6)$$

¹Potremmo supporre che il campo elettrico sia conservativo anche in presenza di correnti alternate a bassa frequenza

dove e_i è l' i -esimo elettrodo e $|e_i|$ è la sua area.

Tuttavia noi non siamo a conoscenza del valore di j ma solamente delle correnti iniettate \mathbf{I} e inoltre sappiamo che non è una buona approssimazione considerare la densità di corrente costante su tutto l'elettrodo. Da risultati sperimentali infatti (vedi [7]) possiamo vedere che questo modello genera una soprastima della resistività² e può quindi essere migliorato considerando elettrodi costituiti da un conduttore perfetto su cui abbiamo potenziale elettrico costante ma una densità di corrente non nota. In questo modo possiamo sostituire all' eq. (2.5) le seguenti condizioni al bordo:

$$\int_{e_i} \sigma \frac{\partial u}{\partial \mathbf{n}} dS = I_i, \quad i = 1, 2, \dots, n \quad (2.7a)$$

$$\sigma \frac{\partial u}{\partial \mathbf{n}} = 0, \quad \text{in } \partial\Omega \setminus \bigcup_{i=1}^n e_i \quad (2.7b)$$

dove la prima condizione impone che il flusso delle cariche su ogni elettrodo sia pari alla corrente iniettata, mentre la seconda impone che il flusso sia nullo sulla superficie dell'oggetto non coperta da elettrodi. La condizione sul potenziale può invece essere scritta come

$$u = U_i \quad \text{su } e_i, \quad i = 1, 2, \dots, n \quad (2.7c)$$

dove le costanti U_i non sono note ma sono i potenziali agli elettrodi che sono quindi parte della soluzione.

Questo modello è molto usato pur non essendo un problema di Neumann standard. Infatti risulta comunque ben formulato e ammette una soluzione unica se aggiungiamo la conservazione della carica

$$\sum_{i=1}^n I_i = 0 \quad (2.8)$$

e una condizione che identifica il potenziale di terra, ad esempio:

$$\sum_{i=1}^n U_i = 0 \quad (2.9)$$

²Ricordiamo che la resistività è l'inverso della conduttività ($\rho = \sigma^{-1}$)

Modello completo

Un ulteriore confronto con i dati sperimentali ci suggerisce un'ultima modifica al nostro modello arrivando a quello che noi chiameremo "*modello completo*" (noto come "*complete electrode model*").

Questo modello tiene conto della impedenza di contatto che si verifica tra la superficie degli elettrodi e quella dell'oggetto. Questa può essere spiegata fisicamente con il verificarsi di una reazione elettrochimica tra il metallo e il solido e viene modellizzata sostituendo l'eq. (2.7c) con

$$u + z_i \sigma \frac{\partial u}{\partial \mathbf{n}} = U_i \quad \text{su } e_i, \quad i = 1, 2, \dots, n \quad (2.10)$$

dove z_i è l'impedenza di contatto dell' i -esimo elettrodo.

Ricapitolando, il nostro problema diretto può essere così formalizzato:

Dati \mathbf{I} e σ , trovare u e \mathbf{U} tali che u sia soluzione di

$$\nabla \cdot \sigma \nabla u = 0 \quad \text{in } \Omega \quad (2.11a)$$

con le condizioni al bordo

$$u + z_i \sigma \frac{\partial u}{\partial \mathbf{n}} = U_i \quad \text{su } e_i, \quad i = 1, 2, \dots, n \quad (2.11b)$$

$$\sigma \frac{\partial u}{\partial \mathbf{n}} = 0, \quad \text{in } \partial\Omega \setminus \bigcup_{i=1}^n e_i \quad (2.11c)$$

sapendo che

$$\int_{e_i} \sigma \frac{\partial u}{\partial \mathbf{n}} dS = I_i, \quad i = 1, 2, \dots, n \quad (2.11d)$$

e

$$\sum_{i=1}^n I_i = 0 \quad , \quad \sum_{i=1}^n U_i = 0 \quad (2.11e)$$

I dati sperimentali (riportati in [7]) evidenziano come, tra tutti i modelli analizzati, questo sia quello che fornisce l'approssimazione migliore. Nel prossimo paragrafo dimostreremo l'esistenza e l'unicità della soluzione per questo problema.

2.3 Esistenza e unicità

Riportiamo qua la dimostrazione dell'esistenza e unicità della soluzione del modello completo di Somersalo e al. ([7]), che è un punto di riferimento importante per tutti gli studi fatti sulla tomografia ad impedenza elettrica e sulla ricostruzione di immagini tramite di essa anche se, come vedremo, il nostro problema specifico non soddisfa tutte le ipotesi fatte in questa dimostrazione.

Questa dimostrazione può essere fatta considerando potenziali e impedenze complesse ma nella nostra trattazione ci siamo limitati a considerare valori reali.

Cominciamo definendo più precisamente alcune notazioni e le ipotesi su Ω e σ . Sia Ω un dominio limitato in \mathbb{R}^m , $m = 2,3$ con un bordo $\partial\Omega$ regolare. Supponiamo inoltre che gli elettrodi e_i siano sottoinsiemi aperti connessi di $\partial\Omega$ la cui chiusura è disgiunta. Nel caso $m = 3$, supponiamo che i bordi degli elettrodi siano curve regolari su $\partial\Omega$.

Supponiamo inoltre che la conduttività σ sia continuamente differenziabile fino al bordo cioè che esista un intorno U di $\partial\Omega$ tale che $\sigma \in C^1(\bar{\Omega} \cap U)$.

Ora analizziamo le condizioni al bordo. Per le proprietà degli spazi di Sobolev, la funzione di traccia $\gamma_0 : u \mapsto u|_{\partial\Omega}$ può essere estesa per continuità, ottenendo

$$\gamma_0 : H^1(\Omega) \longrightarrow H^{1/2}(\partial\Omega) \quad (2.12)$$

La mappa $\gamma_1 : u \mapsto \sigma \frac{\partial u}{\partial \mathbf{n}}$ invece può essere definita con la formula di Green

$$\int_{\Omega} v \nabla \cdot \sigma \nabla u \, d\mathbf{x} = \int_{\partial\Omega} \sigma \frac{\partial u}{\partial \mathbf{n}} v \, dS - \int_{\Omega} \sigma \nabla u \cdot \nabla v \, d\mathbf{x} \quad (2.13)$$

Prima di cominciare la dimostrazione introduciamo lo spazio

$$H = H^1(\Omega) \oplus \mathbb{R}^n \quad (2.14)$$

Questo spazio conterrà la nostra soluzione che, come abbiamo visto, è data dal potenziale nell'interno del corpo e dal potenziale agli n elettrodi.

La seguente proposizione definisce la formulazione variazionale del nostro problema.

Proposizione 2.3.1. *Se $(u, \mathbf{U}) = (u(\mathbf{x}), (U_i)_{i=1}^n) \in H$ soddisfa l'eq (2.11),*

allora, per ogni $(v, \mathbf{V}) \in H$,

$$B((u, \mathbf{U}), (v, \mathbf{V})) = \sum_{i=1}^n I_i V_i \quad (2.15)$$

dove $B : H \times H \rightarrow \mathbb{R}$ è la forma sesquilineare definita come

$$B((u, \mathbf{U}), (v, \mathbf{V})) = \int_{\Omega} \sigma \nabla u \cdot \nabla v d\mathbf{x} + \sum_{i=1}^n \frac{1}{z_i} \int_{e_i} (u - U_i)(v - V_i) dS \quad (2.16)$$

Viceversa, se (u, \mathbf{U}) soddisfa (2.15) per ogni $(v, \mathbf{V}) \in H$, allora (u, \mathbf{U}) soddisfa anche (2.11).

Dimostrazione. Per prima cosa vediamo che le equazioni (2.11b) e (2.11c) possono essere combinate ottenendo

$$\sigma \frac{\partial u}{\partial \mathbf{n}} = \sum_{i=1}^n \frac{1}{z_i} (U_i - u) \chi_i \quad (2.17)$$

dove χ_i è la funzione caratteristica del n -esimo elettrodo che è nulla dappertutto tranne che sull'elettrodo i dove vale 1. Questa uguaglianza vale nello spazio $H^{-\frac{1}{2}}(\partial\Omega)$.

Ora se $(u, \mathbf{U}) \in H$ soddisfa (2.11), prendendo una funzione test $(v, \mathbf{V}) \in H$ arbitraria e usando la formula di Green, otteniamo

$$\int_{\partial\Omega} \sigma \frac{\partial u}{\partial \mathbf{n}} v dS - \int_{\Omega} \sigma \nabla u \cdot \nabla v d\mathbf{x} = 0 \quad (2.18)$$

o, usando la (2.17),

$$\sum_{i=1}^n \frac{1}{z_i} \int_{e_i} (U_i - u) v dS - \int_{\Omega} \sigma \nabla u \cdot \nabla v d\mathbf{x} = 0 \quad (2.19)$$

Considerando invece (2.11b) e (2.11d) possiamo scrivere

$$\int_{e_i} u dS = \int_{e_i} \left(U_i - z_i \sigma \frac{\partial u}{\partial \mathbf{n}} \right) dS = U_i |e_i| - z_i I_i \quad (2.20)$$

e quindi

$$\sum_{i=1}^n \frac{1}{z_i} V_i \left(\int_{e_i} u dS - U_i |e_i| + z_i I_i \right) \quad (2.21)$$

Unendo la (2.21) alla (2.19) otteniamo la (2.15) e la prima parte della dimostrazione è terminata.

Per provare l'altra parte della proposizione supponiamo che $(u, \mathbf{U}) \in H$ soddisfi (2.15) per ogni $(v, \mathbf{V}) \in H$. Le equazioni (2.11a)-(2.11d) si ottengono considerando delle scelte particolari per (v, \mathbf{V})

Se prendiamo $v \in C^\infty$ con $\text{supp}v \subset \Omega$ e $\mathbf{V} = 0$, otteniamo

$$\int_{\Omega} \sigma \nabla u \cdot \nabla v d\mathbf{x} = 0 \quad (2.22)$$

cioè u è soluzione debole di (2.11a). Prendendo invece $v \in H^1(\Omega)$ arbitraria, dalla (2.13) abbiamo che

$$\int_{\partial\Omega} \sigma \frac{\partial u}{\partial \mathbf{n}} v dS = \int_{\Omega} \sigma \nabla u \cdot \nabla v d\mathbf{x} \quad (2.23)$$

Questa relazione con la (2.15), quando $\mathbf{V} = 0$, ci dà

$$\int_{\partial\Omega} \left(\sigma \frac{\partial u}{\partial \mathbf{n}} + \sum_{i=1}^n \frac{1}{z_i} \chi_i (u - U_i) \right) v dS = 0 \quad (2.24)$$

Essendo v arbitraria, u deve soddisfare quindi (2.11b) e (2.11c). Infine, possiamo sostituire la (2.23) e la (2.24) nella (2.15). Poiché possiamo scegliere $\mathbf{V} \in \mathbb{R}^n$ arbitraria, abbiamo

$$- \int_{e_i} u dS + |e_i| U_i = z_i I_i \quad 1 \leq i \leq n \quad (2.25)$$

Quindi, usando questa con la (2.11b),

$$\int_{e_i} \sigma \frac{\partial u}{\partial \mathbf{n}} dS = \frac{1}{z_i} \int_{e_i} (U_i - u) dS = I_i \quad (2.26)$$

cioè u soddisfa anche la (2.11d). □

Ora possiamo ottenere l'esistenza e l'unicità della soluzione utilizzando il lemma di Lax-Milgram per la forma sesquilineare (2.16).

In realtà B non soddisfa tutte le ipotesi del lemma. Se prendiamo infatti

$$B((u, \mathbf{U}), (u, \mathbf{U})) = \int_{\Omega} \sigma |\nabla u|^2 d\mathbf{x} + \sum_{i=1}^n \frac{1}{z_i} \int_{e_i} |u - U_i|^2 dS \quad (2.27)$$

e la poniamo uguale a 0, questo non implica che $(u, \mathbf{U}) = 0$ ma solamente che

$$U_1 = \dots = U_n = \text{cost} \quad (2.28)$$

Da questo è immediato osservare che la forma B non soddisfa la condizione di coercività.

Dobbiamo quindi introdurre lo spazio quoziente

$$\dot{H} = H/\mathbb{R} \quad (2.29)$$

dato dalla seguente relazione di equivalenza in H :

$$(u, \mathbf{U}) \sim (v, \mathbf{V}) \Leftrightarrow U_1 - V_1 = \dots = U_n - V_n = \text{cost} \quad (2.30)$$

Su questo spazio introduciamo la norma

$$\|(u, \mathbf{U})\|_* = \left(\|\nabla u\|_{L^2(\Omega)} + \sum_{i=1}^n \int_{e_i} |u - U_i|^2 dS \right)^{1/2} \quad (2.31)$$

che risulta equivalente³ alla norma quoziente

$$\|(u, \mathbf{U})\| = \inf_{c \in \mathbb{R}} (\|\nabla u - c\|_{H^1(\Omega)} + \|\mathbf{U} - c\|_{\mathbb{R}^n(\Omega)})^{1/2} \quad (2.32)$$

Teorema 2.3.2. *Supponiamo che esistano $c_1, c_2, c_3 \in \mathbb{R}^+$ tali che*

$$c_1 \leq \sigma \leq c_2 \quad e \quad z_i \leq c_3 \quad \text{per } i = 1, \dots, n \quad (2.33)$$

Allora, per ogni pattern di correnti \mathbf{I} che soddisfa (2.8), esiste un'unica $(u, \mathbf{U}) \in \dot{H}$ per cui vale

$$B((u, \mathbf{U}), (v, \mathbf{V})) = \sum_{i=1}^n I_i V_i \quad (2.34)$$

per ogni $(v, \mathbf{V}) \in \dot{H}$.

Dimostrazione. Basta applicare il lemma di Lax-Milgram. Con la norma che abbiamo appena introdotto è facile verificare le condizioni di continuità e coercività su B . Ci rimane ancora da verificare che il funzionale lineare

$$f : (v, \mathbf{V}) \longmapsto \sum_{i=1}^n I_i V_i \quad (2.35)$$

³Per la dimostrazione dell'equivalenza delle due norme si rimanda a [7]

è ben definito e continuo. Se $(v, \mathbf{V}) \sim (\tilde{v}, \tilde{\mathbf{V}})$, allora per la (2.8) abbiamo

$$f(v, \mathbf{V}) = \sum_{i=1}^n I_i V_i = \sum_{i=1}^n I_i (V_i - cost) = \sum_{i=1}^n I_i \tilde{V}_i = f(\tilde{v}, \tilde{\mathbf{V}}) \quad (2.36)$$

cioè f è ben definito. Per verificare la continuità sia $c \in \mathbb{R}$ una costante tale che

$$(\|v - c\|_{H^1(\Omega)}^2 + \|\mathbf{V} - c\|_{\mathbb{C}^n}^2)^{1/2} \leq \|(v, \mathbf{V})\| + \varepsilon \quad (2.37)$$

allora

$$|f(v, \mathbf{V})| = \left| \sum_{i=1}^n I_i (V_i - c) \right| \leq \|\mathbf{I}\|_{\mathbb{C}^n} \|\mathbf{V} - c\|_{\mathbb{C}^n} \leq \|\mathbf{I}\|_{\mathbb{C}^n} (\|(v, \mathbf{V})\| + \varepsilon) \quad (2.38)$$

□

Corollario 2.3.3. *Sotto le stessì ipotesi del teorema precedente, se vale la (2.9) allora il modello completo ha un'unica soluzione.*

Dimostrazione. Un altro modo per provare l'unicità del modello completo è il seguente. Supponiamo che ci siano due soluzioni (u, \mathbf{U}) e $(\tilde{u}, \tilde{\mathbf{U}})$ del problema (2.11), in particolare siano soddisfatte (2.8) e (2.9). Se

$$(v, \mathbf{V}) = (u, \mathbf{U}) - (\tilde{u}, \tilde{\mathbf{U}}) \quad (2.39)$$

allora (v, \mathbf{V}) soddisfa (2.11a)-(2.11c) e

$$\int_{e_i} \sigma \frac{\partial v}{\partial \mathbf{n}} dS = 0 \quad (2.40)$$

Se usiamo (2.11a) e (2.11b) nella formula di Green

$$0 = \int_{\Omega} v \nabla \cdot \sigma \nabla v d\mathbf{x} = \int_{\partial\Omega} v \sigma \frac{\partial v}{\partial \mathbf{n}} dS - \int_{\Omega} \sigma |\nabla v|^2 d\mathbf{x} \quad (2.41)$$

otteniamo che

$$\int_{\Omega} \sigma |\nabla v|^2 d\mathbf{x} = \sum_{i=1}^n \int_{e_i} \left(V_i - z_i \sigma \frac{\partial v}{\partial \mathbf{n}} \right) \sigma \frac{\partial v}{\partial \mathbf{n}} dS \quad (2.42)$$

Usando poi (2.40) nella parte destra di (2.42), troviamo che

$$\int_{\Omega} \sigma |\nabla v|^2 d\mathbf{x} = - \sum_{i=1}^n z_i \left| \sigma \frac{\partial v}{\partial \mathbf{n}} \right|^2 \quad (2.43)$$

Dal momento che $z_i \geq 0$ per ogni i , (2.43) può essere soddisfatto solo se v è costante. Questo implica, per la (2.11b), che tutti i V_i sono uguali a questa costante. Da (2.9) poi troviamo che questa costante è 0. Quindi $(u, \mathbf{U}) = (\tilde{u}, \tilde{\mathbf{U}})$. \square

Osservazione 1. La dimostrazione dell'esistenza e unicità può essere provata anche con domini meno regolari come domini poligonali. Potremmo assumere per esempio che $\partial\Omega$ sia $C^{1,1}$ a tratti (vedi [4]).

Osservazione 2. La coercività della forma bilineare B è molto importante nella costruzione di un algoritmo agli elementi finiti stabile per la soluzione numerica del problema.

Osservazione 3. Una dimostrazione analoga può essere fatta anche per gli altri modelli che abbiamo visto in 2.2.1 prendendo il sottospazio di H in cui $u = \text{cost}$ su ogni elettrodo e_i e modificando opportunamente la forma bilineare B che, senza le impedenze di contatto, risulterebbe

$$\tilde{B}((u, \mathbf{U}), (v, \mathbf{V})) = \int_{\Omega} \sigma \nabla u \cdot \nabla v d\mathbf{x} \quad (2.44)$$

2.4 Linearità di $\mathbf{I} \rightarrow \mathbf{U}$

A prescindere dal metodo numerico usato per discretizzare il problema, o se si riuscisse a risolverlo analiticamente, possiamo sfruttare la linearità dell'operatore $\mathbf{I} \rightarrow \mathbf{U}$ per semplificare i calcoli. Chiamiamo f l'operatore che dalle correnti ci dà il valore del potenziale agli elettrodi. Abbiamo visto che questo operatore è lineare quindi possiamo velocizzare il calcolo introducendo una base su \mathbb{R}^n e calcolando il valore di f su una base.

Non è possibile utilizzare la base canonica di \mathbb{R}^n perché non rispetta l'eq. (2.8) di conservazione della carica. Introduciamo quindi la base

$$\mathcal{B} = \{e_i; 1 \leq i \leq n\} \quad \text{con} \quad \begin{cases} e_i^{(k)} = 1 - \frac{1}{n} & \text{se } k = i \\ e_i^{(k)} = -\frac{1}{n} & \text{altrimenti} \end{cases} \quad (2.45)$$

e scriviamo ogni possibile vettore di correnti \mathbf{I} come combinazione lineare di elementi

di \mathcal{B}

$$\mathbf{I} = \sum_{i=1}^n \alpha_i e_i \quad (2.46)$$

A questo punto, per la linearità di f , possiamo scrivere

$$\mathbf{U} = f(\mathbf{I}) = f\left(\sum_{i=1}^n \alpha_i e_i\right) = \sum_{i=1}^n \alpha_i f(e_i) \quad (2.47)$$

Quindi, se gli esperimenti fatti sul nostro oggetto sono in numero elevato (maggiore di n), quando cioè vogliamo calcolare il potenziale per molti pattern di correnti, possiamo velocizzare l'algoritmo scomponendo il vettore \mathbf{I} , trovando i coefficienti α_i , e calcolando con il nostro modello diretto solamente i valori $f(e_i)$.

N.B. Ricordiamo che il problema diretto risulta lineare rispetto al vettore \mathbf{I} delle correnti iniettate ma non rispetto alla conduttività σ . In tutto questo capitolo infatti, questa è supposta nota e non è stata parametrizzata in alcun modo ma questa non-linearità dell'operatore rispetto alla conduttività è una delle cause che rendono difficoltoso il problema inverso.

2.5 Risoluzione numerica

Sebbene in alcuni casi particolari, quando Ω è omogeneo⁴ e ha delle simmetrie particolari, sia possibile ottenere soluzioni analitiche ([7]), per il nostro problema è stato necessario discretizzare con il metodo degli *elementi finiti* (*FEM*).

In questo metodo il dominio è diviso in elementi tridimensionali (nel nostro caso tetraedri) e la soluzione u è approssimata come combinazione lineare delle funzioni di base φ_i , funzioni lineari "a tenda" che valgono 1 sul nodo i e sono nulle su tutti gli altri nodi della mesh. Il vettore \mathbf{U} invece, usando una base di \mathbb{R}^n , lo rappresentiamo con i coefficienti $\beta = (\beta_1, \dots, \beta_n)$. In realtà, l'eq. (2.9) del potenziale di terra ci permette di ridurre i gradi di libertà del problema quindi in realtà ci basta conoscere

⁴Cioè quando σ è una funzione regolare

i valori per $n - 1$ coefficienti. Avremo quindi

$$(u, \mathbf{U}) \approx (\tilde{u}, \tilde{\mathbf{U}}) = \left(\sum_{i=1}^N \alpha_i \varphi_i, \sum_{i=1}^{n-1} \beta_i \eta_i \right) \quad (2.48)$$

dove N è il numero di nodi della mesh, α_i è il valore del potenziale all' i -esimo nodo e $\eta_1 = (1, -1, 0, \dots, 0)$, $\eta_2 = (1, 0, -1, 0, \dots, 0)$, $\eta_{n-1} = (1, 0, \dots, 0, -1)$.⁵

Usando poi la formulazione variazionale dell'eq. (2.15) il problema si riduce ad un problema lineare del tipo

$$A\mathbf{b} = \mathbf{f} \quad (2.49)$$

Il termine noto $\mathbf{f} \in \mathbb{R}^{(N+n-1)}$ è costruito come

$$\mathbf{f} = \left(\mathbf{0}, \sum_{i=1}^n I_i(\eta_1)_i, \dots, \sum_{i=1}^n I_i(\eta_{n-1})_i \right)^T = \begin{pmatrix} \mathbf{0} \\ \tilde{\mathbf{I}} \end{pmatrix} \in \mathbb{R}^{N+n-1} \quad (2.50)$$

dove $\tilde{\mathbf{I}} = (I_1 - I_2, I_1 - I_3, \dots, I_1 - I_n)$ perché in (2.48) abbiamo preso come riferimento il primo elettrodo. A è invece la matrice a blocchi

$$A = \begin{pmatrix} Y & C \\ C^T & D \end{pmatrix} \quad (2.51)$$

e i blocchi sono ottenuti sostituendo nella forma sesquilineare (2.16) le funzioni di base φ_i e η_i

$$Y_{ij} = \int_{\Omega} \sigma \nabla \varphi_i \cdot \nabla \varphi_j dV + \sum_{k=1}^n \frac{1}{z_k} \int_{e_k} \varphi_i \varphi_j dS; \quad i, j = 1, 2, \dots, N \quad (2.52a)$$

$$C_{ij} = - \left(\frac{1}{z_1} \int_{e_1} \varphi_i dS - \frac{1}{z_{j+1}} \int_{e_{j+1}} \varphi_i dS \right); \quad i = 1, \dots, N; j = 1, \dots, (n-1) \quad (2.52b)$$

$$D_{ij} = \sum_{k=1}^n \frac{1}{z_k} \int_{e_k} (\eta_i)_k (\eta_j)_k dS = \begin{cases} \frac{|e_1|}{z_1}, & i \neq j \\ \frac{|e_1|}{z_1} + \frac{|e_{j+1}|}{z_{j+1}}, & i = j \end{cases} \quad i, j = 1, \dots, (n-1) \quad (2.52c)$$

A questo punto, risolvendo il sistema lineare, troviamo il vettore \mathbf{b} che contiene i coefficienti α_i e β_i

$$\mathbf{b} = (\alpha_1, \dots, \alpha_N, \beta_1, \dots, \beta_{n-1})^T \quad (2.53)$$

⁵Avremmo potuto utilizzare come base per \mathbf{U} e \mathbf{I} anche la (2.45)

In particolare i coefficienti β_i ci consentiranno di calcolare \mathbf{U} .

La matrice A risulta essere simmetrica e fortemente sparsa. Il sistema lineare risultante (2.49) può infine essere risolto con metodi diretti o iterativi. Tra i metodi diretti citiamo solamente la decomposizione di Cholesky che è utilizzata in particolare nelle mesh bidimensionali dove il numero di incognite non è molto elevato. Nel nostro caso, in cui abbiamo una mesh tridimensionale è invece più conveniente utilizzare un metodo iterativo come il gradiente coniugato. La convergenza di questo metodo può essere poi velocizzata scegliendo un preconditionatore opportunamente scelto (per esempio la fattorizzazione incompleta di Cholesky).

2.5.1 Caso semplificato

Useremo la discretizzazione appena illustrata quando, nel capitolo 4, vedremo una simulazione del problema diretto in cui abbiamo un corpo Ω cubico con due elettrodi estesi su due facce opposte. Quando risolveremo invece il problema inverso ci limiteremo ad una situazione in cui è possibile operare opportune semplificazioni, cioè quando gli elettrodi sono puntiformi e non c'è alcuna impedenza di contatto. Possiamo quindi identificarli con alcuni dei nodi esterni della mesh. Con queste semplificazioni non ci rifacciamo più al modello completo ma piuttosto al modello (2.6) in cui però la densità di corrente è una delta di Dirac concentrata nell'elettrodo che utilizziamo. In questo caso le uniche incognite sono i potenziali ai nodi e l'equazione discretizzata diventa

$$Y\mathbf{u} = \mathbf{i} \quad (2.54)$$

dove $\mathbf{u} = (u_1, \dots, u_N)$ e $\mathbf{i} = (i_1, \dots, i_N)$ sono rispettivamente i potenziali e le correnti ai nodi e Y è definita semplicemente da

$$Y_{ij} = \int_{\Omega} \sigma \nabla \varphi_i \cdot \nabla \varphi_j dV \quad (2.55)$$

Ovviamente sarà sufficiente calcolare ogni integrale solo su $\text{supp}(\varphi_i) \cap \text{supp}(\varphi_j)$.

Nelle nostre simulazioni considereremo una serie di k esperimenti con correnti che passano tra due elettrodi e misure di differenza di potenziale fatte tra altri due elettrodi. Con esperimenti di questo tipo il vettore \mathbf{i} sarà diverso da 0 solamente nelle

due posizioni corrispondenti ai nodi scelti per iniettare le correnti. Se per esempio nell'esperimento 1 iniettiamo una corrente tra il nodo i e il nodo j , il vettore \mathbf{i} avrà un 1 nella posizione i e un -1 nella posizione j . Infine, dopo aver applicato la (2.54), se vogliamo calcolare la differenza di potenziale che si instaura tra questi stessi elettrodi, questa sarà $U_1 = u_i - u_j$.

Per gli altri $k - 1$ esperimenti potremo poi prendere in considerazione coppie di nodi diversi e in questo modo otterremo le varie componenti del nostro vettore \mathbf{U} . Per le nostre simulazioni dovremo quindi indicare uno schema di misurazioni che rappresenti gli elettrodi usati e il valore di corrente iniettata per ciascun esperimento.

2.6 Conclusioni

In questo capitolo abbiamo ottenuto una formulazione matematica ed un metodo numerico per risolvere il nostro problema diretto. Nel prossimo capitolo, a partire da questi risultati, potremmo vedere come possiamo trovare la soluzione del problema inverso.

Nella dimostrazione della sezione 2.3 abbiamo supposto σ funzione regolare mentre nel nostro problema è una funzione costante ma con una discontinuità lungo il piano γ . Questo vuol dire che, in generale, la nostra soluzione u non sarà regolare. In [8] e [3] possiamo trovare un approfondimento sulle soluzioni e sulla convergenza dei metodi numerici per problemi di questo tipo (equazioni ellittiche con coefficienti discontinui).

Capitolo 3

Problema inverso

3.1 Introduzione

Un problema diretto può essere interpretato come la ricerca di un unico effetto a partire da cause conosciute, usando appropriati modelli fisici. Questi problemi risultano spesso ben posti ma nelle scienze moderne si incontrano anche problemi inversi, cioè problemi in cui è necessario trovare le cause di un dato effetto o trovare la giusta legge fisica conoscendo la causa e gli effetti.

In base a quanto detto nella sezione 2.1 e alle notazioni usate nella sezione 2.2, il nostro problema inverso consiste nel trovare σ a partire dai dati al bordo $u|_{\partial\Omega}$ e j . Per essere più precisi, nel nostro caso conosciamo solamente i valori agli elettrodi \mathbf{I} e \mathbf{V} . Avendo rappresentato il nostro problema diretto come una mappa $f : (\sigma, \mathbf{I}) \rightarrow \mathbf{V}$, invertendo f rispetto al parametro σ otteniamo una rappresentazione del nostro problema inverso come una mappa $\bar{f} : (\mathbf{I}, \mathbf{V}) \rightarrow \sigma$.

La conduttività σ nel nostro caso dipende solo dalla posizione del piano e quindi dai parametri che usiamo per rappresentare il piano γ , ma in generale può non essere costante e approssimata in vari modi, quindi dipendere da più parametri. Rappresentiamo questa parametrizzazione come una dipendenza di σ dal vettore di parametri \mathbf{m}

$$\sigma = \sigma(\mathbf{m}) \tag{3.1}$$

Poiché il problema diretto, come abbiamo osservato, non è lineare rispetto a \mathbf{m} , il problema inverso che dobbiamo affrontare è *non-lineare*.

Prima di affrontare però questo problema specifico sarà necessario in questo capitolo dare qualche accenno più generale alla teoria dei problemi inversi ed in particolare al problema della buona posizione.

Nella sezione 3.2 sono illustrati, con una notazione molto generale, i diversi tipi di approccio e le principali tecniche di risoluzione dei problemi inversi. Spesso questi non risultano ben posti quindi in 3.3 vedremo anche delle tecniche di regolarizzazione. Nella sezione 3.4 vedremo invece come è stato possibile nel nostro caso risolvere il problema inverso.

Ricordiamo che un problema matematico si definisce *ben posto* (nel senso di Hadamard) se soddisfa le seguenti condizioni

1. *Esiste una soluzione.*
2. *La soluzione è unica.*
3. *La soluzione dipende in modo continuo dai dati cioè a sufficientemente piccole variazioni dei dati di partenza corrispondono arbitrariamente piccole variazioni nella soluzione.*

3.2 Problemi inversi e minimi quadrati

I possibili approcci ai problemi inversi sono due: un approccio *deterministico* o un approccio *statistico*. Noi approfondiremo solo il primo e più avanti descriveremo brevemente il secondo.

Per prima cosa mettiamoci in una situazione molto generale e costruiamo un semplice modello di osservazione della realtà in cui $\mathbf{m} \in \mathcal{M} \subseteq \mathbb{R}^m$ rappresenta i parametri del nostro sistema fisico e $\mathbf{d} \in \mathcal{D} \subseteq \mathbb{R}^n$ rappresenta i risultati delle misurazioni fatte sul sistema. Se conosciamo la relazione che lega queste due grandezze possiamo rappresentare il problema diretto come una mappa $h : \mathcal{M} \rightarrow \mathcal{D}$ e abbiamo quindi la relazione

$$\mathbf{d} = h(\mathbf{m}) \tag{3.2}$$

Se il problema è lineare rispetto ai parametri la relazione si può scrivere come

$$\mathbf{d} = H\mathbf{m}, \quad H \in \mathbb{R}^{n \times m} \quad (3.3)$$

Nel modello precedente, se $h(\mathbf{m})$, o H , e \mathbf{m} sono noti, il risultato \mathbf{d} può essere calcolato. Questo è il classico problema diretto che spesso è ben condizionato e la soluzione è unica. Nel problema inverso \mathbf{d} è noto perché misurato tramite esperimenti o simulato usando un modello diretto, e si deve calcolare il vettore di parametri \mathbf{m} . Dobbiamo quindi trovare \mathbf{m} tale che

$$\mathbf{d} - h(\mathbf{m}) = 0 \quad (3.4)$$

In realtà, nella maggior parte dei casi, non è possibile utilizzare questa equazione perché non ha alcuna soluzione classica in quanto spesso si ha che $\mathbf{d} \notin \mathcal{I}(h)$, dove

$$\mathcal{I}(h) = \{\mathbf{d} \in \mathbb{R}^n : \mathbf{d} = h(\mathbf{m}) \text{ per qualche } \mathbf{m} \in \mathbb{R}^m\} \quad (3.5)$$

Questo può essere dovuto agli inevitabili errori nella modellizzazione o agli errori nelle misurazioni.

Per risolvere questo problema dobbiamo quindi cercare una soluzione generalizzata. Una soluzione di questo tipo la si può trovare con il metodo dei *minimi quadrati* (*LS Method*), definendo la soluzione

$$\mathbf{m}_{LS} = \min_{\mathbf{m} \in M} \|\mathbf{d} - h(\mathbf{m})\|_2^2 \quad (3.6)$$

Questo approccio garantisce l'esistenza di una soluzione; tuttavia potrebbero non essere soddisfatte le ultime due condizioni di Hadamard sull'unicità e sulla continuità rispetto ai dati.

Prima di affrontare questi problemi vediamo ora come si può risolvere l'eq. (3.6) nel caso di un operatore h lineare o non-lineare.

Caso lineare

Nonostante la maggior parte dei problemi inversi sia non-lineare, per risolverli si fa uso di linearizzazioni locali, quindi tutte le considerazioni fatte per problemi

lineari sono molto importanti perché si applicano localmente anche ai problemi non lineari.

La soluzione dell'eq. (3.6) nel caso lineare, se per il momento supponiamo che la matrice H abbia rango uguale al numero di colonne (e quindi $m < n$), è data da

$$H^T H \mathbf{m}_{LS} = H^T \mathbf{d} \quad \Rightarrow \quad \mathbf{m}_{LS} = H^\dagger \mathbf{d} \quad (3.7)$$

dove $H^\dagger = (H^T H)^{-1} H^T$ è l'inversa generalizzata di Moore-Penrose.

Nel caso che m sia maggiore di n non è verificata la seconda condizione di Hadamard. L'unicità della soluzione dipende infatti dal nucleo di h cioè lo spazio $\mathcal{N}(h) = \{\mathbf{m} \in \mathcal{M} : h(\mathbf{m}) = 0\}$. Se questo non è vuoto possiamo prendere un suo elemento \mathbf{m}_N e sommarlo alla soluzione \mathbf{m}_{LS} ottenendo un'altra soluzione del problema. Allo stesso modo ogni vettore $k\mathbf{m}_N + \mathbf{m}_{LS}$ sarà soluzione del problema per ogni k reale. In questi casi può essere utile utilizzare la decomposizione ai valori singolari che vedremo più avanti.

Caso non-lineare

Supponendo di nuovo che m sia minore di n possiamo giungere a conclusioni simili anche per il caso non lineare, utilizzando il metodo di Gauss-Newton, passando cioè attraverso un'approssimazione dell'operatore h e trovando \mathbf{m}_{LS} con un procedimento iterativo.

Il primo passo è approssimare il funzionale $\varphi(\mathbf{m}) = \|\mathbf{d} - h(\mathbf{m})\|^2$ in un intorno di \mathbf{m}_k con la serie di Taylor al secondo ordine

$$\tilde{\varphi}(\mathbf{m}) = \varphi(\mathbf{m}_k) + \frac{\partial \varphi}{\partial \mathbf{m}}(\mathbf{m}_k)(\mathbf{m} - \mathbf{m}_k) + \frac{1}{2}(\mathbf{m} - \mathbf{m}_k)^T \frac{\partial^2 \varphi}{\partial \mathbf{m}^2}(\mathbf{m}_k)(\mathbf{m} - \mathbf{m}_k) \quad (3.8)$$

Troviamo \mathbf{m}_{k+1} , una migliore approssimazione di \mathbf{m}_{LS} imponendo la condizione che minimizza il funzionale approssimato $\tilde{\varphi}$

$$\frac{\partial \tilde{\varphi}}{\partial \mathbf{m}}(\mathbf{m}_{k+1}) = 0 \quad (3.9)$$

Se supponiamo che la matrice Hessiana $\frac{\partial^2 \varphi}{\partial \mathbf{m}^2}(\mathbf{m}_k)$ sia invertibile otteniamo quindi il seguente processo iterativo

$$\mathbf{m}_{k+1} = \mathbf{m}_k - \left(\frac{\partial^2 \varphi}{\partial \mathbf{m}^2}(\mathbf{m}_k) \right)^{-1} \frac{\partial \varphi}{\partial \mathbf{m}}(\mathbf{m}_k) \quad (3.10)$$

Esplicitando poi il gradiente e l'Hessiano come

$$\frac{\partial \tilde{\varphi}(\mathbf{m})}{\partial \mathbf{m}}(\mathbf{m}_k) = -2 \left(\frac{\partial h}{\partial \mathbf{m}^2}(\mathbf{m}_k) \right)^T (\mathbf{d} - h(\mathbf{m}_k)) \quad (3.11)$$

$$\frac{\partial^2 \varphi}{\partial \mathbf{m}^2}(\mathbf{m}_k) = 2 \left(\frac{\partial h}{\partial \mathbf{m}}(\mathbf{m}_k) \right)^T \left(\frac{\partial h}{\partial \mathbf{m}}(\mathbf{m}_k) \right) - 2 \sum_{i=1}^m \left(\frac{\partial^2 h_i}{\partial \mathbf{m}^2}(\mathbf{m}_k) \right) (\mathbf{d}_i - h_i(\mathbf{m}_k))$$

e trascurando i termini del secondo ordine, l'eq. (3.10) diventa

$$\mathbf{m}_{k+1} = \mathbf{m}_k - J_k^\dagger (h(\mathbf{m}_k) - \mathbf{d}) \quad \text{con } J_k^\dagger = (J_k^T J_k)^{-1} J_k^T \quad (3.12)$$

dove $J_k = \frac{\partial h}{\partial \mathbf{m}}(\mathbf{m}_k)$ è lo Jacobiano della trasformazione h calcolato al passo k .

3.3 Regolarizzazione tramite SVD

Come abbiamo visto, nell'inversione del problema subentrano dei problemi quando la matrice H o lo Jacobiano J hanno un numero di righe minore del numero di colonne, quando non hanno rango massimo, oppure quando queste matrici hanno un numero di condizionamento elevato. Nei primi due casi siamo di fronte ad un problema mal posto in cui la soluzione non è unica. Nell'ultimo caso invece il problema risulta mal condizionato, cioè pur verificando le tre condizioni di buona posizione, una variazione dei dati viene amplificata nel risultato da un fattore molto elevato. Quindi anche se il problema risulta ben posto in senso stretto, quando operiamo una discretizzazione ed un'approssimazione numerica, ci troviamo di fronte in pratica alle stesse difficoltà dei problemi mal posti che non verificano la terza condizione di Hadamard. Spesso questi problemi vengono chiamati *problemi discreti mal-posti* (*discrete ill-posed problems*).

In tutti questi casi è possibile utilizzare la decomposizione ai valori singolari (*SVD, Singular Values Decomposition*) per studiare e poter poi regolarizzare il problema.

La decomposizione ai valori singolari consiste nello scrivere una matrice $A \in \mathbb{R}^{n \times m}$ come

$$A = U A V^T = \sum_{i=1}^p \mathbf{u}_i \lambda_i \mathbf{v}_i^T \quad (3.13)$$

dove $p = \min(n, m)$, $U = (\mathbf{u}_1 \dots \mathbf{u}_n) \in \mathbb{R}^{n \times n}$ e $V = (\mathbf{v}_1 \dots \mathbf{v}_m) \in \mathbb{R}^{m \times m}$ sono matrici ortonormali mentre $\Lambda = \text{diag}(\lambda_1 \dots \lambda_p) \in \mathbb{R}^{n \times m}$ è una matrice diagonale. I vettori \mathbf{u}_i e \mathbf{v}_i sono chiamati rispettivamente *vettori singolari di sinistra e di destra* mentre i λ_i sono i *valori singolari* per cui vale la relazione:

$$\lambda_1 \geq \lambda_2 \geq \dots \lambda_r \geq \dots \geq \lambda_p \geq 0 \quad (3.14)$$

Se il rango r della matrice A risulta minore di p , gli ultimi $\lambda_{r+1} \dots \lambda_p$ sono nulli. Tanti più valori singolari positivi sono vicini allo zero tanto più la matrice A risulta avere numero di condizionamento elevato e di conseguenza il problema lineare $\|A\mathbf{x} - \mathbf{b}\|_2^2$ risulta mal condizionato.

Utilizzando questo tipo di decomposizione per la matrice H del nostro modello e riscrivendo \mathbf{m} come combinazione lineare dei vettori singolari \mathbf{v}_i , l'equazione $\mathbf{d} = H\mathbf{m}$ diventa

$$\mathbf{d} = \sum_{i=1}^r \lambda_i (\mathbf{v}_i^T \mathbf{m}) \mathbf{u}_i \quad (3.15)$$

Nel nostro problema i vettori singolari sinistri e destri hanno una interpretazione importante che ci permette di capire meglio l'inversione del problema. Infatti i vettori \mathbf{v}_i generano lo spazio dei parametri e gli \mathbf{u}_i quello delle misure. La soluzione del nostro problema inverso potrà quindi essere espressa come combinazione lineare dei \mathbf{v}_i con coefficienti che risultano essere $\frac{\mathbf{u}_i^T \mathbf{d}}{\lambda_i}$. I vettori singolari sono ortonormali quindi il fattore che amplifica o riduce il contributo di un determinato vettore \mathbf{v}_i rispetto alle misure è il valore singolare corrispondente λ_i .

Quando un valore singolare $\lambda_i > 0$ è molto piccolo, vuol dire che per delle variazioni nei parametri parallele al vettore \mathbf{v}_i , anche molto grandi, si hanno delle variazioni molto piccole nelle misurazioni \mathbf{d} . Risulta quindi difficile invertire il problema perché la soluzione risulta molto sensibile agli errori di misurazione o di modellizzazione. Quando invece un valore singolare λ_i è nullo, vuol dire che per variazioni nei parametri parallele al vettore \mathbf{v}_i , il risultato non cambia. Il problema di minimizzazione ha quindi infinite soluzioni, in questo caso dobbiamo disporre di ulteriori informazioni per poter scegliere quale delle infinite soluzioni è quella che ci interessa.

In entrambi i casi per trovare il valore \mathbf{m}_{LS} si fa ricorso alla *decomposizione ai valori singolari troncata (TSVD)*. Possiamo cioè, dopo aver trovato la decomposizione ai valori singolari, fermare la ricostruzione all'indice $k_0 \leq r \leq p$ tale per cui $\lambda_k < \varepsilon \forall k > k_0$ con $\varepsilon \in \mathbb{R}$ fissato. Ciò trascuriamo le componenti che risulterebbero troppo sensibili ad eventuali errori di misurazione di \mathbf{d} . Nel caso lineare l'equazione (3.7) si può quindi riscrivere come

$$\mathbf{m}_{LS} = \sum_{i=1}^{k_0} \frac{\mathbf{u}_i^T \mathbf{d}}{\lambda_i} \mathbf{v}_i \quad (3.16)$$

Per il caso non-lineare valgono le stesse considerazioni fatte per il caso lineare. Infatti ad ogni passo del processo iterativo, se ci troviamo di fronte ad una matrice jacobiana J con rango non massimo o con numero di condizionamento elevato possiamo sostituirla con la sua decomposizione troncata.

L'analisi dei valori singolari ci permette quindi di suddividere lo spazio dei parametri \mathcal{M} nei seguenti sottospazi:

- Il sottospazio generato da $\{\mathbf{v}_{(r+1)}, \dots, \mathbf{v}_q\}$ che è il nucleo di H e che dà origine ad osservazioni nulle. Una variazione dei parametri appartenente a questo sottospazio non influisce sulle misurazioni \mathbf{d} . Se questo sottospazio non è vuoto, sono necessarie ulteriori informazioni per invertire in modo unico il problema a partire dai dati \mathbf{d} .
- Il sottospazio generato da $\{\mathbf{v}_{(k_0+1)}, \dots, \mathbf{v}_r\}$ che dà luogo ad variazioni molto piccole nei dati. Nell'inversione del problema, la soluzione risulta quindi molto sensibile a piccole variazioni ed errori di misura di \mathbf{d} .
- Il sottospazio generato da $\{\mathbf{v}_1, \dots, \mathbf{v}_z\}$ che dà origine a osservazioni affidabili.

Il grado di mal condizionamento del nostro problema diretto può essere espresso del numero di condizionamento della matrice H che con la SVD è dato da

$$\text{cond}(H) = \frac{\lambda_1}{\lambda_p} \quad (3.17)$$

Questo assumerà valori molto grandi quando l'ultimo valore singolare considerato è piccolo e tende all'infinito se la matrice non ha rango massimo. Fermandoci invece

all'autovalore λ_k , riusciamo ad ottenere un numero di condizionamento per il nostro problema sufficientemente basso.

Usando la decomposizione troncata della matrice jacobiana nella ricerca della soluzione dei problemi inversi non lineari possiamo ottenere un algoritmo più stabile ma si perdono alcune informazioni. In particolare, nella ricotruzione di immagini tramite EIT, si ottiene il fenomeno illustrato nel capitolo 1 che rende impossibile ricostruire immagini ad elevata frequenza spaziale. I valori singolari più piccoli infatti sono quelli associati a queste frequenze.

Nel nostro problema invece, utilizzando le informazioni a priori, non dobbiamo fare una ricostruzione spaziale della conduttività e i parametri da stimare sono solo quelli del piano. Abbiamo quindi utilizzato questo metodo nel caso di una matrice jacobiana con rango non massimo o con numero di condizionamento elevato. In questo modo otteniamo delle stime dei parametri più stabili e che tengono conto solo delle componenti più significative dei dati. Nel prossimo capitolo vedremo in dettaglio come abbiamo potuto utilizzare questa decomposizione.

3.3.1 Altre tecniche di regolarizzazione

Un altro metodo di regolarizzazione molto usato che ottiene risultati simili alla decomposizione troncata, è la regolarizzazione di *Tikhonov*. In questo metodo si aggiunge un termine al funzionale da minimizzare. In questo modo l'equazione da risolvere diventa

$$\mathbf{m}_{(\alpha,L)} = \min_{\mathbf{m} \in M} (\|\mathbf{d} - h(\mathbf{m})\|^2 + \alpha F(\mathbf{m})) \quad (3.18)$$

dove $F(\mathbf{m}) \geq 0$ è un funzionale di regolarizzazione e $\alpha \geq 0$ è il parametro di regolarizzazione. Di solito il funzionale F assume la forma

$$F(\mathbf{m}) = \|L(\mathbf{m} - \mathbf{m}_0)\|^2 \quad (3.19)$$

dove L è una matrice di regolarizzazione appropriata e \mathbf{m}_0 è una stima a priori dei parametri.

Nel caso lineare questa scelta porta a trovare la soluzione

$$\mathbf{m}_{(\alpha,L)} = (H^T H + \alpha L^T L)^{-1} (H^T \mathbf{d} + \alpha L^T L \mathbf{m}_0) \quad (3.20)$$

mentre nel caso non lineare otteniamo

$$\mathbf{m}_{(\alpha,L)}^{(k)+1} = \mathbf{m}_{(\alpha,L)}^{(k)} - (J_k^T J_k + \alpha L^T L)^{-1} [J_k^T (\mathbf{m}_{(\alpha,L)}^{(k)} - \mathbf{d}) + \alpha L^T L (\mathbf{m}_{(\alpha,L)}^{(k)} - \mathbf{m}_0)] \quad (3.21)$$

Prendendo $L = I$ e $\mathbf{m}_0 = 0$ e utilizzando la decomposizione ai valori singolari l'eq. (3.20) diventa

$$\mathbf{m}_{(\alpha,L)} = \sum_{i=1}^M \frac{\lambda_i^2}{\lambda_i^2 + \alpha} \frac{\mathbf{u}_i^T \mathbf{d}}{\lambda_i} \mathbf{v}_i \quad (3.22)$$

In questo modo possiamo quindi vedere come opera questa regolarizzazione. Il contributo dei valori singolari molto bassi viene smorzato dal termine $\frac{\lambda_i^2}{\lambda_i^2 + \alpha}$. La scelta di L può essere diversa e va studiata per ogni singolo problema.

Altri metodi di regolarizzazione si ottiene con un approccio ai problemi inversi di tipo probabilistico, in cui si rappresenta il vettore dei parametri come una variabile casuale multivariata e si utilizza un punto di vista bayesiano. A partire da una densità a priori che rappresenta le nostre informazioni a priori e dai dati del problema, si calcola una funzione di densità a posteriori. Da questa poi si trova la soluzione del problema inverso come una stima puntuale calcolata attraverso i metodi Monte Carlo. I vari metodi di regolarizzazione consistono nello scegliere la giusta funzione di densità a priori a seconda delle informazioni a priori che abbiamo.

Il confine tra i due approcci tuttavia non è netto, infatti molti metodi deterministici possono essere pensati come particolari metodi probabilistici.

3.4 Inversione del problema

Nel nostro caso specifico l'unico parametro da stimare è la conduttività σ e questa può assumere solamente due valori costanti σ_1 e σ_2 . Supponendo noti questi due valori, e rappresentando il piano di separazione γ come

$$\gamma : ax + by + cz + d = 0, \quad (3.23)$$

il nostro vettore \mathbf{m} da stimare è

$$\mathbf{m} = (a,b,c,d) \quad (3.24)$$

Per trovare la soluzione del problema inverso consideriamo il modello semplificato analizzato in 2.5.1, in cui gli elettrodi puntiformi sono posti sui nodi esterni del corpo. Supponiamo di avere effettuato sul nostro oggetto una serie di k esperimenti facendo passare, come spiegato in 2.5.1, una corrente elettrica attraverso due elettrodi e calcolando la differenza di potenziale tra altri due elettrodi; indichiamo con il vettore $\mathbf{V} \in \mathbb{R}^k$ il risultato di questi esperimenti.

Il nostro modello, con la discretizzazione (2.54), ci permette di trovare il valore dei potenziali ai nodi per un determinato valore di \mathbf{m} come

$$\mathbf{u} = Y^{-1}\mathbf{i} \quad (3.25)$$

Otteniamo quindi \mathbf{u} in funzione di \mathbf{m} . Come abbiamo visto in in 2.5.1, le nostre misurazioni V_r , $1 \leq r \leq k$ non rappresentano i potenziali su tutti i nodi ma solamente differenze di potenziali tra determinati nodi che consideriamo come elettrodi. Quindi dovremmo in realtà considerare, per ogni esperimento, una combinazione lineare di \mathbf{u} del tipo: $(u_i - u_j)$, con i e j che ci vengono dati dal particolare schema di misurazioni che effettueremo. Questi valori formeranno il vettore $\mathbf{U} = \mathbf{U}(\mathbf{m})$ che dovremo confrontare con \mathbf{V} .

In base a quanto detto sui problemi inversi, dobbiamo quindi calcolare lo jacobiano

$$J = \frac{\partial \mathbf{U}}{\partial \mathbf{m}} = \begin{pmatrix} \frac{\partial U_1}{\partial a} & \cdots & \frac{\partial U_1}{\partial d} \\ \cdots & \cdots & \cdots \\ \frac{\partial U_k}{\partial a} & \cdots & \frac{\partial U_k}{\partial d} \end{pmatrix} \quad (3.26)$$

Ogni elemento poi si può scrivere come

$$\frac{\partial U_m}{\partial a,b,c,d} = \frac{\partial u_{m,i}}{\partial a,b,c,d} - \frac{\partial u_{m,j}}{\partial a,b,c,d} \quad m = 1, \dots, k \quad (3.27)$$

É quindi necessario trovare $\frac{\partial \mathbf{u}}{\partial a,b,c,d}$.

Partendo dall'eq. $Y\mathbf{u} = \mathbf{i}$, se deriviamo entrambi i membri e osserviamo che \mathbf{i} non dipende dai parametri, si ottiene

$$Y \frac{\partial \mathbf{u}}{\partial a,b,c,d} = -\frac{\partial Y}{\partial a,b,c,d} \mathbf{u} = -\frac{\partial Y}{\partial a,b,c,d} Y^{-1} \mathbf{i} \quad (3.28)$$

da cui

$$\frac{\partial \mathbf{u}}{\partial a,b,c,d} = -Y^{-1} \frac{\partial Y}{\partial a,b,c,d} Y^{-1} \mathbf{i} \quad (3.29)$$

Il problema quindi, nota Y , si riduce a calcolare $\frac{\partial Y}{\partial a,b,c,d}$. Nel capitolo 4 vedremo come abbiamo potuto calcolare questa matrice nel nostro caso specifico.

Utilizzando questo procedimento è quindi possibile calcolare lo jacobiano e utilizzarlo nella formula (3.12) per trovare una stima di a,b,c,d e quindi di γ .

3.5 Conclusione

In questo capitolo abbiamo dato i requisiti teorici per poter invertire il nostro problema. È evidente però che, rappresentando il piano γ come

$$ax + by + cz + d = 0 \quad (3.30)$$

ci troviamo di fronte ad un problema mal posto. Ogni piano infatti non ha una rappresentazione unica ed è possibile moltiplicare i parametri per una qualunque costante ottenendo lo stesso piano.

Rappresentando il piano con tre soli parametri, per esempio dividendo la (3.30), non è possibile rappresentare i piani paralleli all'asse x . Per ottenere una rappresentazione unica possiamo però aggiungere un vincolo del tipo

$$a + b + c + d = cost \quad (3.31)$$

In questo modo però il nostro problema di inversione, utilizzando il metodo dei minimi quadrati, diventa un problema di minimo vincolato. Un metodo per risolverlo può essere quello di inserire il vincolo (3.31) in forma implicita all'interno del funzionale da minimizzare ottenendo

$$\mathbf{m}_{LS} = \min_{\mathbf{m} \in \mathbf{M}} (\|\mathbf{d} - h(\mathbf{m})\|^2 + \alpha(a + b + c + d - 1)^2) \quad (3.32)$$

dove abbiamo scelto una costante uguale a 1 e dove α è un parametro per dare più o meno importanza al secondo termine rispetto al primo.

Nella letteratura sulla ricostruzione di immagini tramite EIT ([9]), metodi di questo tipo sono usati per incorporare nel funzionale le informazioni a priori. In particolare quando sappiamo che i parametri devono essere compresi tra dei valori

precisi, possiamo costruire dei metodi come la *barriera logaritmica*, in cui il vincolo da soddisfare è inserito nel funzionale tramite una funzione che cresce molto rapidamente quando i parametri non soddisfano il vincolo.

Come vedremo nella sezione 4.4, l'eq (3.32) non ci ha portato ad ottenere buoni risultati. Nel nostro caso infatti abbiamo preso una funzione lineare che, non avendo una crescita elevata, non riesce a rispettare sempre il vincolo.

Capitolo 4

Sintesi del lavoro svolto e test eseguiti

4.1 Introduzione

In questo capitolo cercheremo, spiegando le varie fasi del lavoro svolto, di analizzare i vari problemi incontrati strada facendo e di spiegare quali sono stati i metodi usati per risolverli, mettendo in luce soprattutto gli aspetti matematici interessanti. Saranno anche spiegati tutti i metodi e i test utilizzati per verificare e testare i programmi sviluppati e saranno riportate le varie modifiche applicate ai programmi per poter risolvere i problemi che si presentavano andando avanti col lavoro.

Nella seconda sezione ci occuperemo della prima parte del lavoro cioè l'assemblaggio della matrice degli elementi finiti; in 4.3 vedremo invece come abbiamo potuto calcolare la derivata di questa matrice rispetto ai parametri del piano. L'ultimo passo è stato l'elaborazione della funzione per trovare la soluzione del problema inverso, questa fase del lavoro è riportata in 4.4. Infine in 4.5 vedremo un'importante test conclusivo che riassume i risultati di tutto il lavoro. La versione definitiva di alcune delle funzioni elaborate è invece riportata nell'appendice A.

4.2 Assemblaggio della matrice degli EF

Come abbiamo visto in 2.5.1, usando la discretizzazione degli elementi finiti riduciamo il nostro problema diretto al calcolo delle soluzioni di un sistema lineare del tipo

$$A\mathbf{x} = \mathbf{b} \quad (4.1)$$

dove \mathbf{b} è il vettore che contiene le informazioni sui valori al bordo, nel nostro caso le correnti iniettate, mentre \mathbf{x} è la nostra incognita che rappresenta i potenziali ai nodi della mesh e A è una matrice che dipende dalla mesh definita come

$$A = \{A_{ij}\} \quad \text{dove} \quad A_{ij} = \int_{\Omega} \sigma \nabla \varphi_i \cdot \nabla \varphi_j dV \quad (4.2)$$

dove φ_i e φ_j sono le funzioni di base che valgono 1 sui nodi i e j e 0 su tutti gli altri e σ è la conduttività.

La prima funzione che abbiamo sviluppato è quindi quella che assembla la matrice A che chiameremo `FEMmatrix.m` a partire da una mesh tridimensionale, dalla conoscenza dei valori σ_1 , σ_2 , a , b , c , d e dei gradienti delle funzioni di base. Per prima cosa è stato necessario elaborare una routine chiamata `inters.m` che calcola le posizioni degli elementi tetraedrici della mesh rispetto al piano.

In questa funzione, per ogni nodo (x_i, y_i, z_i) viene calcolato il valore

$$f(x_i, y_i, z_i) = ax_i + by_i + cz_i + d \quad (4.3)$$

e in base al suo segno si può capire se il nodo sta sopra, sotto o sul piano.

Quindi ogni elemento viene caratterizzato con tre interi che rappresentano rispettivamente il numero di vertici per cui si ha $f(x_i, y_i, z_i) > 0$, $f(x_i, y_i, z_i) = 0$ e $f(x_i, y_i, z_i) < 0$.

A questo punto, la funzione principale, per calcolare la matrice A , compie per ogni tetraedro E un doppio ciclo sugli indici i e j che variano tra i quattro nodi del tetraedro.

- Per i tetraedri che non intersecano il piano il contributo al termine A_{ij} della matrice è semplicemente

$$\sigma \nabla \varphi_i \cdot \nabla \varphi_j \int_E dV \quad \text{con} \quad \sigma = \sigma_1, \sigma_2 \quad (4.4)$$

dove $\varphi_i|_E$ è il gradiente della funzione di base φ_i calcolato sull'elemento E . Questo risulta costante su ogni elemento in quanto le φ_i sono funzioni lineari. Note le coordinate dei vertici (x_1, y_1, z_1) , (x_2, y_2, z_2) , (x_3, y_3, z_3) , (x_4, y_4, z_4) risulta facile calcolare il volume $V_E = \int_E dV$ usando la seguente formula:

$$V_E = \frac{1}{6} |\det C| \quad \text{con} \quad C = \begin{pmatrix} (x_2 - x_1) & (y_2 - y_1) & (z_2 - z_1) \\ (x_3 - x_1) & (y_3 - y_1) & (z_3 - z_1) \\ (x_4 - x_1) & (y_4 - y_1) & (z_4 - z_1) \end{pmatrix} \quad (4.5)$$

In questo caso si inseriscono anche i tetraedri che toccano il piano con un vertice, con uno spigolo o con una faccia ma senza intersecarlo;

- Per i tetraedri che hanno tre vertici da una parte del piano e uno solo dall'altra calcoliamo con la stessa formula di prima il volume $V_{\bar{E}}$ del tetraedro formato dal vertice isolato e dalle tre intersezioni (x_5, y_5, z_5) , (x_6, y_6, z_6) , (x_7, y_7, z_7) del piano con gli spigoli del tetraedro. A questo punto il contributo al termine A_{ij} della matrice si può scrivere come

$$\nabla\varphi_i|_E \cdot \nabla\varphi_j|_E (\sigma_1 V_{\bar{E}} + \sigma_2 (V_E - V_{\bar{E}})) \quad (4.6)$$

se il vertice isolato verifica $f > 0$ o scambiando σ_1 e σ_2 se $f < 0$. Rientrano in questo caso anche i tetraedri in cui uno, due o tre dei tre vertici non isolati stanno esattamente sul piano.

- Più complesso risulta il calcolo dell'integrale per i tetraedri che hanno 2 vertici da una parte del piano e due dall'altra. Il tetredro risulta infatti diviso in due solidi con sei vertici. Per calcolare il volume di questi solidi è stato quindi necessario dividerli ognuno in tre tetraedri, per ognuno calcolare il volume con la formula (4.5), sommarli insieme e infine usare una formula analoga alla (4.6).

4.2.1 Verifica del risultato

Per verificare il calcolo del problema diretto ed in particolare della matrice A abbiamo quindi effettuato due simulazioni utilizzando delle mesh cubiche con degli elettrodi estesi che ricoprono interamente due facce opposte del cubo (figura 4.1).

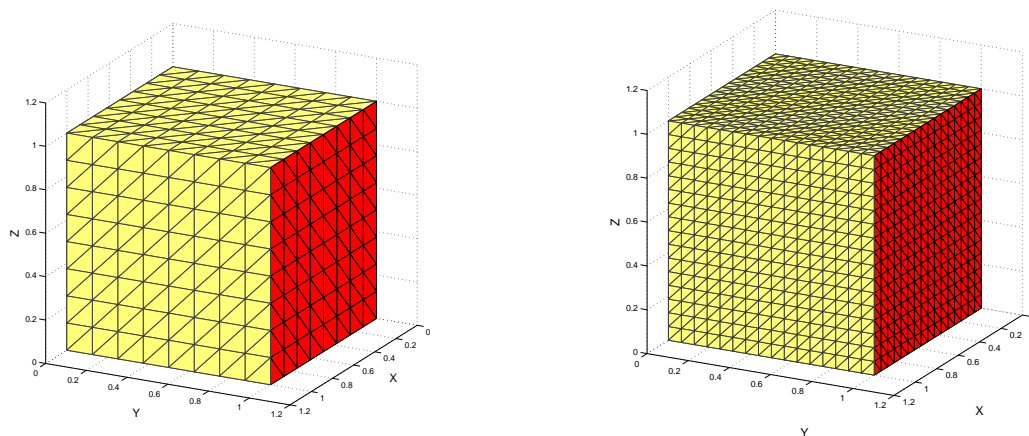


Figura 4.1. Mesh cubiche con gli elettrodi (colorati di rosso) sulle facce $y = 0$ e $y = 1$. Quella a sinistra è composta da 3072 elementi, quella a destra da 24576

Per rendere possibile questa simulazione la matrice appena costruita è stata modificata con funzioni sviluppate dall'ing. Borsic coerentemente con il modello completo definito in 2.5.

I risultati di questa simulazione li abbiamo quindi confrontati con dei risultati teorici; infatti se la conduttività è costante e gli elettrodi ricoprono completamente le facce laterali, le linee del campo elettrico risultano parallele ed è immediato il calcolo analitico della differenza di potenziale su questi elettrodi utilizzando la formula

$$V = RI \quad \text{con} \quad R = z_c + \frac{\rho l}{A} \quad (4.7)$$

dove $\rho = \frac{1}{\sigma}$ è la resistività, l è la distanza tra gli elettrodi, A è l'area degli elettrodi e z_c l'impedenza di contatto totale. Considerando poi un cubo di volume unitario con un vertice nell'origine e gli spigoli paralleli agli assi, degli elettrodi estesi alle facce $y = 0$ e $y = 1$ e un piano $\gamma : y = c$, con $c \in [0,1]$, risulta

$$V = \left(z_1 + z_2 + \frac{1-c}{\sigma_1 A_1} + \frac{c}{\sigma_2 A_2} \right) I \quad (4.8)$$

dove $A_1 = A_2 = 1 \text{ m}^2$.

Nella nostra simulazione abbiamo preso $\sigma_1 = 1 \Omega^{-1} \text{m}^{-1}$ e $\sigma_2 = 0.5 \Omega^{-1} \text{m}^{-1}$, $z_1 = z_2 = 1 \Omega$ e una corrente $I = 1$ Ampere.

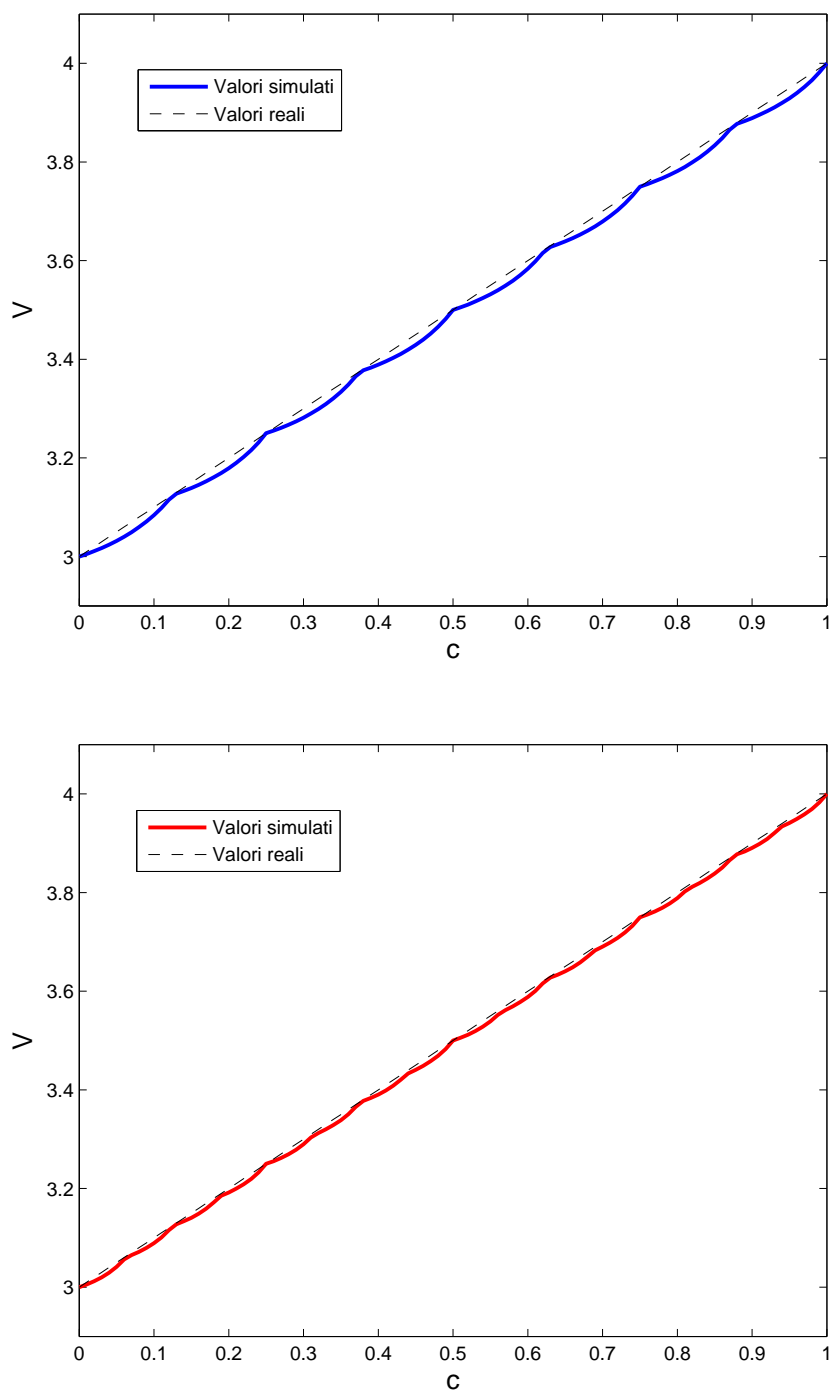


Figura 4.2. Confronto dei valori simulati con i risultati teorici. In alto la simulazione con la mesh di 3072 elementi, in basso con quella più fine di 24576 elementi

Il grafico 4.2 mostra il valore di V al variare di c . Secondo la relazione 4.8 questo dovrebbe variare linearmente tra 3 Volt e 4 Volt. Vediamo che la soluzione si discosta da quella esatta e l'errore è tanto maggiore quanto più il piano è distante dai nodi della mesh. Notiamo inoltre un miglioramento notevole dell'approssimazione quando consideriamo una mesh più fine. Per come è stato modellizzato e poi discretizzato il problema nel capitolo 2 avremo una buona approssimazione di \mathbf{U} solo quando anche la conduttività σ è ben approssimata dalle funzioni di base φ_i . Possiamo infatti osservare che, quando il piano si posiziona in modo tale da non intersecare nessun tetraedro, i nostri risultati coincidono esattamente con quelli teorici.

Per evitare questo problema dovremmo discretizzare il nostro dominio in modo tale che non ci siano mai elementi intersecati dal nostro piano. Tuttavia questo possiamo farlo solo quando dobbiamo risolvere il problema diretto, nel problema inverso le cose sono diverse perché non conosciamo a priori la posizione del piano γ .

Nei prossimi paragrafi vedremo infatti che possiamo trovare γ attraverso un procedimento iterativo in cui ad ogni passo il nostro piano si avvicina sempre di più alla posizione del piano reale. Cambiare la discretizzazione ad ogni passo in modo tale da evitare questo errore di approssimazione sarebbe difficoltoso e comporterebbe un tempo di calcolo molto elevato. Inoltre ci sono dei risultati ([11]) che, nel caso bidimensionale, mostrano come questa inesattezza non influisca sul calcolo di una buona stima dei parametri. Quindi, così come in questa simulazione, anche nei prossimi paragrafi considereremo una discretizzazione fissa anche quando questa porta ad avere degli elementi intersecati dal piano. Per un approfondimento su queste problematiche si rimanda a [3] e [8].

4.3 Calcolo dello Jacobiano

In base a quanto visto nel capitolo precedente, per invertire il nostro problema, che è non-lineare, dobbiamo calcolare lo Jacobiano dell'operatore che, a partire dai parametri, ci dà il valore dei potenziali per un determinato pattern di correnti.

Dobbiamo cioè calcolare

$$J = \frac{\partial \mathbf{U}}{\partial \mathbf{m}} = \begin{pmatrix} \frac{\partial U_1}{\partial a} & \cdots & \frac{\partial U_1}{\partial d} \\ \cdots & \cdots & \cdots \\ \frac{\partial U_k}{\partial a} & \cdots & \frac{\partial U_k}{\partial d} \end{pmatrix} \quad (4.9)$$

Come abbiamo visto nella sezione 3.3, per fare questo è necessario calcolare la derivata della matrice degli elementi finiti rispetto a a , b , c e d .

La funzione che calcola queste derivate è chiamata `dFEMmatrix.m` che al suo interno richiama un'altra funzione chiamata `deriv.m`. Quest'ultima utilizza la classificazione dei tetraedri illustrata in 4.2 e per ognuno calcola le derivate

$$\frac{\partial A_{ij}}{\partial a, b, c, d} = \left(\frac{\partial}{\partial a} \int_{\Omega} \sigma \nabla \varphi_i \cdot \nabla \varphi_j dV; \frac{\partial}{\partial b} \int \cdots; \frac{\partial}{\partial c} \int \cdots; \frac{\partial}{\partial d} \int \cdots \right) \quad (4.10)$$

I tetraedri che non sono toccati dal piano avranno questi termini tutti nulli. Per gli elementi intersecati, a prescindere dal tipo di intersezione, se chiamiamo \bar{E} la parte del tetraedro che sta sopra il piano, in base a quanto detto in 4.2, possiamo sempre considerarla come un tetraedro o come somma di tre tetraedri. Quindi possiamo riprendere l'eq. (4.6) e derivarla ottenendo

$$\frac{\partial}{\partial a, b, c, d} \nabla \varphi_i|_E \cdot \nabla \varphi_j|_E (\sigma_1 V_{\bar{E}} + \sigma_2 (V_E - V_{\bar{E}})) = (\sigma_1 - \sigma_2) \nabla \varphi_i|_E \cdot \nabla \varphi_j|_E \frac{\partial V_{\bar{E}}}{\partial a, b, c, d} \quad (4.11)$$

Infatti solamente il termine $V_{\bar{E}}$ dipende dai parametri del piano mentre tutto il resto si può portare fuori dal segno di derivata. Per semplicità di notazione consideriamo ora solo la derivata rispetto ad a .

Il volume nella formula (4.5) è espresso in funzione delle coordinate dei vertici quindi per calcolare la derivata di $V_{\bar{E}}$ abbiamo applicato la regola di derivazione a catena che ci permette di scrivere

$$\frac{\partial V_{\bar{E}}}{\partial a} = \sum_{i=1}^4 \frac{\partial V_{\bar{E}}}{\partial x_i} \frac{\partial x_i}{\partial a} + \sum_{i=1}^4 \frac{\partial V_{\bar{E}}}{\partial y_i} \frac{\partial y_i}{\partial a} + \sum_{i=1}^4 \frac{\partial V_{\bar{E}}}{\partial z_i} \frac{\partial z_i}{\partial a} \quad (4.12)$$

dove $(x_1, y_1, z_1), \dots, (x_4, y_4, z_4)$ sono i quattro vertici del tetraedro considerato. Questi possono essere o dei nodi della mesh o le intersezioni del piano con gli spigoli dei tetraedri della mesh. Nel primo caso le derivate rispetto ai parametri del piano sono nulle quindi rimangono solo i termini relativi al secondo caso.

I termini $\frac{\partial V_{\bar{E}}}{\partial x,y,z}$ li otteniamo facilmente derivando la formula (4.5) mentre le altre derivate le otteniamo parametrizzando lo spigolo tra il nodo i e j come

$$\begin{cases} x = x_i + t(x_j - x_i) \\ y = y_i + t(y_j - y_i) \\ z = z_i + t(z_j - z_i) \end{cases} \quad \text{con } t \in [0,1] \quad (4.13)$$

Le intersezioni le troviamo imponendo l'equazione del piano, in quei punti si ha

$$t = \frac{ax_i + by_i + cz_i + d}{a(x_i - x_j) + b(y_i - y_j) + c(z_i - z_j)} \quad (4.14)$$

Inserendo questo valore di t nell'eq. (4.13), possiamo derivare rispetto ad a, b, c e d e completare il calcolo dell'eq. (4.12).

Una volta calcolati questi integrali su ogni tetraedro, per costruire l'elemento $\frac{\partial A_{ij}}{\partial a,b,c,d}$, è sufficiente sommare tutti i contributi dei tetraedri appartenenti a $\text{supp}(\varphi_i) \cap \text{supp}(\varphi_j)$ e moltiplicarli per i gradienti che, come abbiamo visto, sono noti. Questa operazione è fatta nella funzione `dFEMmatrix.m`. Nella nostra funzione questo lo realizziamo inserendo, all'interno del ciclo sui tetraedri, due cicli annidati per i e j che variano tra i quattro nodi di ogni elemento.

A questo punto elaborato una funzione chiamata `jacobian.m` che calcola lo jacobiano secondo le formule (3.27) e (3.29).

4.3.1 Verifica del risultato

Come abbiamo visto, la nostra matrice A dipende dalla posizione del piano quindi

$$A = A(a,b,c,d) \quad (4.15)$$

Per verificare il calcolo della derivata della matrice A rispetto ad uno dei parametri, abbiamo costruito, usando la definizione di derivata, una matrice calcolata con il rapporto incrementale di A . Sappiamo infatti che se prendiamo $\delta \in \mathbb{R}$ sufficientemente piccolo

$$\frac{\partial A}{\partial a} \approx \frac{A(a + \delta, b, c, d) - A(a, b, c, d)}{\delta} \quad (4.16)$$

e allo stesso modo possiamo calcolare un'approssimazione di $\frac{\partial A}{\partial b}$, $\frac{\partial A}{\partial c}$ e $\frac{\partial A}{\partial d}$. Quanto più piccolo è δ tanto più l'approssimazione sarà buona. Tuttavia, dal momento che lavoriamo con precisione di calcolo finita, non possiamo prendere valori δ troppo piccoli perché incorreremo nella cancellazione numerica.

Per lo Jacobiano $\frac{\partial \mathbf{U}}{\partial a}$ abbiamo proceduto allo stesso modo confrontandolo con

$$\frac{\mathbf{U}(a + \delta, b, c, d) - \mathbf{U}(a, b, c, d)}{\delta} \quad (4.17)$$

Prendendo

$$\delta \approx 10^{-4} \|\mathbf{m}\|_2 \quad \text{con } \mathbf{m} = (a, b, c, d)^T \quad (4.18)$$

abbiamo quindi verificato che i due risultati coincidono in quanto per ogni elemento otteniamo un errore relativo dell'ordine di 10^{-5} .

4.4 Inversione del problema

Supponiamo di avere a disposizione dei risultati di determinati esperimenti su un oggetto diviso in due parti da un piano γ con conduttività note σ_1 e σ_2 . Nel nostro caso, questi risultati sono delle misure di potenziale che indichiamo con \mathbf{V} .

Con queste informazioni, per risolvere il problema inverso abbiamo sviluppato una funzione chiamata `inverse_solve.m` che con il procedimento iterativo illustrato nella formula (3.12), a partire da una stima iniziale di a , b , c e d , trova le successive stime del piano γ . L'obiettivo è trovare un'approssimazione sufficientemente precisa di \bar{a} , \bar{b} , \bar{c} e \bar{d} che sono i parametri del piano reale.

Nelle applicazioni non è possibile fare un confronto diretto con questi parametri perché non sono noti quindi la stima della distanza del piano stimato da quello reale al passo k è data dal funzionale

$$\phi = \|\mathbf{V} - \mathbf{U}_k\| \quad (4.19)$$

dove \mathbf{U}_k è il vettore che contiene le simulazioni degli esperimenti calcolate tramite il modello diretto usando i parametri del passo k . Possiamo trovare \mathbf{U}_k facilmente risolvendo il problema diretto, in particolare l'eq. da risolvere è la (2.54). Per far questo abbiamo elaborato la funzione `measures.m`.

Per invertire il problema abbiamo quindi utilizzato l'eq. (3.12). Cambiando alcune notazioni, il procedimento iterativo da utilizzare ad ogni passo è quindi

$$\delta \mathbf{P}_k = \mathbf{P}_{k+1} - \mathbf{P}_k = -J_k^\dagger(\mathbf{U}_k - \mathbf{V}) \quad (4.20)$$

dove $\mathbf{P}_k = (a_k, b_k, c_k, d_k)^T$ e $J_k = \frac{\partial \mathbf{U}_k}{\partial a, b, c, d}$.

L'algoritmo si fermerà quando il funzionale da minimizzare raggiunge la tolleranza desiderata. Nelle applicazioni, a causa degli errori di misura, questo potrebbe non avere un minimo nello zero quindi potremmo non raggiungere mai la tolleranza desiderata. Per questo abbiamo anche introdotto un numero massimo di iterazioni dopo le quali il programma si arresta.

In tutte le nostre prove e simulazioni invece siamo sicuri che il funzionale deve convergere a zero perchè anche \mathbf{V} è calcolato tramite il modello diretto. In questo modo possiamo inoltre fare ad ogni passo un confronto diretto con i parametri \bar{a} , \bar{b} , \bar{c} e \bar{d} e verificare la correttezza del risultato.

Nell'algoritmo partiamo quindi da quattro valori a_0 , b_0 , c_0 e d_0 che individuano il piano iniziale e che possiamo introdurre come input della funzione. Nelle nostre simulazioni abbiamo sempre preso dei valori casuali. Al primo passo ($k = 0$), come a tutti gli altri k , dobbiamo quindi calcolare lo jacobiano (con i parametri \mathbf{P}_k) usando le funzioni viste nella sezione precedente.

A questo punto è stato però necessario regolarizzare il problema. Utilizzando infatti quattro parametri abbiamo infinite rappresentazioni di ogni piano, potremmo cioè moltiplicare a , b , c e d per una costante qualsiasi e otterremmo lo stesso piano. Questo comporta che lo jacobiano risulta non avere rango massimo, abbiamo cioè un numero di condizionamento che tende all'infinito. Risulta quindi impossibile il calcolo di J^\dagger .

Per evitare questo problema abbiamo pensato inizialmente di tentare una ricostruzione con tre soli parametri tenendo costante il parametro $c = -1$. Per fare questo, se escludiamo la rappresentazione dei piani paralleli all'asse z (per cui avremmo $c = 0$), basta dividere tutti i parametri per $(-c)$.

Successivamente, analizzando lo jacobiano con la decomposizione SVD, abbiamo potuto risolvere diversamente questo problema. Troviamo infatti che un valore singolare risulta sempre nullo, possiamo quindi usare la decomposizione troncata

prendendo solamente i primi tre elementi della sommatoria. La matrice jacobiana risulta avere infatti in tutto quattro valori singolari perché ha quattro colonne e un numero di righe pari al numero di esperimenti. Per poter invertire il problema questi devono essere sicuramente almeno quattro.

L'equazione da risolvere ad ogni passo diventa quindi:

$$\delta \mathbf{P}_k = \mathbf{P}_{k+1} - \mathbf{P}_k = \sum_{i=1}^3 \frac{\mathbf{u}_i^T (\mathbf{U}_k - \mathbf{V})}{\lambda_i} \mathbf{v}_i \quad (4.21)$$

dove \mathbf{v}_i e \mathbf{u}_i sono i vettori singolari destri e sinistri e λ_i sono i valori singolari.

Normalmente con questo algoritmo non si incontrano più problemi, tuttavia possiamo migliorare ulteriormente l'algoritmo. Infatti in alcune posizioni particolari si ottengono dei valori $\delta \mathbf{P}_k$ troppo piccoli o troppo grandi. Nel primo caso l'algoritmo tende a convergere verso un punto \mathbf{P}_* in cui ϕ assume valori ancora lontani dallo zero.¹ Possiamo ipotizzare quindi che \mathbf{P}_* sia un punto di minimo relativo per il funzionale ϕ e l'algoritmo non riesce a uscire dal bacino di attrazione di questi punti. Nel secondo caso invece può succedere che l'algoritmo arrivi ad un piano esterno al nostro oggetto e in questo caso non è più possibile calcolare un'approssimazione successiva perché lo jacobiano risulterebbe nullo.

In entrambi i casi è quindi necessario modificare $\delta \mathbf{P}_k$. Dal momento che, rappresentando il piano con quattro parametri, possiamo moltiplicare \mathbf{P}_k per una costante qualsiasi, abbiamo pensato di normalizzare ad ogni ciclo il vettore \mathbf{P}_k dividendolo per la sua norma. Quando ci accorgiamo che il metodo iterativo provocherebbe un $\delta \mathbf{P}_k$ troppo grande rischiando di far uscire completamente il piano dal solido, ridimensioniamo $\delta \mathbf{P}_k$ in modo tale che abbia una norma pari a 0.5. In pratica utilizziamo una sorta di "line search" in cui manteniamo la direzione di $\delta \mathbf{P}_k$ ma inseriamo un coefficiente $\theta > 0$ opportuno ottenendo

$$\mathbf{P}_{k+1} = \mathbf{P}_k + \theta(\delta \mathbf{P}_k) \quad (4.22)$$

Abbiamo scelto di prendere $\theta = \frac{1}{2\|\delta \mathbf{P}_k\|_2}$ quando $\|\delta \mathbf{P}_k\|_2 > 1$, altrimenti $\theta = 1$.

¹Dal momento che nelle nostre prove conosciamo già a priori $\bar{\mathbf{P}}$, cioè i parametri del piano reale, possiamo anche verificare che $\mathbf{P}_* \neq \bar{\mathbf{P}}$.

Per risolvere l'altro problema invece abbiamo deciso di intervenire quando ϕ diminuisce troppo lentamente (meno del 10%), cioè quando ci accorgiamo che non sta convergendo verso lo zero.² In questo caso, oltre a ridimensionare $\delta\mathbf{P}_k$, sommiamo ad ogni sua componente un numero casuale compreso tra -1 e 1 .

Nonostante l'esistenza di minimi locali sia da approfondire, con queste modifiche abbiamo sempre constatato, in tutte le simulazioni effettuate, la convergenza verso la soluzione esatta.

L'algoritmo appena illustrato lo utilizzeremo per il test conclusivo nella prossima sezione ed è schematizzato in figura 4.4.

```

P0 = P0/norm(P0)
for k=1 to max_iterazioni
    Uk = U(Pk)
    zk=norm(Uk - V)
    if z<Tolleranza return
    svd(J)
    i=1
     $\delta\mathbf{P} = (0,0,0,0)^T$ 
    while  $\lambda_i > 10^{-3}$  and i<4
         $\delta\mathbf{P} = \delta\mathbf{P} - \frac{\mathbf{u}_i^T(\mathbf{U}_k - \mathbf{V})}{\lambda_i} \mathbf{v}_i$ 
        i=i+1
    end while
    if (norm( $\delta\mathbf{P}$ )>1)
         $\delta\mathbf{P} = \delta\mathbf{P}/(2\text{norm}(\delta\mathbf{P}))$ 
    end if
    if (|zk - zk-1|/zk<0.1)
         $\delta\mathbf{P} = \delta\mathbf{P}/(2\text{norm}(\delta\mathbf{P})) + (\text{rand}, \text{rand}, \text{rand}, \text{rand})^T$ 
    end if
    Pk+1 = Pk +  $\delta\mathbf{P}$ 
    Pk+1 = Pk+1/norm(Pk+1)

```

Figura 4.3. Algoritmo utilizzato per l'inversione del problema.

Per rappresentare graficamente il piano ai vari passi dell'algoritmo, abbiamo poi costruito la funzione `diseгна.m` che calcola quattro punti che appartengono al piano e rappresenta la superficie tra questi punti.

²Nelle applicazioni, come abbiamo visto, il minimo di θ potrebbe non essere 0 quindi questo criterio dovrebbe essere modificato.

Per poter stimare il piano utilizzando quattro parametri abbiamo anche effettuato delle simulazioni applicando l'eq. (3.32) in cui la funzione da minimizzare è

$$\tilde{\phi} = \|\mathbf{V} - \mathbf{U}_k\| + (a + b + c + d - 1)^2 \quad (4.23)$$

Con gli stessi passaggi che ci hanno portato a trovare l'eq. (3.12), abbiamo trovato un nuovo procedimento iterativo che ci permette di minimizzare il nuovo funzionale

$$\delta \mathbf{P}_k = \mathbf{P}_{k+1} - \mathbf{P}_k = -(J_k^T J_k + \alpha I)^{-1} [J_k^T (\mathbf{U}_k - \mathbf{V}) - \alpha (I \mathbf{P}_k - 1)] \quad (4.24)$$

Abbiamo notato però che, rispetto alla decomposizione troncata, questo metodo ci portava ad un algoritmo che dava luogo ad una convergenza più lenta e che incorreva molto più spesso in variazioni troppo elevate. Per questo motivo, nel test conclusivo abbiamo deciso di utilizzare il metodo della decomposizione troncata.

4.5 Test conclusivo

Dopo aver sviluppato queste funzioni, per verificare tutto il nostro lavoro abbiamo eseguito delle simulazioni in cui abbiamo trovato la soluzione di particolari problemi inversi. Abbiamo deciso quindi di riportare in questa tesi i risultati di una di queste simulazioni.

Abbiamo usato una mesh cilindrica con 6120 elementi e 24 elettrodi puntiformi senza impedenza di contatto sparsi sulle superfici laterali e di base, rappresentata in figura 4.4.

Per poter invertire il problema abbiamo constatato l'importanza di avere una serie di esperimenti che permetta di rilevare tutti le possibili posizioni del piano. Aumentando però troppo il numero di esperimenti aumenta il tempo di calcolo, è importante quindi scegliere un buono schema di misurazioni.

Nella simulazione qua riportata abbiamo effettuato 14 esperimenti ognuno dei quali simula l'iniezione di una corrente unitaria in una coppia di elettrodi. 8 di questi utilizzano coppie di elettrodi sulle facce, gli altri 6 invece utilizzano gli elettrodi laterali. I vettori \mathbf{V} e \mathbf{U} conterranno quindi 14 elementi.

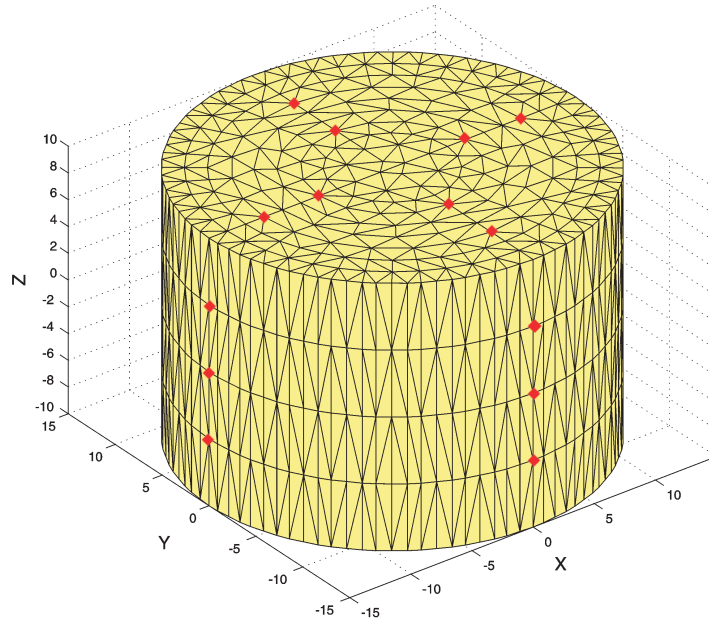


Figura 4.4. Rappresentazione della mesh cilindrica utilizzata per il test conclusivo. Gli elettrodi utilizzati sono stati rappresentati in rosso.

Abbiamo calcolato, con il nostro modello diretto, dei valori di potenziale per una determinata posizione del piano. Quindi, partendo da un piano qualsiasi abbiamo visto come, in pochi passi, l'algoritmo ci porta a ritrovare gli stessi parametri usati per calcolare i potenziali.

Infine abbiamo rappresentato i vari passi dell'algoritmo in fig. 4.5 e riportato i risultati nella tabella 4.1.

Passo k	a_k	b_k	c_k	d_k	$\ \mathbf{V} - \mathbf{U}_k\ _2$
0	0.1525	0.1525	0.7625	0.6100	0.4768
1	0.4545	0.5129	0.5971	0.4170	0.1647
2	0.5438	0.5497	0.4665	0.4296	0.0553
3	0.5830	0.5879	0.2981	0.4750	0.0369
4	0.6827	0.7160	0.0289	-0.1433	0.0163
5	0.7526	0.5870	-0.2028	-0.2191	0.0046
6	0.7808	0.5877	-0.1964	0.0797	0.0003

Tabella 4.1. Valori dei parametri e della funzione obiettivo per i primi passi dell'algoritmo. L'ultimo passo ha raggiunto la tolleranza desiderata e nei passi successivi, non si ottengono più variazioni significative dei parametri.

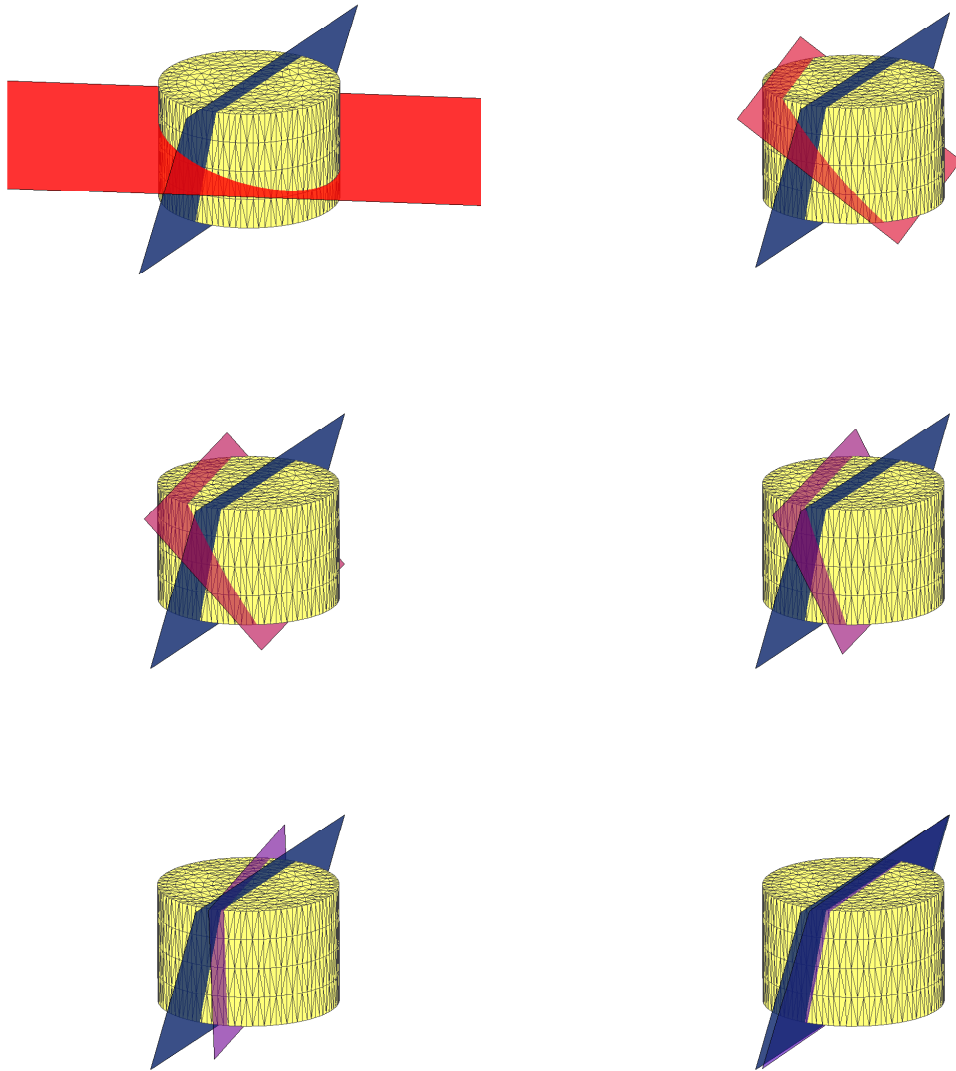


Figura 4.5. Rappresentazione del piano per i primi sei passi dell’algoritmo. In rosso è rappresentato la stima iniziale del piano, in blu la posizione reale del piano.

Dalle immagini possiamo effettivamente vedere come il piano iniziale (in rosso) tenda ad avvicinarsi sempre di più al piano reale (blu) fino a sovrapporsi ad esso.

Dalla tabella possiamo invece notare come la quantità $\|\mathbf{V} - \mathbf{U}\|$, che rappresenta il funzionale da minimizzare, diminuisca sempre fino ad arrivare alla tolleranza

desiderata. Nella nostra simulazione abbiamo impostato una tolleranza di 10^{-3} .

Capitolo 5

Conclusioni e possibili sviluppi

In questa tesi abbiamo cercato di riportare i risultati di questa collaborazione che, come abbiamo visto, ci ha portato a raggiungere gli obiettivi che ci eravamo prefissi.

Come abbiamo detto, sono numerosi in questi anni gli studi sulla tomografia elettrica e sulla ricostruzione di immagini e questo lavoro, pur avendo risolto un problema piuttosto specifico, ci ha permesso di comprendere le problematiche e i metodi matematici di questo settore di ricerca e ci permette ora di analizzare dei possibili sviluppi a partire da quanto fatto in questa tesi.

In molte applicazioni (basti pensare al campo medico) ci troviamo di fronte a situazioni in cui ci sono variazioni elevate di conduttività e dove le superfici che dividono queste regioni a conduttività diverse non sono dei piani ma possono essere delle superfici più complesse (come in figura 5.1). Per stimare il bordo di questo tipo di superfici è possibile approssimarle con delle superfici parametrizzate. Queste superfici, per esempio delle funzioni splines su \mathbb{R}^3 , possiamo poi approssimarle su ogni singolo elemento della discretizzazione come funzioni lineari cioè come piani e applicare quindi un procedimento analogo a quello affrontato in questa tesi.

Questo tipo di approccio, usato per ricostruire immagini e stimare il bordo di regioni bidimensionali in [5] e [12], è stato il punto di partenza per tutto il nostro lavoro.

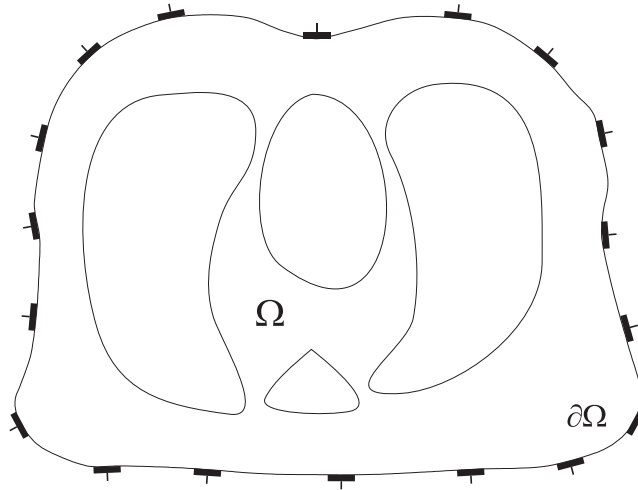


Figura 5.1. Esempio di EIT in cui le regioni da ricostruire sono interne al corpo. In questo esempio è rappresentata la sezione del torace umano, e sono evidenziati polmoni, cuore e spina dorsale, organi con una conduttività diversa rispetto al tessuto circostante.

Inoltre nel nostro problema abbiamo sempre supposto noti i due valori di conduttività σ_1 e σ_2 . In realtà è possibile inserire queste grandezze tra i parametri da stimare usando lo stesso procedimento di inversione del problema e calcolando le derivate delle misure rispetto a σ .

Un altro possibile sviluppo è l'applicazione dei metodi visti in questa tesi per problemi analoghi ma in campi diversi come nei problemi di meccanica. In questo caso dovremmo considerare un modello matematico diverso, quindi nella risoluzione numerica con il metodo agli elementi finiti comparirebbero integrali che risultano più difficile da calcolare.

Nei nostri calcoli infatti compaiono solamente integrali del tipo

$$\int_{\Omega} \sigma \nabla \varphi_i \cdot \nabla \varphi_j dV \quad (5.1)$$

dove le funzioni φ_i e φ_j sono funzioni lineari quindi il termine sotto il segno di integrale risulta costante su ogni elemento.

Per problemi di natura elastica invece compaiono spesso integrali del tipo

$$\int_{\Omega} \sigma \varphi_i \varphi_j dV \quad (5.2)$$

che hanno sotto il segno di integrale una funzione quadratica. Risulta quindi più

difficile il calcolo dello jacobiano dell'operatore diretto perché dovremmo calcolare le derivate di questi integrali rispetto ai parametri. Tuttavia, superate queste difficoltà di calcolo, possiamo pensare di usare un approccio analogo al nostro per risolvere il problema inverso.

Appendice A

Listati dei programmi

In questo capitolo abbiamo riportato i programmi sviluppati in questa collaborazione che ci hanno permesso di effettuare le simulazioni del capitolo 4.

In tutti i programmi riportati in questa appendice viene dato come input la struttura `mesh` che è costituito dei seguenti campi:

- `vtx_f`: matrice che contiene le coordinate dei nodi della mesh
- `elem_f`: matrice che, per ogni riga k contiene l'indice dei nodi che compongono il k -esimo tetraedro
- `gx`, `gy`, `gz`: matrici che, al posto (i,j) , contengono il valore all'interno dell'elemento i , della derivata della funzione di base. Rispettivamente $\frac{\partial \varphi_j}{\partial x}$, $\frac{\partial \varphi_j}{\partial y}$, $\frac{\partial \varphi_j}{\partial z}$
- `pE`: array che contiene l'elenco degli elettrodi puntiformi identificati con il numero del nodo corrispondente
- `fE`: array che contiene l'elenco degli elettrodi estesi identificati attraverso i nodi che stanno su di essi. Per ogni elettrodo è specificata anche l'impedenza di contatto.
- `vols`: array contenente il volume di ogni elemento della mesh

- `u2v`: array che contiene una base per calcolare il risultato per un qualunque pattern di correnti, considerandolo come combinazione lineare di elementi della base.

La matrice `meas_spec` serve invece per definire gli elettrodi usati e le correnti iniettate. Per ogni riga, che corrisponde ad un esperimento, sono specificati due elettrodi per il calcolo del potenziale, due elettrodi per il passaggio della corrente e il valore della corrente iniettata.

È stato necessario scrivere anche una funzione che calcola il volume di un tetraedro a partire dalle coordinate dei vertici secondo la formula (4.5). Questa funzione, chiamata `voltet.m` viene richiamata più volte in tutte le altre funzioni.

A.1 inters.m

```
function [x]=inters(msh,a,b,c,d)
% Calcola per ogni elemento della mesh il tipo di intersezione col piano
% INPUT msh=struttura mesh; a,b,c,d=parametri del piano
% OUTPUT x=[n°vert pos,n°vert nulli,n°vert neg]
vtx=msh.vtx_f;
elem=msh.elem_f;
for i= 1:size(elem,1)
    f(1)=a*vtx(elem(i,1),1)+b*vtx(elem(i,1),2)+c*vtx(elem(i,1),3)+d;
    f(2)=a*vtx(elem(i,2),1)+b*vtx(elem(i,2),2)+c*vtx(elem(i,2),3)+d;
    f(3)=a*vtx(elem(i,3),1)+b*vtx(elem(i,3),2)+c*vtx(elem(i,3),3)+d;
    f(4)=a*vtx(elem(i,4),1)+b*vtx(elem(i,4),2)+c*vtx(elem(i,4),3)+d;
    pos=0;
    neg=0;
    for k=1:4
        if f(k)>0
            pos=pos+1;
        elseif f(k)<0
            neg=neg+1;
        end
    end
    x(i,:)=[pos,(4-pos-neg),neg];
end
```

A.2 FEMmatrix.m

```
function [AA]=FEMmatrix(msh,a,b,c,d,s1,s2)
% Calcola l'integrale del prodotto dei gradienti delle funzioni di base i e j
% moltiplicato per la proprieta s2(se ax+by+cz+d<0) o s1(altrimenti)
```

```

% INPUT msh=struttura mesh; a,b,c,d=parametri del piano; s1,s2=conduttività
% OUTPUT AA=matrice FEM
int=inters(msh,a,b,c,d);
vtx=msh.vtx_f;
elem=msh.elem_f;
AA=spalloc(size(vtx,1),size(vtx,1),ceil(0.01*size(vtx,1)^2));
for k=1:size(elem,1)
    vol=msh.vols(k);
    p=[vtx(elem(k,1),1),vtx(elem(k,1),2),vtx(elem(k,1),3);
        vtx(elem(k,2),1),vtx(elem(k,2),2),vtx(elem(k,2),3);
        vtx(elem(k,3),1),vtx(elem(k,3),2),vtx(elem(k,3),3);
        vtx(elem(k,4),1),vtx(elem(k,4),2),vtx(elem(k,4),3)];
    cost=[a;b;c];
    for r=1:4
        f(r)=p(r,1:3)*cost+d;
    end
    for i=1:4
        for j=1:4
            gi=[msh.gx(k,elem(k,i)),msh.gy(k,elem(k,i)),msh.gz(k,elem(k,i))];
            gj=[msh.gx(k,elem(k,j)),msh.gy(k,elem(k,j)),msh.gz(k,elem(k,j))];
            if int(k,1)==0
                AA(elem(k,i),elem(k,j))=AA(elem(k,i),elem(k,j))+
                    +(s2*vol*(gi*gj'));
            elseif int(k,3)==0
                AA(elem(k,i),elem(k,j))=AA(elem(k,i),elem(k,j))+
                    +(s1*vol*(gi*gj'));
            elseif int(k,3)==1
                for h=1:4
                    if f(h)<0 %trova il vertice da solo da una parte del piano
                        n=h;
                    end
                end
                x(1)=p(n,1);
                y(1)=p(n,2);
                z(1)=p(n,3);
                w=2;
                for h=1:4
                    if h~=n
                        t=(p(n,1:3)*cost+d)/((p(n,1:3)*cost)-(p(h,1:3)*cost));
                        x(w)=p(n,1)-t*(p(n,1)-p(h,1));
                        y(w)=p(n,2)-t*(p(n,2)-p(h,2));
                        z(w)=p(n,3)-t*(p(n,3)-p(h,3));
                        w=w+1;
                    end
                end
                vol1=voltet(0,[x',y',z']);
                vol2=vol-vol1;
                AA(elem(k,i),elem(k,j))=AA(elem(k,i),elem(k,j))+
                    +(s1*vol2*(gi*gj'))+(s2*vol1*(gi*gj'));
            end
        end
    end
end

```

```

elseif int(k,1)==1
    for h=1:4
        if f(h)>0 %trova il vertice da solo da una parte del piano
            n=h;
        end
    end
    x(1)=p(n,1);
    y(1)=p(n,2);
    z(1)=p(n,3);
    w=2;
    for h=1:4
        if h~=n
            t=(p(n,1:3)*cost+d)/((p(n,1:3)*cost)-(p(h,1:3)*cost));
            x(w)=p(n,1)-t*(p(n,1)-p(h,1));
            y(w)=p(n,2)-t*(p(n,2)-p(h,2));
            z(w)=p(n,3)-t*(p(n,3)-p(h,3));
            w=w+1;
        end
    end
    vol1=voltet(0,[x',y',z']);
    vol2=vol-vol1;
    AA(elem(k,i),elem(k,j))=AA(elem(k,i),elem(k,j))+
        +(s2*vol2*(gi*gj'))+(s1*vol1*(gi*gj'));
elseif int(k,:)==[2,0,2]
    w=1;
    ww=4;
    for www=1:4
        if f(www)<0
            n(w)=www;
            w=w+1; %n1 e n2 vertici negativi; n3 e n4 positivi
        else
            n(ww)=www;
            ww=ww-1;
        end
    end
    x(1)=p(n(1),1); %primo tetraedro
    y(1)=p(n(1),2);
    z(1)=p(n(1),3);
    x(2)=p(n(2),1);
    y(2)=p(n(2),2);
    z(2)=p(n(2),3);
    t=(p(n(2),1:3)*cost+d)/((p(n(2),1:3)*cost)-(p(n(3),1:3)*cost));
    x(3)=p(n(2),1)-t*(p(n(2),1)-p(n(3),1));
    y(3)=p(n(2),2)-t*(p(n(2),2)-p(n(3),2));
    z(3)=p(n(2),3)-t*(p(n(2),3)-p(n(3),3));
    t=(p(n(2),1:3)*cost+d)/((p(n(2),1:3)*cost)-(p(n(4),1:3)*cost));
    x(4)=p(n(2),1)-t*(p(n(2),1)-p(n(4),1));
    y(4)=p(n(2),2)-t*(p(n(2),2)-p(n(4),2));
    z(4)=p(n(2),3)-t*(p(n(2),3)-p(n(4),3));

```

```

    vol1=voltet(0,[x',y',z']);
    t=(p(n(1),1:3)*cost+d)/((p(n(1),1:3)*cost)-(p(n(3),1:3)*cost));
    x(2)=p(n(1),1)-t*(p(n(1),1)-p(n(3),1)); %secondo tetraedro
    y(2)=p(n(1),2)-t*(p(n(1),2)-p(n(3),2));
    z(2)=p(n(1),3)-t*(p(n(1),3)-p(n(3),3));
    vol1=vol1+voltet(0,[x',y',z']);
    t=(p(n(1),1:3)*cost+d)/((p(n(1),1:3)*cost)-(p(n(4),1:3)*cost));
    x(3)=p(n(1),1)-t*(p(n(1),1)-p(n(4),1)); %terzo tetraedro
    y(3)=p(n(1),2)-t*(p(n(1),2)-p(n(4),2));
    z(3)=p(n(1),3)-t*(p(n(1),3)-p(n(4),3));
    vol1=vol1+voltet(0,[x',y',z']);
    vol2=vol-vol1;
    AA(elem(k,i),elem(k,j))=AA(elem(k,i),elem(k,j))+
        +(s1*vol2*(gi*gj'))+(s2*vol1*(gi*gj'));
    end
end
end
end
end

```

A.3 measures.m

```

function V=measures(msh,a,b,c,d,s1,s2,meas_spec)
% Calcola le misure di potenziale
% INPUT: msh=struttura mesh; a,b,c,d=parametri del piano
% s1,s2=conduttività
% meas_spec=tipo di misurazione, in ogni riga [ V+ V- I+ I- I ]
% OUTPUT: V=vettore delle ddp tra V+ e V-
A=FEMmatrix(msh,a,b,c,d,s1,s2);
num_meas=size(meas_spec,1);
num_nodes=size(A,1);
A(msh.gnd,:)=0;
A(:,msh.gnd)=0; %specifica un nodo di terra a pot nullo
A(msh.gnd,msh.gnd)=1;
for i=1:num_meas
    bb=zeros(num_nodes,1);
    bb(msh.pE{meas_spec(i,3),1}(1,2))=meas_spec(i,5);
    bb(msh.pE{meas_spec(i,4),1}(1,2))=-meas_spec(i,5);
    bb(msh.gnd)=0;
    [uu,flag]=pcg(A,bb,1e-6,500);
    V(i)=uu(msh.pE{meas_spec(i,1),1}(1,2))-uu(msh.pE{meas_spec(i,2),1}(1,2));
end
V=V';

```

A.4 deriv.m

```

function [dv]=deriv(msh,a,b,c,d,s1,s2)
%Calcola la derivata dei volumi dei tetraedri rispetto al coefficiente a,b,c,d

```

```

% moltiplicata per (s1-s2)
% INPUT msh=struttura mesh; a,b,c,d=parametri del piano; s1,s2=conduttività
% OUTPUT dv=matrice (n.elem,4) [d/da,d/db,d/dc,d/dd]
int=inters(msh,a,b,c,d);
dv=zeros(size(msh.elem_f,1),4);
sigma=s1-s2;
vtx=msh.vtx_f;
elem=msh.elem_f;
for m=1:size(elem,1)
    vol=voltet(m);
    p=[vtx(elem(m,1),1),vtx(elem(m,1),2),vtx(elem(m,1),3);
        vtx(elem(m,2),1),vtx(elem(m,2),2),vtx(elem(m,2),3);
        vtx(elem(m,3),1),vtx(elem(m,3),2),vtx(elem(m,3),3);
        vtx(elem(m,4),1),vtx(elem(m,4),2),vtx(elem(m,4),3)];
    cost=[a;b;c];
    for k=1:4
        f(k)=p(k,1:3)*cost+d;
    end
    if int(m,1)==1 %caso con un vertice da una parte e 3 dall'altra
        for k=1:4
            if f(k)>0 %trova il vertice che sta da solo da una parte del piano
                n=k;
            end
        end
        end
        x(1)=p(n,1);
        dx(1,1:4)=0;
        y(1)=p(n,2);
        dy(1,1:4)=0;
        z(1)=p(n,3);
        dz(1,1:4)=0;
        i=2;
        for k=1:4
            if k~=n
                t=(p(n,1:3)*cost+d)/((p(n,1:3)*cost)-(p(k,1:3)*cost));
                x(i)=p(n,1)-t*(p(n,1)-p(k,1));
                dx(i,1)=((p(n,1)-p(k,1))*(b*(p(k,1)*p(n,2)-p(n,1)*p(k,2))+
                    c*(p(k,1)*p(n,3)-p(n,1)*p(k,3))+d*(p(k,1)-p(n,1))))
                    /(a*(p(k,1)-p(n,1))+b*(p(k,2)-p(n,2))+c*(p(k,3)-p(n,3)))^2);
                dx(i,2)=((p(n,1)-p(k,1))*(a*(p(k,2)*p(n,1)-p(n,2)*p(k,1))+
                    c*(p(k,2)*p(n,3)-p(n,2)*p(k,3))+d*(p(k,2)-p(n,2))))
                    /(a*(p(k,1)-p(n,1))+b*(p(k,2)-p(n,2))+c*(p(k,3)-p(n,3)))^2);
                dx(i,3)=((p(n,1)-p(k,1))*(a*(p(k,3)*p(n,1)-p(n,3)*p(k,1))+
                    b*(p(k,3)*p(n,2)-p(n,3)*p(k,2))+d*(p(k,3)-p(n,3))))
                    /(a*(p(k,1)-p(n,1))+b*(p(k,2)-p(n,2))+c*(p(k,3)-p(n,3)))^2);
                dx(i,4)=((p(n,1)-p(k,1))/(a*(p(k,1)-p(n,1))+b*(p(k,2)-p(n,2))+c*(p(k,3)-p(n,3))));
                y(i)=p(n,2)-t*(p(n,2)-p(k,2));
                dy(i,1)=((p(n,2)-p(k,2))*(b*(p(k,1)*p(n,2)-p(n,1)*p(k,2))+
                    c*(p(k,1)*p(n,3)-p(n,1)*p(k,3))+d*(p(k,1)-p(n,1))))
                    /(a*(p(k,1)-p(n,1))+b*(p(k,2)-p(n,2))+c*(p(k,3)-p(n,3)))^2);
            end
        end
    end
end

```



```

dy(i,2)=((p(n,2)-p(k,2))*(a*(p(k,2)*p(n,1)-p(n,2)*p(k,1))+
          c*(p(k,2)*p(n,3)-p(n,2)*p(k,3))+d*(p(k,2)-p(n,2)))/
          (a*(p(k,1)-p(n,1))+b*(p(k,2)-p(n,2))+c*(p(k,3)-p(n,3)))^2);
dy(i,3)=((p(n,2)-p(k,2))*(a*(p(k,3)*p(n,1)-p(n,3)*p(k,1))+
          b*(p(k,3)*p(n,2)-p(n,3)*p(k,2))+d*(p(k,3)-p(n,3)))/
          (a*(p(k,1)-p(n,1))+b*(p(k,2)-p(n,2))+c*(p(k,3)-p(n,3)))^2);
dy(i,4)=((p(n,2)-p(k,2))/(a*(p(k,1)-p(n,1))+b*(p(k,2)-p(n,2))+c*(p(k,3)-p(n,3))));
z(i)=p(n,3)-t*(p(n,3)-p(k,3));
dz(i,1)=((p(n,3)-p(k,3))*(b*(p(k,1)*p(n,2)-p(n,1)*p(k,2))+
          c*(p(k,1)*p(n,3)-p(n,1)*p(k,3))+d*(p(k,1)-p(n,1)))/
          (a*(p(k,1)-p(n,1))+b*(p(k,2)-p(n,2))+c*(p(k,3)-p(n,3)))^2);
dz(i,2)=((p(n,3)-p(k,3))*(a*(p(k,2)*p(n,1)-p(n,2)*p(k,1))+
          c*(p(k,2)*p(n,3)-p(n,2)*p(k,3))+d*(p(k,2)-p(n,2)))/
          (a*(p(k,1)-p(n,1))+b*(p(k,2)-p(n,2))+c*(p(k,3)-p(n,3)))^2);
dz(i,3)=((p(n,3)-p(k,3))*(a*(p(k,3)*p(n,1)-p(n,3)*p(k,1))+
          b*(p(k,3)*p(n,2)-p(n,3)*p(k,2))+d*(p(k,3)-p(n,3)))/
          (a*(p(k,1)-p(n,1))+b*(p(k,2)-p(n,2))+c*(p(k,3)-p(n,3)))^2);
dz(i,4)=((p(n,3)-p(k,3))/(a*(p(k,1)-p(n,1))+b*(p(k,2)-p(n,2))+c*(p(k,3)-p(n,3))));
i=i+1;
end
end
diffx=[-(y(2)*(z(3)-z(4))+y(3)*(z(4)-z(2))+y(4)*(z(2)-z(3))),
        (y(1)*(z(3)-z(4))+y(3)*(z(4)-z(1))+y(4)*(z(1)-z(3))),
        -(y(1)*(z(2)-z(4))+y(2)*(z(4)-z(1))+y(4)*(z(1)-z(2))),
        (y(1)*(z(2)-z(3))+y(2)*(z(3)-z(1))+y(3)*(z(1)-z(2)))];
diffy=[(x(2)*(z(3)-z(4))+x(3)*(z(4)-z(2))+x(4)*(z(2)-z(3))),
        -(x(1)*(z(3)-z(4))+x(3)*(z(4)-z(1))+x(4)*(z(1)-z(3))),
        (x(1)*(z(2)-z(4))+x(2)*(z(4)-z(1))+x(4)*(z(1)-z(2))),
        -(x(1)*(z(2)-z(3))+x(2)*(z(3)-z(1))+x(3)*(z(1)-z(2)))];
diffz=[-(x(2)*(y(3)-y(4))+x(3)*(y(4)-y(2))+x(4)*(y(2)-y(3))),
        (x(1)*(y(3)-y(4))+x(3)*(y(4)-y(1))+x(4)*(y(1)-y(3))),
        -(x(1)*(y(2)-y(4))+x(2)*(y(4)-y(1))+x(4)*(y(1)-y(2))),
        (x(1)*(y(2)-y(3))+x(2)*(y(3)-y(1))+x(3)*(y(1)-y(2)))];
dv(m,1:4)=sigma*sign(vol)*(diffx*dx+diffy*dy+diffz*dz)/6;
elseif int(m,3)==1 %caso con un vertice da una parte e 3 dall'altra
for k=1:4
if f(k)<0 %trova il vertice che sta da solo da una parte del piano
n=k;
end
end
end
x(1)=p(n,1);
dx(1,1:4)=0;
y(1)=p(n,2);
dy(1,1:4)=0;
z(1)=p(n,3);
dz(1,1:4)=0;
i=2;
for k=1:4
if k~=n

```

```

t=(p(n,1:3)*cost+d)/((p(n,1:3)*cost)-(p(k,1:3)*cost));
x(i)=p(n,1)-t*(p(n,1)-p(k,1));
dx(i,1)=... %come sopra
dx(i,2)=... %come sopra
dx(i,3)=... %come sopra
dx(i,4)=... %come sopra
y(i)=p(n,2)-t*(p(n,2)-p(k,2));
dy(i,1)=... %come sopra
dy(i,2)=... %come sopra
dy(i,3)=... %come sopra
dy(i,4)=... %come sopra
z(i)=p(n,3)-t*(p(n,3)-p(k,3));
dz(i,1)=... %come sopra
dz(i,2)=... %come sopra
dz(i,3)=... %come sopra
dz(i,4)=... %come sopra
i=i+1;
end
end
diffx=... %come sopra
diffy=... %come sopra
diffz=... %come sopra
dv(m,1:4)=-sigma*sign(vol)*(diffx*dx+diffy*dy+diffz*dz)/6;
elseif int(m,1)==2 % caso 2 vertici positivi
i=1;
j=4;
for k=1:4
if f(k)>0
n(i)=k;
i=i+1; %n1 e n2 vertici positivi
else
n(j)=k;
j=j-1;
end
end
end
x(1)=p(n(1),1); %primo tetraedro
y(1)=p(n(1),2);
z(1)=p(n(1),3);
x(2)=p(n(2),1);
y(2)=p(n(2),2);
z(2)=p(n(2),3);
dx(1,1:4)=0;
dy(1,1:4)=0;
dz(1,1:4)=0;
dx(2,1:4)=0;
dy(2,1:4)=0;
dz(2,1:4)=0;
t=(p(n(2),1:3)*cost+d)/((p(n(2),1:3)*cost)-(p(n(3),1:3)*cost));
x(3)=p(n(2),1)-t*(p(n(2),1)-p(n(3),1));

```

```

y(3)=p(n(2),2)-t*(p(n(2),2)-p(n(3),2));
z(3)=p(n(2),3)-t*(p(n(2),3)-p(n(3),3));
dx(3,1)=... %come sopra i=3
dx(3,2)=... %come sopra i=3
dx(3,3)=... %come sopra i=3
dx(3,4)=... %come sopra i=3
dy(3,1)=... %come sopra i=3
dy(3,2)=... %come sopra i=3
dy(3,3)=... %come sopra i=3
dy(3,4)=... %come sopra i=3
dz(3,1)=... %come sopra i=3
dz(3,2)=... %come sopra i=3
dz(3,3)=... %come sopra i=3
dz(3,4)=... %come sopra i=3
t=(p(n(2),1:3)*cost+d)/((p(n(2),1:3)*cost)-(p(n(4),1:3)*cost));
x(4)=p(n(2),1)-t*(p(n(2),1)-p(n(4),1));
y(4)=p(n(2),2)-t*(p(n(2),2)-p(n(4),2));
z(4)=p(n(2),3)-t*(p(n(2),3)-p(n(4),3));
dx(4,1)=... %come sopra i=4
dx(4,2)=... %come sopra i=4
dx(4,3)=... %come sopra i=4
dx(4,4)=... %come sopra i=4
dy(4,1)=... %come sopra i=4
dy(4,2)=... %come sopra i=4
dy(4,3)=... %come sopra i=4
dy(4,4)=... %come sopra i=4
dz(4,1)=... %come sopra i=4
dz(4,2)=... %come sopra i=4
dz(4,3)=... %come sopra i=4
dz(4,4)=... %come sopra i=4
diffx=... %come sopra
diffy=... %come sopra
diffz=... %come sopra
dv(m,1:4)=sigma*sign(vol)*(diffx*dx+diffy*dy+diffz*dz)/6;
%secondo tetraedro
t=(p(n(1),1:3)*cost+d)/((p(n(1),1:3)*cost)-(p(n(3),1:3)*cost));
x(2)=p(n(1),1)-t*(p(n(1),1)-p(n(3),1));
y(2)=p(n(1),2)-t*(p(n(1),2)-p(n(3),2));
z(2)=p(n(1),3)-t*(p(n(1),3)-p(n(3),3));
dx(2,1)=... %come sopra i=2
dx(2,2)=... %come sopra i=2
dx(2,3)=... %come sopra i=2
dx(2,4)=... %come sopra i=2
dy(2,1)=... %come sopra i=2
dy(2,2)=... %come sopra i=2
dy(2,3)=... %come sopra i=2
dy(2,4)=... %come sopra i=2
dz(2,1)=... %come sopra i=2
dz(2,2)=... %come sopra i=2

```

```

dz(2,3)=... %come sopra i=2
dz(2,4)=... %come sopra i=2
diffx=... %come sopra
diffy=... %come sopra
diffz=... %come sopra
dv(m,1:4)=dv(m,1:4)+sigma*sign(vol)*(diffx*dx+diffy*dy+diffz*dz)/6;
    %terzo tetraedro
t=(p(n(1),1:3)*cost+d)/((p(n(1),1:3)*cost)-(p(n(4),1:3)*cost));
x(3)=p(n(1),1)-t*(p(n(1),1)-p(n(4),1));
y(3)=p(n(1),2)-t*(p(n(1),2)-p(n(4),2));
z(3)=p(n(1),3)-t*(p(n(1),3)-p(n(4),3));
dx(3,1)=... %come sopra i=3
dx(3,2)=... %come sopra i=3
dx(3,3)=... %come sopra i=3
dx(3,4)=... %come sopra i=3
dy(3,1)=... %come sopra i=3
dy(3,2)=... %come sopra i=3
dy(3,3)=... %come sopra i=3
dy(3,4)=... %come sopra i=3
dz(3,1)=... %come sopra i=3
dz(3,2)=... %come sopra i=3
dz(3,3)=... %come sopra i=3
dz(3,4)=... %come sopra i=3
diffx=... %come sopra
diffy=... %come sopra
diffz=... %come sopra
dv(m,1:4)=dv(m,1:4)+sigma*sign(vol)*(diffx*dx+diffy*dy+diffz*dz)/6;
end
end

```

A.5 dFEMmatrix.m

```

function [AA]=dFEMmatrix(msh,a,b,c,d,s1,s2)
% Calcola la derivata della matrice A
der=deriv(msh,a,b,c,d,s1,s2);
AA=zeros(size(msh.vtx_f,1),size(msh.vtx_f,1),4);
int=inters(msh,a,b,c,d);
elem=msh.elem_f;
for k=1:size(elem,1)
    for i=1:4
        for j=1:4
            d_el=der(k,:);
            gi=[msh.gx(k,elem(k,i)),msh.gy(k,elem(k,i)),msh.gz(k,elem(k,i))];
            gj=[msh.gx(k,elem(k,j)),msh.gy(k,elem(k,j)),msh.gz(k,elem(k,j))];
            AA(elem(k,i),elem(k,j),1)=AA(elem(k,i),elem(k,j),1)+
                -(d_el(1)*(gi*gj'));
            AA(elem(k,i),elem(k,j),2)=AA(elem(k,i),elem(k,j),2)+
                -(d_el(2)*(gi*gj'));

```

```

AA(elem(k,i),elem(k,j),3)=AA(elem(k,i),elem(k,j),3)+
                               -(d_el(3)*(gi*gj'));
AA(elem(k,i),elem(k,j),4)=AA(elem(k,i),elem(k,j),4)+
                               +(d_el(4)*(gi*gj'));
    end
end
end

```

A.6 jacobian.m

```

function J=jacobian(msh,a,b,c,d,s1,s2,meas_spec)
% Calcola la derivata delle misure rispetto ai parametri
% INPUT: msh=struttura mesh; a,b,c,d=parametri del piano
% s1,s2=conduttività
% meas_spec=tipo di misurazione, in ogni riga [ V+ V- I+ I- I ]
% OUTPUT: J=(m x n) jacobiano
A=FEMmatrix(msh,a,b,c,d,s1,s2);
num_meas=size(meas_spec,1);
num_nodes=size(A,1);
A(msh.gnd,:)=0;
A(:,msh.gnd)=0;
A(msh.gnd,msh.gnd)=1;
d_A=dFEMmatrix(msh,a,b,c,d,s1,s2);
for i=1:num_meas
    V1=msh.pE{meas_spec(i,1),1}(1,2);
    V2=msh.pE{meas_spec(i,2),1}(1,2);
    bb=zeros(num_nodes,1);
    bb(msh.pE{meas_spec(i,3),1}(1,2))=meas_spec(i,5);
    bb(msh.pE{meas_spec(i,4),1}(1,2))=-meas_spec(i,5);
    bb(msh.gnd)=0;
    [uu,flag]=pcg(A,bb,1e-10,500);
    [du_da,flag]=pcg(A,-d_A(:, :, 1)*uu,1e-6,500);
    [du_db,flag]=pcg(A,-d_A(:, :, 2)*uu,1e-6,500);
    [du_dc,flag]=pcg(A,-d_A(:, :, 3)*uu,1e-6,500);
    [du_dd,flag]=pcg(A,-d_A(:, :, 4)*uu,1e-6,500);
    J(i,1)=du_da(V1)-du_da(V2);
    J(i,2)=du_db(V1)-du_db(V2);
    J(i,3)=du_dc(V1)-du_dc(V2);
    J(i,4)=du_dd(V1)-du_dd(V2);
end

```

A.7 inverse_solve.m

```

function P=inverse_solve(msh,s1,s2,meas_spec,meas,MAXIT,TOL,P0)
% Risolve il problema inverso
% INPUTS: msh=struttura mesh; a,b,c,d=parametri del piano
% s1,s2=conduttività; meas=vettore delle misure;

```

```

% meas_spec=tipo di misurazione, in ogni riga [ V+ V- I+ I- I ]
% MAXIT=num massimo iterazioni; TOL=tolleranza; P0=stima iniziale
%OUTPUTS: P=[a,b,c,d] parametri del piano
scaip_disp_3D_surf_msh(msh);
hold on;
axis off;
axis([-30 30 -30 30 -15 15]);
P(1,1:4)=P0./norm(P0);
h(1)=disegna(msh,P(1,1),P(1,2),P(1,3),P(1,4));
set(h(1),'FaceColor',[1 0 0]);
set(h(1),'FaceAlpha',0.8);
for i=1:MAXIT
    m=measures(msh,P(i,1),P(i,2),P(i,3),P(i,4),s1,s2,meas_spec);
    P(i,5)=norm(m-meas);
    if P(i,5)<TOL
        return
    end
    jac=jacobian(msh,P(i,1),P(i,2),P(i,3),P(i,4),s1,s2,meas_spec);
    [U,S,V]=svd(jac);
    k=1;
    dP=[0;0;0;0];
    while k<=size(S,2) & S(k,k)>1e-3
        dP=dP-(U(:,k)')*(m-meas)*(1/S(k,k))*V(:,k));
        k=k+1;
    end
    normdP=norm(dP);
    if normdP>1
        dP=dP./(2*normdP)
    end
    if i>1
        if abs((P(i,5)-P(i-1,5))/P(i,5))<0.1
            dP=dP./(2*normdP)+[rand*2-1;rand*2-1;rand*2-1;rand*2-1];
        end
    end
    Pnew=P(i,1:4)+dP';
    P(i+1,1:4)=Pnew/norm(Pnew);
    h(i+1)=disegna(msh,P(i+1,1),P(i+1,2),P(i+1,3),P(i+1,4));
    set(h(i+1),'FaceColor',[(1-i/MAXIT) 0 i/MAXIT]);
    set(h(i+1),'FaceAlpha',0.6);
end
end

```

A.8 disegna.m

```

% Disegna il piano ax+by+cz+d=0
% e restituisce some output l'handle h
function [h]=disegna(msh,a,b,c,d)
x1=1.1*max(msh.vtx_f(:,1));
y1=1.1*max(msh.vtx_f(:,2));

```

```
z1=-(a*x1+b*y1+d)/c;
x2=1.1*max(msh.vtx_f(:,1));
y2=1.1*min(msh.vtx_f(:,2));
z2=-(a*x2+b*y2+d)/c;
x3=1.1*min(msh.vtx_f(:,1));
y3=1.1*min(msh.vtx_f(:,2));
z3=-(a*x3+b*y3+d)/c;
x4=1.1*min(msh.vtx_f(:,1));
y4=1.1*max(msh.vtx_f(:,2));
z4=-(a*x4+b*y4+d)/c;
P=[x1,y1,z1
    x2,y2,z2
    x3,y3,z3
    x4,y4,z4];
f=[1,2,3,4];
h=patch('Vertices',P,'Faces',f);
set(h,'FaceColor',[rand rand rand]);
set(h,'FaceAlpha',0.8);
```


Bibliografia

- [1] BORSIC A., *Regularisation methods for imaging from electrical measurements*, PhD thesis, Oxford Brookes University, 2002.
- [2] CHENEY M., ISAACSON D. e NEWELL J. C., "Electrical impedance tomography" in *SIAM Review*, vol. 41, n. 1, p. 85-101, 1999.
- [3] DRYJA M., SARKIS M. V. e WIDLUND O. B., "Multilevel Schwarz methods for elliptic problems with discontinuous coefficients in three dimension" in *Numerische Mathematik*, n. 72, p.313-348, 1996.
- [4] GRISVARD P., *Elliptic Problems in Nonsmooth Domains*, Pitman, Boston, 1985.
- [5] KOLEHMAINEN V., *Novel approaches to image reconstruction in diffusion tomography*, PhD thesis, Kuopio University, 2001.
- [6] LIONHEART W., POLYDORIDES N. e BORSIC A., *Electrical impedance tomography: methods, history, and applications*, Capitolo 1, Bristol, Institute of Physics Publishing, , 2004.
- [7] SOMERSALO E., CHENEY M. e ISAACSON D., "Existence and uniqueness for electrode models for electric current computed tomography" in *SIAM Journal Appl. Math.*, vol. 52, n. 4, p.1023-1040, 1992.
- [8] PETZOLDT M., "A posteriori error estimators for elliptic equations with discontinuous coefficients" in *Advances in Computational Mathematics*, n. 16, p.47-75, 2002. Nicholas - Image reconstruction algorithms for soft-field tomography
- [9] POLYDORIDES N., *Image reconstruction algorithms for soft-field tomography*, PhD thesis, University of Manchester, 2002.
- [10] TARANTOLA A., *Inverse Problem Theory and Methods for Model Parameter Estimation*, Siam, 2005.

- [11] TOSSAVAINEN O. P., KOLEHMAINEN V. e VAUHKONEN M., "Free-surface and admittivity estimation in electrical impedance tomography" in *International Journal of Numerical Methods in Engineering*, in stampa, 2005.
- [12] TOSSAVAINEN O. P., VAUHKONEN M., HEIKKINEN L. M. e SAVOLAINEN T., "Estimating shapes and free surfaces with electrical impedance tomography" in *Meas Sci Technol*, n. 15, p.1402-1411, 2004.
- [13] VAUHKONEN M., *Electrical impedance tomography and prior information*, PhD thesis, Kuopio University, 1997.
- [14] WEBSTER J.G., *Electrical impedance tomography*, Adam Hilger Series, 1990.