

UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA INDUSTRIALE

CORSO DI LAUREA MAGISTRALE IN  
INGEGNERIA DELL'ENERGIA ELETTRICA

**Applicazione del metodo di riduzione  
d'ordine a formulazioni integrali per  
simulazioni elettromagnetiche**

*Relatore:*

PROF. RICCARDO TORCHIO

*Laureando:*

MATTIA MILANI

1205988

Anno Accademico 2021/2022



# Indice

<b>Introduzione</b>	<b>1</b>
<b>1 Tableau Analysis per la risoluzione di circuiti elettrici</b>	<b>3</b>
1.1 Teoria dei grafi . . . . .	4
1.2 Formulazioni matriciali dei circuiti elettrici . . . . .	7
1.2.1 KCL . . . . .	8
1.2.2 KVL . . . . .	9
1.2.3 Equazioni dei componenti . . . . .	10
1.3 Tableau analysis . . . . .	14
1.3.1 Caso statico . . . . .	14
1.3.2 Caso dinamico . . . . .	16
<b>2 Model Order Reduction</b>	<b>19</b>
2.1 Tableau Analysis in Matlab . . . . .	19
2.1.1 Algoritmo principale . . . . .	19
2.1.2 Algoritmo modificato . . . . .	24
2.2 Model Order Reduction: contenuti e finalità del metodo . . . . .	27
2.2.1 Implementazione MOR alla Tableau Analysis . . . . .	28
<b>3 Applicazione del MOR a un circuito generico</b>	<b>33</b>
3.1 Analisi circuito con risonanza parallelo . . . . .	33
3.1.1 Circuito preso in esame . . . . .	33
3.1.2 Confronto risultati con e senza MOR . . . . .	35
3.2 Analisi linea di trasmissione elettrica a costanti concentrate . . . . .	42
3.2.1 Modello a costanti concentrate . . . . .	42
3.2.2 Script per costruzione netlist della linea . . . . .	44
3.2.3 Applicazione Tableau Analysis con MOR . . . . .	47

---

<b>4</b>	<b>Introduzione al metodo PEEC</b>	<b>53</b>
4.1	Metodi di risoluzione per problemi elettromagnetici . . . . .	53
4.1.1	FDM: metodo alle differenze finite . . . . .	54
4.1.2	FEM: metodi agli elementi finiti . . . . .	55
4.1.3	IEM: metodi alle equazioni integrali . . . . .	57
4.2	PEEC . . . . .	59
<b>5</b>	<b>Applicazione del MOR al metodo PEEC</b>	<b>63</b>
5.1	Utilità del modello di riduzione d'ordine nel metodo PEEC . . . . .	63
5.2	Modifica script del MOR per programma PEEC 1D . . . . .	64
5.3	Simulazione PEEC 1D con metodo MOR: risultati ed osservazioni	68
5.3.1	Cavo bifilare . . . . .	69
5.3.2	Antenna NFC . . . . .	73
	<b>Conclusioni</b>	<b>79</b>
	<b>Bibliografia</b>	<b>81</b>

# Introduzione

Con il progresso dell'elettronica e con la sempre più elevata complessità dei circuiti elettrici nei vari settori d'applicazione, i metodi computazionali per la progettazione e il collaudo sono divenuti fondamentali per una realizzazione delle mansioni efficiente e che fosse eseguita tempi rapidi.

Le basi dell'elettrotecnica moderna hanno le loro radici nella fine del XVIII secolo e più precisamente nel 1800, anno in cui Alessandro Volta dà vita alla pila elettrica: la pila di Volta. In quel momento storico l'elettricità non era ancora compresa appieno nella sua natura, ma era considerata come un fluido: successivamente, grazie alle scoperte di vari personaggi di spicco come Hertz, Laplace, Faraday, Ohm e molti altri, si riuscì ad apprendere di più sul mondo dell'elettricità. A pari passo con le scoperte nel campo dell'elettricità si iniziò a capire che questa era profondamente legata ad un'altra realtà, quella magnetica. Dalle teorie che emersero, particolare rilevanza hanno le equazioni di Maxwell che nel 1865 pubblicò "A Dynamical Theory of the Electromagnetic Field", opera che riassume e integra le varie formulazioni sull'elettromagnetismo.

Dalla fine del XIX secolo queste scoperte trovarono vita in molte applicazioni commerciali come macchine elettriche (motori, generatori, trasformatori) e la trasmissione elettrica a fili.

Con l'avvento della rivoluzionaria elettronica di potenza e l'utilizzo dei calcolatori, tutte le formulazioni fatte fino a quel momento trovarono terreno fertile per svilupparsi in un ambiente molto più dinamico e con grande prospettive di miglioramento, motivo per cui parliamo di elettrotecnica computazionale. Questa nuova disciplina ha permesso di realizzare software e codici per la realizzazione di studi dettagliati sui campi magnetici ed elettrici, linfa vitale per la ricerca in tale ambito.

Questi codici tuttavia, con la complessità sempre più elevata dei circuiti elettrici ma soprattutto elettronici [1], hanno bisogno di essere sviluppati tenendo conto delle capacità di computazione dei calcolatori.

La passione per la materia, unita alla necessità da parte dell'elettrotecnica di avere dei metodi di risoluzioni efficienti, rapidi e con basso utilizzo di memoria di sistema per l'esecuzione dei programmi, hanno portato alla decisione di realizzare il seguente lavoro di tesi.

# Capitolo 1

## Tableau Analysis per la risoluzione di circuiti elettrici

Considerando i circuiti elettrici presenti in ambito industriale e commerciale, essi possono essere suddivisi in circuiti di piccole dimensioni, ad esempio nel campo dell'elettronica e della microelettronica, e in circuiti di medie/grandi dimensioni, come può essere ad esempio la rete di distribuzione di energia elettrica italiana. Questi due tipi di circuiti condividono la peculiarità di essere composti da una moltitudine di componenti, fino anche a milioni di parti; ciò rende la complessità del circuito ovviamente troppo elevata e rende necessaria un'efficiente risoluzione computazionale.

Affinchè un metodo di risoluzione di tipo computazionale sia corretto e pro-pedeutico alle varie situazioni che si possono manifestare, esso deve rispettare tre principi fondamentali:

**generalità** Il metodo deve essere il più generale possibile per poter essere utilizzato in più casistiche diverse tra di loro.

**efficienza** Il metodo deve poter risolvere il problema in un tempo ridotto, e poter essere memorizzato in una memoria di sistema il più piccola possibile; questo punto è il predominante, in riferimento al lavoro che è stato svolto in questa tesi.

**facilità di implementazione** questo fattore è un vantaggio per le realtà commerciali, dove l'implementazione di un software deve poter essere il più flessibile possibile.

Si introduce ora la teoria dei grafi, base fondamentale per l'analisi di circuiti in ambiente computazionale.

## 1.1 Teoria dei grafi

Si parla di grafo [2] quando si fa generalmente riferimento ad un insieme di  $n$  nodi ed  $e$  lati, in cui ad ogni lato si può associare un coppia di nodi agli estremi. Se i lati sono orientati si parla di grafo orientato.

Per descrivere al meglio un grafo si delineano alcune proprietà da esso possedute:

- Connesso Significa che esiste almeno un percorso formato da uno o più lati che connette qualsiasi coppia di nodi presa come riferimento [3].

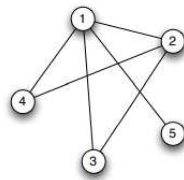


Figura 1.1: Grafo connesso

- Non connesso Un grafo privo della proprietà precedente è detto non connesso.

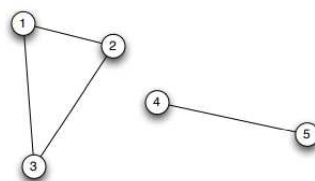


Figura 1.2: Grafo non connesso



Tuttavia, si può ricondurre un grafo non connesso ad uno connesso con l'aggiunta di un lato che connetta due nodi [3] (*hinging*).

Si prende ad esempio il caso di un trasformatore, che può essere rappresentato da un grafo sconnesso, dato che vi è isolamento galvanico tra primario e secondario, ma che può essere ricondotto ad uno connesso con questa tecnica.

Elettricamente parlando il circuito non cambia poichè non si ha un ritorno della corrente, motivo per cui non si hanno modifiche nei voltaggi dei nodi.

- Planarietà Un grafo è detto planare quando è possibile disegnarlo senza intersezioni dei lati.

Questo fattore è fondamentale per applicazioni quali gestione logistica di magazzino o progettazione di circuiti stampati.

Tutti i grafi con

$$n < 4 \tag{1.1}$$

sono planari.

Si distinguono ora i principali elementi topologici che compongono un grafo: questi elementi sono considerati per eseguire e/o facilitare operazioni e valutarne le caratteristiche proprie.

- Maglia Una maglia di un grafo è un sottografo che soddisfa due condizioni:
  1. sia connesso
  2. ogni nodo sia legato a due e due soli lati
- Anello Viene chiamato anello una maglia che copre una superficie non attraversata da alcun lato del grafo.
- Albero Si parla di albero di un grafo, quando si fa riferimento ad un sottografo che soddisfa alcune caratteristiche:
  1. include tutti i nodi
  2. non forma maglie
  3. è connesso
- Coalbero Viene indicato come coalbero l'insieme di lati complementari all'albero di riferimento.

- Insieme di taglio Un insieme di taglio è un gruppo di elementi del grafo che soddisfa ciò che segue:
  1. la rimozione di tutti i lati escludendone uno non modifica la caratteristica del grafo di essere connesso
  2. la rimozione di tutti i lati rende il grafo non connesso

Prima di proseguire, è bene definire un concetto importante per comprendere i successivi ragionamenti.

Si definisce NETLIST un file contenente tutte le informazioni topologiche e tipologiche di un circuito; semplificando, consiste in una lista di componenti elettrici/elettronici coinvolti nel circuito e i relativi nodi a cui sono collegati.

Un esempio è mostrato di seguito:

starting node	ending node	type	value
1	2	V	20
2	3	R	5
3	4	R	10
4	1	R	8

**Tabella 1.1:** Esempio di netlist.

Il circuito descritto da questa NETLIST è formato da soli bipoli resistivi e un'alimentazione. Si nota che i nodi sono orientati seguendo l'orientamento del grafo: nella prima colonna è indicato il nodo di ingresso della corrente, nel componente, il quale viene stabilito a priori, mentre nella seconda colonna il nodo di uscita. Tutto ciò viene fatto secondo convenzione dell'utilizzatore per i componenti, per cui si avrà il potenziale maggiore (+) al nodo d'ingresso e il potenziale minore al nodo di uscita (-). Nella terza colonna viene indicato il tipo di bipolo presente in quel lato compreso tra i due specifici nodi presi in considerazione: in questo caso caso è presente un'alimentazione e 3 elementi resistivi. Nella quarta colonna è indicato il valore di riferimento del bipolo.

Nel caso di elementi a più poli, una rappresentazione diretta non è possibile seguendo tale schema ma una generalizzazione è praticabile.

## 1.2 Formulazioni matriciali dei circuiti elettrici

La rappresentazione di un circuito può quindi essere fatta attraverso la teoria dei grafi. Da qui si può svolgere un'analisi matriciale [3] secondo precisi passaggi atti a risolvere il circuito, il che significa trovare, per ogni bipolo, corrente e tensione ai capi.

Non è possibile costruire un software generale che si basi sull'analisi di ogni singolo anello, perchè dovrei verificare la planarietà, cosa generalmente molto difficile da fare. Un possibile approccio alternativo sarebbe quello di analizzare alberi e coalberi; difatti i rami di coalbero hanno una proprietà: ognuno di essi chiude una maglia con i relativi rami di albero, presi singolarmente. Tuttavia ciò non è sufficiente, poichè trovare le giuste maglie per risolvere il circuito non è elementare.

Si prende ora a riferimento un generico circuito:

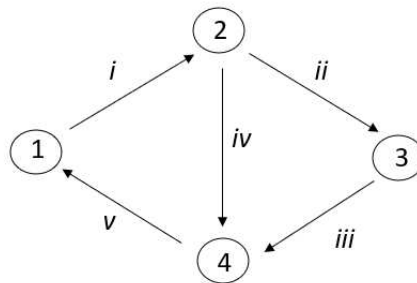


Figura 1.3: Grafo semplice

Si definisce matrice d'incidenza  $\mathbf{A}$  di tale grafo la matrice:

$$A_c = \begin{bmatrix} +1 & 0 & 0 & 0 & -1 \\ -1 & +1 & 0 & +1 & 0 \\ 0 & -1 & +1 & 0 & 0 \\ 0 & 0 & -1 & -1 & +1 \end{bmatrix} \quad (1.2)$$

Il numero di righe corrisponde al numero di nodi e il numero di colonne cor-

risponde al numero di lati. Si può costruire la matrice d'incidenza riferendomi sia ai lati (orientati) sia ai nodi; prendendo come riferimento i lati si indica con +1 il corrispondente nodo di partenza mentre con un -1 il nodo di arrivo. Tutti gli altri elementi della colonna hanno valore 0.

Se si sommano tutte le righe della matrice d'incidenza, si ha come risultante un vettore nullo  $((0), (0), \dots(0))$ ; ciò significa che i vettori della matrice sono tra di loro linearmente dipendenti. Da questa matrice quindi ne risulterebbe un sistema di equazioni non linearmente indipendenti.

Questa matrice è detta matrice d'incidenza completa  $A_c$ .

Se dalla matrice d'incidenza completa si elimina una riga qualsiasi, si ottengono ora dei vettori linearmente indipendenti tra loro: questa nuova matrice è detta matrice d'incidenza ridotta  $A$ . Tale matrice può essere la seguente, se si elimina l'ultima riga:

$$A_c = \begin{bmatrix} +1 & 0 & 0 & 0 & -1 \\ -1 & +1 & 0 & +1 & 0 \\ 0 & -1 & +1 & 0 & 0 \end{bmatrix} \quad (1.3)$$

Per risolvere il circuito si utilizzano le *Leggi di Kirchhoff* tramite l'utilizzo della matrice d'incidenza.

### 1.2.1 KCL

La prima legge di Kirchhoff è la legge sulle correnti nei nodi ed afferma che la sommatoria delle correnti prese con il corretto segno, entranti ed uscenti da una generica superficie che contiene parte di un circuito elettrico, ha risultante zero. È possibile quindi utilizzare la matrice d'incidenza ridotta per scrivere in modo semplice le equazioni ai nodi delle correnti seconda la prima legge di Kirchhoff; il vettore delle correnti  $\mathbf{i}$  avrà dimensione  $(nb, 1)$ , dove  $nb$  è il numero dei lati del circuito (o del grafo). Una volta risolto il circuito queste variabili saranno trovate.

Le equazioni della KCL [3] (*Kirchhoff's current law*) si ottengono quindi moltiplicando la matrice d'incidenza ridotta  $A$  per il vettore delle correnti:

$$\begin{bmatrix} +1 & 0 & 0 & 0 & -1 \\ -1 & +1 & 0 & +1 & 0 \\ 0 & -1 & +1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (1.4)$$

che può essere espresso in forma matriciale come segue:

$$\mathbf{A}_{(nn-1)*nb} * \mathbf{i}_{nb*1} = \mathbf{0}_{(nn-1)*1} \quad (1.5)$$

In questo modo si hanno ad ora  $(nb)$  variabili e  $(nn - 1)$  equazioni per la risoluzione del circuito.

### 1.2.2 KVL

La seconda legge di Kirchhoff riguarda le maglie di un circuito: essa dice che la sommatoria delle tensioni in un percorso chiuso, come lo è una maglia o un anello di un circuito, è pari a 0.

Come si è detto precedentemente ciò sarebbe complicato da esplicitare, motivo per cui si segue un'altra strada. Alla base della legge di Kirchhoff giace la seguente considerazione: il campo elettrico, in elettrostatica, soddisfa:

$$\nabla \times \mathbf{E} = 0 \quad (1.6)$$

ossia è irrotazionale, conservativo, il che permette di scrivere le tensioni come differenze di potenziali nodali nel caso circuitale. Ciò comporta l'aggiunta di ulteriori variabili al problema, ma permette una risoluzione semplice ed efficace. Si può quindi scrivere [3]:

$$\mathbf{A}_{nb*(nn-1)}^T * \mathbf{P}_{(nn-1)*1} - \mathbf{v}_{nb*1} = \mathbf{0}_{nb*1} \quad (1.7)$$

dove  $\mathbf{p}$  è il vettore dei potenziali nodali ed avrà dimensione  $(nn-1)$  poichè, come precedentemente detto, un nodo viene preso come riferimento e portato al valore nullo,  $\mathbf{v}$  è il vettore delle tensioni ed avrà dimensione  $(nb)$  e  $A^T$  è la matrice d'incidenza trasposta.

In riferimento alla figura 1.3 si avrà:

$$\begin{cases} p_1 - p_2 - v_1 = 0 \\ p_2 - p_3 - v_2 = 0 \\ p_3 - v_3 = 0 \\ p_2 - v_4 = 0 \\ -p_1 - v_5 = 0 \end{cases} \quad (1.8)$$

e se consideriamo la maglia più esterna:

$$\left\{ v_1 + v_2 + v_3 + v_5 = 0 \right. \implies p_1 - p_2 + p_2 - p_3 + p_3 - p_1 = 0 \quad (1.9)$$

il che verifica la legge di Kirchhoff delle tensioni.

In questo caso il nodo 4 è stato portato al valore nullo, per cui nella valutazione dei potenziali non viene conteggiato.

È interessante notare la tipologia di equazioni che si ottengono con questo metodo; per circuiti complessi risulterebbero molte equazioni ma contenenti pochi termini. Ne deriveranno quindi matrici molto sparse, il che va a favore di una riduzione di spazio utilizzato per la loro memorizzazione.

Con le nuove equazioni ora, il computo del circuito sarà di  $2(nb) + (nn - 1)$  variabili ossia correnti, tensioni e potenziali, a fronte di sole  $(nb) + (nn - 1)$  equazioni.

### 1.2.3 Equazioni dei componenti

Le restanti  $(nb)$  equazioni mancanti per la risoluzione dell'analisi circuitale, vengono estratte dalle informazioni che si hanno sui componenti in ogni lato.

Si considerino per ora solo elementi circuitali che siano bipoli, mentre per altre tipologie (doppi bipoli,...) si farà un accenno a fine sottosezione.

Si parte dal considerare che per ogni elementi del  $k$ -esimo lato di un circuito preso in analisi, si possa identificare un'equazione generale univoca che ne identifica la tipologia di bipolo:

$$Ri + Gv + L\frac{di}{dt} + C\frac{dv}{dt} = b \quad (1.10)$$

che in forma matriciale possiamo riscrivere come:

$$[R]\{i\} + [G]\{v\} + [L]\left\{\frac{di}{dt}\right\} + [C]\left\{\frac{dv}{dt}\right\} = \{b\} \quad (1.11)$$

dove  $[R]$  è la matrice dei coefficienti associati alla  $\mathbf{i}$ ,  $[G]$  la matrice dei coefficienti associati alla  $\mathbf{v}$ ,  $[L]$  la matrice delle induttanze,  $[C]$  la matrice delle conduttanze e  $\mathbf{b}$  il vettore dei termini noti. Le 4 matrici sono quadrate avranno dimensione  $nb * nb$  e i vettori avranno dimensione  $nb * 1$ : si otterranno per cui (nb) equazioni ma senza aggiunta di altre variabili.

Si analizza ora il procedimento per implementare l'equazione dei componenti nelle varie tipologie di bipolo, il che darà indicazione del valore dei coefficienti e delle posizioni di questi ultimi nelle varie matrici sovraccitate. I principali casi che di distinguono e che utilizzeremo per lo svolgimento di questa tesi, possono essere i seguenti:

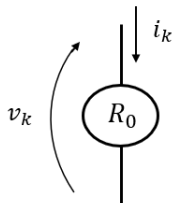
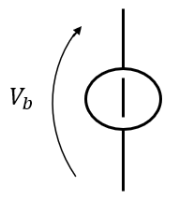
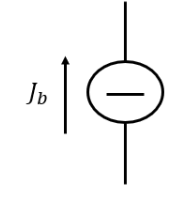
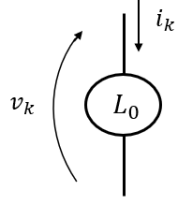
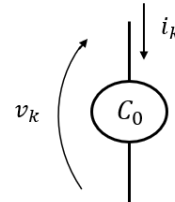
Tipo di bipolo	$R_{kk}$	$G_{kk}$	$L_{kk}$	$C_{kk}$	$b_k$
	$-R_0$	1	0	0	0
	0	1	0	0	$V_b$
	1	0	0	0	$J_b$
	0	1	$-L_0$	0	0
	1	0	0	$-C_0$	0

Tabella 1.2: Parametri per le equazione dei componenti principali.



Con questa tipologia di componenti si otterranno matrici diagonali, dato che ogni elemento è inserito nella posizione  $kk$  che fa riferimento alla sola diagonale. Nel caso si fosse considerata la presenza di doppi bipoli, come ad esempio un trasformatore, si sarebbero ottenute matrici non più diagonali ma con elementi al di fuori della diagonale.

Tuttavia, in qualunque di questi casi, ciò che è fondamentale notare è che le matrici ottenute sono molte sparse, il che comporta un elevato risparmio in termini di memoria.

Per evidenziare un esempio pratico, sempre in riferimento alla figura 1.3, si immagini che al lato  $iii$  si trovi un resistore di valore  $R_{33} = 20\Omega$  e si ipotizzi che sia l'unico resistore presente nel circuito; all'equazione generale dei componenti si avrà:

$$-R_{33} \cdot i_3 + 1 \cdot v_3 = 0 \quad (1.12)$$

ossia

$$v_3 = R_{33} \cdot i_3 \quad (1.13)$$

per cui la matrice di dimensione  $5 \times 5$  avrà un solo termine alla posizione 33 e sarà:

$$R = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -20 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (1.14)$$

## 1.3 Tableau analysis

Dalle equazioni delle correnti, delle tensioni e dei componenti si hanno ora  $(2nb) + (nn - 1)$  equazioni e  $(2nb) + (nn - 1)$  variabili; ciò significa che il problema avrebbe soluzione se si dimostra che le equazioni sono linearmente indipendenti. Dalle valutazioni fatte sulla matrice d'incidenza ridotta si deduce che i sistemi relativi alla KVL e alla KCL sono formati da equazioni tra loro linearmente indipendenti, mentre sulle equazioni dei componenti si può fare una considerazione: ogni elemento presente sulla diagonale, ad esempio della matrice R, non dipende dalle altre righe del sistema poichè esse faranno testo ad altri elementi della diagonale o tutt'al più a valori nulla, motivo per cui ogni riga non potrebbe essere scritta come combinazione lineare delle altre, il che le rende tra di loro linearmente indipendenti.

### 1.3.1 Caso statico

Si formalizza quindi l'*Analisi di Tableau* [4] [5]. Unendo le forme matriciali [3] precedentemente viste si ottiene una matrice di ordine maggiore che da una visuale più chiara di quello che è il nuovo sistema e degli elementi da cui è formato. Si analizza in prima battuta il caso statico, ponendo quindi:

$$\frac{di}{dt} = 0; \quad \frac{dv}{dt} = 0; \quad \frac{dp}{dt} = 0; \quad (1.15)$$

In questo caso il sistema che si ottiene sarà il seguente:

$$\begin{bmatrix} A & \emptyset & \emptyset \\ \emptyset & -\mathbb{1} & A^T \\ R & G & 0 \end{bmatrix} \cdot \begin{bmatrix} i_{nb \cdot 1} \\ v_{nb \cdot 1} \\ p_{(nn-1) \cdot 1} \end{bmatrix} = \begin{bmatrix} 0_{nb \cdot 1} \\ 0_{nb \cdot 1} \\ b_{(nn-1) \cdot 1} \end{bmatrix} \quad (1.16)$$

dove

- $A$  è la matrice d'incidenza ridotta di dimensioni  $nn - 1 \times nb$
- $-\mathbb{1}$  è la matrice quadrata identità presa con il segno negativo di dimensioni  $nb \times nb$
- $R$  è la matrice dei coefficienti legati alla  $i$  di dimensioni  $nb \times nb$
- $G$  è la matrice dei coefficiente legati alla  $v$  di dimensioni  $nb \times nb$
- $\emptyset$  è la matrice nulla, di dimensioni  $nn - 1 \times nn - 1$  per quanto riguarda quella presente nella prima riga, e di dimensioni  $nb \times nb$  per quelle presenti nella altre due righe della matrice

Se si pone:

$$M = \begin{bmatrix} A & \emptyset & \emptyset \\ \emptyset & -\mathbb{1} & A^T \\ R & G & 0 \end{bmatrix}; \quad x = \begin{bmatrix} i_{nb \cdot 1} \\ v_{nb \cdot 1} \\ p_{(nn-1) \cdot 1} \end{bmatrix}; \quad r = \begin{bmatrix} 0_{nb \cdot 1} \\ 0_{nb \cdot 1} \\ b_{(nn-1) \cdot 1} \end{bmatrix} \quad (1.17)$$

si può riscrivere l'equazione matriciale vista poco fa come:

$$M \cdot x = r \quad (1.18)$$

L'ambiente in cui viene svolto questo lavoro di tesi è Matlab, strumento molto potente che negli ultimi anni sta emergendo sempre più, anche all'interno di realtà aziendali, soprattutto grazie ad applicazioni correlate come Simulink.

Per la risoluzione di equazioni come questa in Matlab si trovano due vie, raffigurate di seguito:

```
1 x=inv(M)*b; % trovo matrice inversa di M e calcolo
```

```

2 x=M\r;           % dato M e dato r, trovo x secondo metodo proprio
  di Matlab

```

**Listing 1.1:** Metodi di risoluzione per sistemi di equazione in Matlab

Nella prima, viene trovata la matrice inversa di  $M$  ed eseguito il calcolo; nella seconda, posti come dati del problema  $M$  ed  $r$ , si trova  $x$  secondo un metodo proprio di Matlab basato sull'eliminazione di Gauss. Entrambi i metodi sono validi, ma a differenza del secondo, il primo comporta il calcolo della matrice inversa  $M^{-1}$  la quale richiede un'ulteriore spazio di archiviazione in memoria; si deve cercare di ridurre quest'ultima quando possibile, motivo per cui verrà usato il secondo algoritmo per i calcoli successivi.

### 1.3.2 Caso dinamico

Dopo aver visto il caso statico, si espone ora quello dinamico: più precisamente si coinvolgono i casi in cui anche le induttanze e le capacità giocano il loro ruolo nel circuito. Dall'equazione vista precedentemente, si integra ora come segue:

$$\begin{bmatrix} A & \emptyset & \emptyset \\ \emptyset & -\mathbb{1} & A^T \\ R & G & 0 \end{bmatrix} \cdot \begin{bmatrix} i_{nb \cdot 1} \\ v_{nb \cdot 1} \\ p_{(nn-1) \cdot 1} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ L & C & 0 \end{bmatrix} \cdot \frac{d}{dt} \cdot \begin{bmatrix} i_{nb \cdot 1} \\ v_{nb \cdot 1} \\ p_{(nn-1) \cdot 1} \end{bmatrix} = \begin{bmatrix} 0_{nb \cdot 1} \\ 0_{nb \cdot 1} \\ b_{(nn-1) \cdot 1} \end{bmatrix} \quad (1.19)$$

Si nota che la nuova matrice definita come:

$$M_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ L & C & 0 \end{bmatrix} \quad (1.20)$$

è una matrice molto sparsa; le prime righe di matrici sono state poste nulle poichè le KVL e le KCL non vengono modificate passando dal caso statico a quello dinamico. La terza riga invece vede comparire le matrici L e C, poichè le equazioni dei componenti hanno infatti bisogno di un'integrazione, la quale ha dipendenza dalle derivate delle correnti e delle tensioni.

Il nuovo sistema di equazioni differenziali può essere riassunto in:

$$M_1 \cdot x + M_2 \cdot \dot{x} = r \quad (1.21)$$

Il metodo ottenuto per la risoluzione del circuito è il più generale possibile, come da obiettivo, e non presenta limiti di equazioni da cui può essere composto. Essendo equazioni differenziali del primo ordine si necessitano le condizioni iniziali.

Per lo studio che è stato implementato in questa tesi, i codici presenti nei capitoli successivi faranno riferimento, ove non specificato, a segnali sinusoidali; questo significa che verranno usati dei fasori e più nello specifico l'equazione (1.19) diverrà:

$$\begin{bmatrix} A & \emptyset & \emptyset \\ \emptyset & -\mathbb{1} & A^T \\ R & G & 0 \end{bmatrix} \cdot \begin{bmatrix} \bar{I}_{nb.1} \\ \bar{V}_{nb.1} \\ \bar{P}_{(nn-1).1} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ L & C & 0 \end{bmatrix} \cdot j\omega \cdot \begin{bmatrix} \bar{I}_{nb.1} \\ \bar{V}_{nb.1} \\ \bar{P}_{(nn-1).1} \end{bmatrix} = \begin{bmatrix} 0_{nb.1} \\ 0_{nb.1} \\ \bar{B}_{(nn-1).1} \end{bmatrix} \quad (1.22)$$

che può essere riassunto in:

$$M_1 \cdot \bar{x} + j\omega M_2 \cdot \bar{x} = \bar{r} \quad (1.23)$$

$$(M_1 \cdot + j\omega M_2 \cdot) \bar{x} = \bar{r} \quad (1.24)$$

$$\dot{M}_3 \cdot \bar{x} = \bar{r} \quad (1.25)$$



# Capitolo 2

## Model Order Reduction

Si affronta ora l'implementazione in Matlab dei codici di base necessari per sviluppare l'algoritmo del metodo di riduzione d'ordine (*MOR*) e ,in battuta successiva, applicarlo alla Tableau Analysis per verificarne il funzionamento .

### 2.1 Tableau Analysis in Matlab

Dopo aver esposto la parte teorica del metodo, d'ora in avanti si lavora in ambiente Matlab per la stesura degli script necessari. Dunque, si inizia implementando la Tableau Analysis al calcolatore, andando a costruire le matrici necessarie per la risoluzione dell'equazione matriciale in (1.22).

#### 2.1.1 Algoritmo principale

Il codice per la risoluzione dalla metodologia descritta nel capitolo precedente è il seguente:

```
1 clear all
2 clc
3
4 %prompt=input('inserire nome file dati per la netlist.txt ', 's')
5 %netlist=input(prompt)
6 %eval(prompt)
7 %load prompt
8 %load=input('netlist2.txt ')
9
```

```

10 load netlist2.txt
11 topo=netlist2(:,1:2); %informazioni topologiche
12 tipo=netlist2(:,3:4); %informazioni tipologiche
13 %1=resistenza; 2=induttanza; 3=capacit ; 4=generatore di
    tensione
14 nb=size(topo,1); %numero di lati (branches)
15 nn=max(topo(:,1)); %numero di nodi (nodes)
16
17 %Creazione di A
18 A=zeros(nn,nb); %preallocazione di A
19 for k=1:nb
20     A(topo(k,1),k)=1;
21     A(topo(k,2),k)=-1;
22 end
23 A=A(1:nn-1,:);
24
25 %Creazione di R,G,L,C,b
26 R=zeros(nb,nb);
27 G=zeros(nb,nb);
28 L=zeros(nb,nb);
29 C=zeros(nb,nb);
30 b=zeros(nb,1);
31 for k=1:nb
32     if netlist2(k,3)==1;
33         R(k,k)=-netlist2(k,4);
34         G(k,k)=1;
35         L(k,k)=0;
36         C(k,k)=0;
37         b(k,1)=0;
38     elseif netlist2(k,3)==2;
39         R(k,k)=0;
40         G(k,k)=1;
41         L(k,k)=-netlist2(k,4);
42         C(k,k)=0;
43         b(k,1)=0;
44     elseif netlist2(k,3)==3;
45         R(k,k)=1;
46         G(k,k)=0;
47         L(k,k)=0;
48         C(k,k)=-netlist2(k,4);
49         b(k,1)=0;
50     elseif netlist2(k,3)==4
51         R(k,k)=0;
52         G(k,k)=1;

```



```

53     L(k,k)=0;
54     C(k,k)=0;
55     b(k,1)=-netlist2(k,4);
56     else
57         R(k,k)=0;
58         G(k,k)=0;
59         L(k,k)=0;
60         C(k,k)=0;
61         b(k,1)=0;
62     end
63 end
64
65 %creazione di M1,M2,M3
66 M1=zeros(nn-1+nb+nb,nb+nb+nn-1);
67 M1=[A zeros(nn-1,nb) zeros(nn-1); zeros(nb) -speye(nb) A'; R G
     zeros(nb,nn-1)];
68 M2=[zeros(nn-1,nb) zeros(nn-1,nb) zeros(nn-1); zeros(nb) zeros(nb)
     ) zeros(nb,nn-1); L C zeros(nb,nn-1)];
69 M3=M1+j*1*M2; %risultato con omega=1
70 %creazione di r
71 r=zeros(nn-1+nb+nb,1);
72 r=[zeros(nn-1+nb,1);b];
73 %risoluzione del sistema
74 x=M3\r;
75
76 %creazione di x al variare di omega
77 omega=linspace(-10,10);
78 size=size(omega);
79 nomega=size(1,2);
80 for KK=1:nomega
81 M=M1+j*omega(1, KK)*M2;
82 x2=M\r;
83 s(:,KK)=x2;
84 end
85 %plot di x al variare di omega
86 smodulo=abs(s);
87 plot(omega, smodulo(1,:)) %modulo corrente nodo da 1 a 2

```

Listing 2.1: Tableau Analysis in Matlab.

Inizialmente si carica la netlist, scritta precedentemente in un file di testo, come matrice nel programma; successivamente si definiscono, sfruttando le informazioni fornite dalla netlist stessa, il numero di lati ed il numero di nodi del circuito.

Fatto ciò, si iniziano a comporre le matrici necessarie per risolvere 1.22. La prima è la matrice di incidenza ridotta  $A$ . Successivamente, tramite l'utilizzo di cicli **if** vengono riconosciuti i tipi di componenti del circuito tramite i dati della netlist e quindi costruite le matrici  $R, L, G, C$  ed il vettore  $b$ . Per esemplificazione si ipotizzino degli elementi circuitali tramite una netlist per il circuito 1.3:

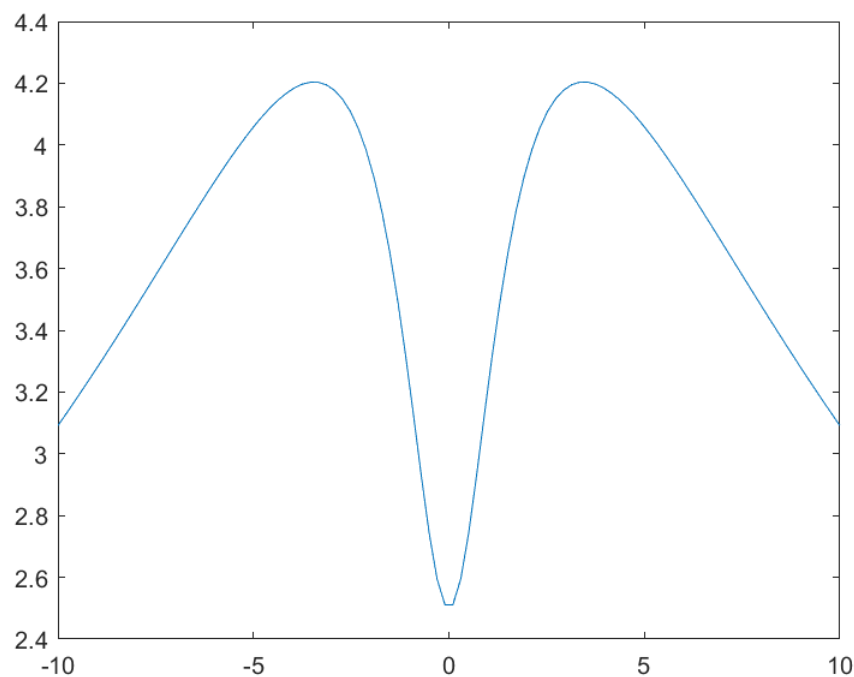
starting node	ending node	type	value
1	2	4	20
2	3	3	0.06
3	4	1	10
2	4	1	8
4	1	2	0.5

**Tabella 2.1:** Netlist 2.

Dopo che tutte le matrici sono state composte, a riga 74 del codice, viene calcolato il vettore soluzione del circuito per  $\omega = 1$ .

$$x = \begin{bmatrix} 3.1487 + 0.6584i \\ 0.6076 + 0.8552i \\ 0.6076 + 0.8552i \\ 2.5412 - 0.1968i \\ 3.1487 + 0.6584i \\ -20.0000 + 0.0000i \\ 14.2534 - 10.1264i \\ 6.0758 + 8.5520i \\ 20.3292 - 1.5744i \\ -0.3292 + 1.5744i \\ 0.3292 - 1.5744i \\ 20.3292 - 1.5744i \\ 6.0758 + 8.5520i \end{bmatrix} \quad (2.1)$$

Al fine di avere una panoramica più completa, da riga 77 a riga 87, viene eseguito un grafico cartesiano avente come ascisse valori di  $\omega$  compresi tra -10 e 10: il comando *linspace* crea un vettore di 100 elementi (se non specificato) equidistanti da due estremi indicati. In ordinata viene indicato il modulo della corrente uscente dal nodo 1 ed entrante nel nodo 2, per ogni valore di  $\omega$ . Il grafico ottenuto sarà il seguente:



**Figura 2.1:** Modulo corrente lato 1

Si possono fare delle osservazioni per verificare la validità del codice: per  $\omega = 0$  si ha il minimo del valore del modulo della corrente uscente dal nodo 1. Difatti, l'impedenza  $Z_{tot}$  ha un massimo proprio in  $\omega = 0$ , come mostrato in. Andando a calcolare matematicamente si trovano anche i due massimi mostrati in figura 2.1. Si pone la derivata del modulo dell'impedenza equivalente del circuito uguale 0:

$$\frac{d}{dt} \left| \frac{(10 - \frac{i}{0.06\omega}) \cdot 8}{(10 - \frac{i}{0.06\omega}) + 8} + 0.5\omega i \right| = 0 \quad (2.2)$$

il che darà come soluzioni  $\omega = 3.45122$  e  $\omega = -3.45122$ .

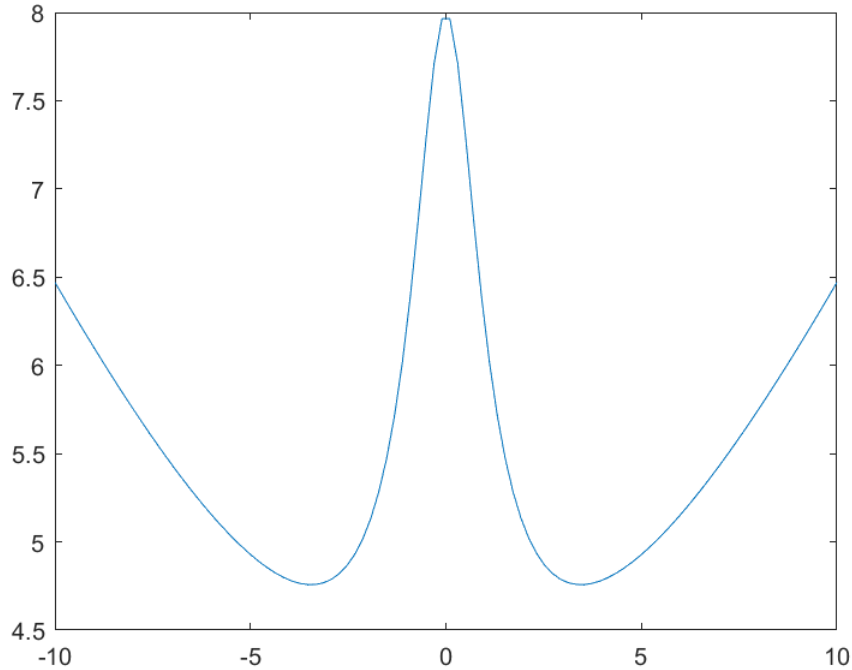


Figura 2.2: Impedenza totale

### 2.1.2 Algoritmo modificato

Per ovviare a problemi di compatibilità con la creazione del MOR in seguito, si sceglie un metodo alternativo ugualmente valido per la risoluzione della Tableau Analysis. Sarà questo algoritmo quello che verrà utilizzato come punto di partenza per creare il codice di riduzione d'ordine (MOR).

La struttura delle matrici utilizzate all'interno del metodo alternativo rimane la medesima vista in precedenza, tuttavia l'equazione matriciale che risolve il problema circuitale sarà la seguente:

$$\begin{bmatrix} R + j\omega L + \frac{j}{\omega C} & A^T \\ A & 0 \end{bmatrix} \cdot \begin{bmatrix} \bar{I} \\ \bar{P} \end{bmatrix} = \begin{bmatrix} \bar{B} \\ 0 \end{bmatrix} \quad (2.3)$$

Come si può notare, in questa risoluzione, i potenziali nodali sono state correlati

alle cadute di tensione sui componenti per dare luogo ad un'unica equazione. A differenza di quanto visto precedentemente non sarà più necessaria la matrice  $G$ , il che significa che ci si limiterà ad analizzare circuiti le cui alimentazioni sono assimilabili a dei soli generatori di tensione. Il codice è mostrato di seguito:

```
1 clear all
2 clc
3
4 %prompt=input('inserire nome file dati per la netlist.txt ', 's')
5 %netlist=input(prompt)
6 %eval(prompt)
7 %load prompt
8 %load=input('netlist2.txt ')
9
10 load netlist.txt
11 topo=netlist(:,1:2); %informazioni topologiche
12 tipo=netlist(:,3:4); %informazioni tipologiche
13 %1=resistenza; 2=induttanza; 3=capacit ; 4=generatore di
    tensione
14 nb=size(topo,1); %numero di lati (branches)
15 nn=max(topo(:,1)); %numero di nodi (nodes)
16
17 %Creazione di A
18 A=zeros(nn,nb); %preallocazione di A
19 for k=1:nb
20     A(topo(k,1),k)=1;
21     A(topo(k,2),k)=-1;
22 end
23 A=A(1:nn-1,:);
24
25 %Creazione di R,G,L,C,b
26 R=zeros(nb,nb);
27 L=zeros(nb,nb);
28 C=zeros(nb,nb);
29 b=zeros(nb,1);
30 for k=1:nb
31     if netlist(k,3)==1;
32         R(k,k)=netlist(k,4);
33         L(k,k)=0;
34         C(k,k)=0;
35         b(k,1)=0;
36     elseif netlist(k,3)==2;
37         R(k,k)=0;
```

```

38     L(k,k)=netlist(k,4);
39     C(k,k)=0;
40     b(k,1)=0;
41     elseif netlist(k,3)==3;
42         R(k,k)=0;
43         L(k,k)=0;
44         C(k,k)=1/netlist(k,4);
45         b(k,1)=0;
46     elseif netlist(k,3)==4
47         R(k,k)=0;
48         L(k,k)=0;
49         C(k,k)=0;
50         b(k,1)=netlist(k,4);
51     else
52         R(k,k)=0;
53         L(k,k)=0;
54         C(k,k)=0;
55         b(k,1)=0;
56     end
57 end
58
59 %creazione di M1,M2,M3
60 M1=zeros(nn-1+nb,nb+nn-1);
61 M1=[R 0*A'; A zeros(nn-1)];
62 M2=[j*L A'+zeros(nb,nn-1); zeros(nn-1,nb) zeros(nn-1)];
63 M3=[C/j zeros(nb,nn-1); zeros(nn-1,nb) zeros(nn-1)];
64
65 M=M1+omega*M2+M3/omega;
66
67 %creazione di r
68 r=zeros(nn-1+nb,1);
69 r=[b; zeros(nn-1,1)];
70
71 % soluzione del sistema
72 x=M\r;

```

Listing 2.2: Tableau Analysis modificata in Matlab.

Come si può notare da riga 59 a riga 63 la matrice principale viene suddivisa in 3 sottomatrici: questo passaggio viene eseguito per poter studiare la variazione del parametro variabile  $\omega$  per agevolare e rendere più semplice la creazione dello script per il MOR successivamente; sono state create quindi:

$$M_1 = \begin{bmatrix} R & 0 \\ A & 0 \end{bmatrix}; \quad M_2 = \begin{bmatrix} jL & A_T \\ 0 & 0 \end{bmatrix}; \quad M_3 = \begin{bmatrix} \frac{C}{j} & 0 \\ A & 0 \end{bmatrix}; \quad (2.4)$$

questa variazione porta a considerare la matrice principale  $M$  come somma delle 3 matrici mostrate in 2.4:

$$M = M_1 + \omega M_2 + \frac{M_3}{\omega} \quad (2.5)$$

## 2.2 Model Order Reduction: contenuti e finalità del metodo

Si ponga come caso di studio un circuito molto complesso, con rango delle matrici quindi elevato. Questo fattore comporta tempistiche e costi computazionale per eseguire il programma molto importanti, sia in termini di memoria, perchè le variabili devono poter essere memorizzate, sia in termini di tempo effettivo di calcolo [9].

Proprio questi fattori conducono alla necessità di sviluppare modelli di calcolo alternativi [6], in grado di moderare le problematiche che derivano dagli aspetti elencati sopra. In questo lavoro di tesi, si cercherà di sviluppare un codice, a partire dall'analisi di Tableau vista precedentemente, che cerchi di soddisfare più possibile tale necessità: per ciò, si parlerà di Metodo di Riduzione d'Ordine del Modello, o *Model Order Reduction* [7] [8]. Nel caso che si prenderà in esame, l'implementazione del codice è stata eseguita valutandone un parametro variabile, ossia la frequenza  $\omega$ ; si potrà assumere che il metodo sarà valido entro uno specifico range di frequenza  $\Delta\omega$  considerata.

In virtù di questo fattore si tratterà di *Parametric Model Order Reduction* [10].

### 2.2.1 Implementazione MOR alla Tableau Analysis

In questa sottosezione si espone e si discute il contenuto in dell'algoritmo MOR *step by step*, il quale verrà creato in seguito in Matlab:

---

#### Algorithm 1 PARAMETRIC MODEL ORDER REDUCTION

---

**Input:** matrici  $M1, M2, M3$ ; vettore  $r$ ;

**Output:** matrici  $M1p, M2p, M3p$ ; vettore  $rp$ ; counter  $c$ ;  
matrice di cambiamento di base  $V$ ;

```

1:  $V_0 = \emptyset \leftarrow$  assegno valore iniziale
2:  $c = 1 \leftarrow$  assegno valore iniziale
3:  $\omega = 1 \leftarrow$  assegno valore iniziale
4:  $re = +\infty \leftarrow$  assegno valore iniziale
5:  $\eta \leftarrow$  assegno valore di tolleranza
6:  $N_{rand} \leftarrow$  assegno valore
7:
8: while  $re > \eta$  do
9:    $M = M_1 + \omega M_2 + \frac{M_3}{\omega}$ 
10:   $x(\omega) = M \setminus r \leftarrow$  trovo soluzione con  $\omega$  assegnata
11:   $V_c = GSO([V_{c-1}, x(\omega)]) \leftarrow$  aggiorno base ortonormale
12:   $M1p = V_c^T \times M1 \times V_c \leftarrow$  genero il
13:   $M2p = V_c^T \times M2 \times V_c \leftarrow$  modello d'ordine ridotto
14:  ...
15:  genero  $N_{rand}$  valori di  $\omega$ 
16:  for  $h=1, \dots, N_{rand}$  do
17:    seleziono l' $h_{th}$  valore random di  $\omega$ , e trovo la soluzione
18:     $\hat{x}(\omega_{th})$  del problema di ordine ridotto, poi calcolo
19:     $\tilde{x}(\omega_{th}) = V_c \hat{x}(\omega_{th})$  la soluzione approssimata, infine valuto
20:    il residuale  $r'_h = \frac{\|M_{\omega_{th}} \tilde{x}(\omega_{th}) - r\|}{\|r\|}$  per ogni valore  $\omega_{th}$ 
21:  end for
22:
23:  Trovo valore massimo tra i residuali ottenuti
24:   $r'_M = \max(r'_h)$  dove  $h$  varia da 1 a  $N_{rand}$ 
25:   $r = r'_M \leftarrow$  aggiorno il residuale per il ciclo successivo
26:   $\omega = \omega_M \leftarrow$  aggiorno  $\omega$ 
27:  if  $re > \eta$  then
28:     $c=c+1$ ;
29:  end if
30: end while

```

---



Come primo step, si inizializzano alcune variabili necessarie l'implementazione del codice:

- la matrice di proiezione  $V_c$  è settata come insieme vuoto.
- il contatore  $c$ , che identifica il numero di iterazioni del sistema, ma anche le nuove dimensioni delle matrici del modello d'ordine ridotto, è inizializzato a 1.
- l'errore è inizializzato a  $+\infty$  in modo che il ciclo **while** successivo sicuramente abbia un'incipit.
- $N_{rand}$  viene settato, e indica il numero di valori randomici del parametro di riferimento, in questo caso la pulsazione, che verranno generati ad ogni iterazione. C'è da precisare che più valori vengono generati e più grande è il range entro il quale vengono generati tali valori, più il metodo risulterà preciso una volta ottenuto il nuovo modello di ordine ridotto.
- viene settato un valore di tolleranza  $\eta$  entro il quale i risultati del codice vengono considerati accettabili: se l'errore sarà infatti maggiore della tolleranza il ciclo proseguirà, come indicato dal ciclo **if** a righe 27, 28, 29

La soluzione del problema per una data  $\omega$  viene calcolata.

Successivamente la matrice di proiezione  $V_c$  viene valutata applicando un codice di 'Ortogonalizzazione di Gram-Schmidt' [12] al set di vettori delle soluzioni composto dalle soluzioni  $x(\omega)$ . La matrice  $V_c$  sarà composta da vettori basi ortonormali dello spazio generato dalle soluzioni di riga 10 trovate ad ogni iterazione; essa avrà come dimensioni  $N_x \times c$ , dove  $N_x$  è la dimensione del vettore soluzione del problema.

Il modello d'ordine ridotto è ora costruito proiettando le matrici  $M_1, M_2, M_3$  e il vettore dei termini noti  $r$  sullo spazio vettoriale delimitato da  $V_c$ . Le matrici  $M_{1p}, M_{2p}, M_{3p}$  del nuovo modello avranno dimensioni  $c \times c$  e il vettore  $r_p$  avrà dimensione  $c \times 1$ . Il modello di ordine ridotto di dimensione  $c$  è ora ottenuto, e viene utilizzato per ottenere la soluzione approssimata  $\tilde{x}$  del problema parametrico iniziale; successivamente, il nuovo modello è testato su un numero  $N_{rand}$  di parametri randomici, per verificarne l'accuratezza. Se il residuale, ossia lo scostamento tra la soluzione approssimata e quella originale, è inferiore alla tolleranza, allora il codice si ferma e il modello è corretto. In caso contrario, il

residuale che massimizza lo scostamento è selezionato come prossimo residuale per la successiva iterazione, e il contatore  $c$  viene aumentato di un'unità: grazie all'ortonormalizzazione di Gram-Schmidt solo le componenti ortogonali delle soluzioni  $x(\omega)$  utilizzate nella GSO saranno considerate per aggiornare la matrice di proiezione  $V_c$ .

Si espone ora la stesura del codice per produrre il nuovo modello, avendo i dati di input, in Matlab:

```

1 function [M1p,M2p,M3p,rp,V,c,omegav] = MOR(M1,M2,M3,r,Nrand)
2
3 V=[];
4 c=1; % counter
5 omega=1;
6 re=inf; % residuale del problema approssimato (
   inizializzato a +inf)
7 eta=1e-3; % errore accettabile dal problema
8
9
10
11 % inizio ciclo while che da soluzioni finch la soluzione
   approssimata del MOR ha
12 % un errore apprezzabile
13
14 while re>eta
15
16     % calcolo in primis il risultato x con le matrici "normali"
   del
17     % problema
18
19     M=M1+(omega.*M2)+(M3./omega);
20     x=M\r;
21
22     % ortonormalizzo il risultato x in nuove basi ortonormali
23     V=mgson([V,x]);
24
25     % cambiamento di basi di M1,M2,M3 ed r
26     M1p=V'*M1*V;
27     M2p=V'*M2*V;
28     M3p=V'*M3*V;
29     rp=V'*r;
30

```

```
31 % genero Nrand valori casuali di omega e ne creo un vettore
32 % in questo caso da 0 a 1000
33 omegav=1+999*rand(Nrand,1);
34
35 % inizializzo vettore dei residuali che andr a calcolare
36 % e xp vettore della soluzione con basi ortonormali
37 rh=zeros(Nrand,1);
38
39 for h=1:Nrand
40
41     % seleziono l'h-esimo valore di omega generato
42     omega=omegav(h);
43
44     % trovo la soluzione in pMOR per ogni h
45     Mp=M1p+(omega.*M2p)+(M3p./omega);
46     xp=Mp\rp;
47     M=M1+omega*M2+M3/omega;
48
49     % trovo soluzione approssimata di dimensioni (2nb+n-1,1)
50     xh=V*xp;
51
52     % creo vettore dei residuali per ogni soluzione
53     % approssimata
54     rh(h)=(norm(r))\((norm(M*xh-r));
55
56 end
57
58 % setto il residuale maggiore tra quelli ottenuti per la
59 % success iterazione
60 [rM, posh]=max(rh);
61 re=rM;
62
63 % setto come omega l'omega del residuale maggiore
64 omega=omegav(posh);
65
66 % verifico se il residuale preso ancora maggiore dell'
67 % incertezza massima desiderata
68 % e se lo , aumento di 1 il counter c per la successiva
69 % iterazione
70
71 if re>eta
72     c=c+1;
73 end
```

```
71 end
72 end
```

**Listing 2.3:** Script MOR per la Tableau Analysis modificata

A riga 23 si trova per l'appunto la function *mgson*, intesa essere la funzione di ortogonalizzazione di Gram-Schmidt; questa ci consente di creare dei vettori ortogonali all'insieme di vettori di partenza, che sono in questo caso le soluzioni del problema iniziale ad alcuni valori di  $\omega$ . Il codice viene riportato di seguito:

```
1 function [Q, R] = mgson(X)
2 % Modified Gram-Schmidt orthonormalization (numerical stable
3   version of Gram-Schmidt algorithm)
4 [d,n] = size(X);
5 m = min(d,n);
6 R = zeros(m,n);
7 Q = zeros(d,m);
8 for i = 1:m
9     v = X(:,i);
10    for j = 1:i-1
11        R(j,i) = Q(:,j)'*v;
12        v = v-R(j,i)*Q(:,j);
13    end
14    R(i,i) = norm(v);
15    Q(:,i) = v/R(i,i);
16 end
17 R(:,m+1:n) = Q'*X(:,m+1:n);
```

**Listing 2.4:** Script ortogonalizzazione Gram-Schmidt

# Capitolo 3

## Applicazione del MOR a un circuito generico

### 3.1 Analisi circuito con risonanza parallelo

Si affronta ora un'esempio di applicazione del programma di riduzione d'ordine ad un caso di circuito analizzato con l'algoritmo di Tableau Analysis semplificato 2.3.

#### 3.1.1 Circuito preso in esame

Si prende in considerazione un semplice circuito:

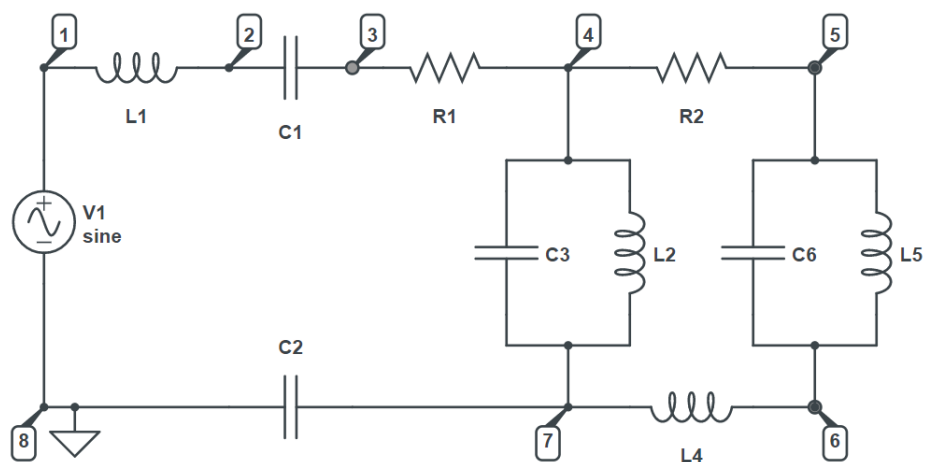


Figura 3.1: Circuito di prova

composto da elementi base quali induttori, resistori, capacità e un'alimentazione a tensione impressa sinusoidale. Si noti che il nodo 8 è stato considerato come nodo di terra, quindi posto al potenziale nullo; ciò corrisponde allo stesso tempo all'eliminazione di una riga (l'ultima) dalla matrice di incidenza  $A_c$  per ottenere la matrice d'incidenza ridotta  $A$  che rappresenta il circuito.

Vengono quindi dati dei valori agli elementi circuitali, i quali vengono rappresentati dalla seguente netlist:

starting node	ending node	type	value
1	2	L	0.2
2	3	C	$10^{-6}$
3	4	R	20
4	5	R	10
4	7	L	0.05
4	7	C	$6 \cdot 10^{-4}$
5	6	L	0.05
5	6	C	$6 \cdot 10^{-4}$
6	7	L	0.04
7	8	C	$2 \cdot 10^{-4}$
8	1	V	200

**Tabella 3.1:** Netlist del circuito di figura 3.1.

Il circuito è composto da 8 nodi e 11 lati, considerando che tra i nodi  $4 \rightarrow 7$  e tra i nodi  $5 \rightarrow 6$  si trovano collegati 2 bipoli in parallelo: questi elementi, in particolare, assieme ad altri del circuito sono interessanti dal punto di vista dell'analisi alle frequenze, per un'eventuale presenza di risonanze.

Dalla semplice applicazione della Tableau Analysis ci si aspetta di ottenere delle matrici di dimensione:

$$m \times n = 18 \times 18 \quad (3.1)$$

Si va a confrontare le risoluzioni del circuito con e senza applicazione del metodo di riduzione d'ordine.

### 3.1.2 Confronto risultati con e senza MOR

Si applichi ora il metodo pMOR visto nel capitolo precedente al circuito preso in analisi. Vengono forniti come dati:

- $N_{rand} = 1000$
- $\eta = 10^{-3}$

#### ESECUZIONE DEL CODICE

...  
...

Esso termina dopo alcune iterazioni e da risultato positivo; ciò significa che si è riusciti a costruire il modello ricercato, costituito dalle matrici e vettori di ordine ridotto, che può essere utilizzato per poter operare con il circuito in alternativa al modello di "base". Si osservino ora i risultati ottenuti, in particolare:

- $c = 8$
- $V = 18 \times 7$
- dimensione  $M_{1p} = 8 \times 8$
- dimensione  $M_{2p} = 8 \times 8$
- dimensione  $M_{3p} = 8 \times 8$
- dimensione  $r_p = 8 \times 1$

Come si può notare, sono state ottenute delle matrici quadrate di ordine 8; l'obiettivo di ridurre la memoria impiegata per la memorizzazione è stato quindi raggiunto e dimostrato da ciò. Una volta ottenuti i risultati nello spazio vettoriale in cui si collocano le matrici del nuovo modello, la matrice  $V$  impiegata per la trasformazione lineare sarà necessaria per ritornare allo spazio vettoriale originario e ottenere la vera soluzione con il significato fisico che ricerchiamo.

Si avrà dunque:

$$x_p = M_p \backslash r_p \quad (3.2)$$

e, per ottenere il vettore soluzione reale:

$$x = V \times x_p \quad (3.3)$$

Per ottenere una visione più completa dell'efficacia del metodo di riduzione, si scrive un codice per raffigurare i risultati nelle due casistiche di metodo utilizzato. Il parametro preso in considerazione per il confronto è l'impedenza del circuito vista dal generatore  $V$ , e si analizzano di questo modulo e fase. Il range di pulsazione scelto sarà

$$1 < \omega < 10000 \quad (3.4)$$

ed in particolare si prende un vettore di  $10^4$  elementi, ossia tutti i numeri interi compresi nell'intervallo.

```

1 % confronto delle soluzioni usando prima le matrici normali e poi
  quelle di dimensione ridotta
2
3 % preallocazione dei vettori
4
5 omega=linspace(1,10000,10000);
6 curr=zeros(10000,1);
7 currp=zeros(10000,1);
8 z=zeros(10000,1);
9 zp=zeros(10000,1);
10
11
12 % creazione del vettore dell'impedenza vista dal
  generatore al
13 % variare della pulsazione, con e senza applicazione del MOR

```



```
14
15 for q=1:10000
16
17     M=M1+(omega(q).*M2)+(M3./omega(q));
18     x=M\r;
19     curr(q)=x(1);
20     z(q)=20/curr(q);
21
22     Mp=M1p+(omega(q).*M2p)+(M3p./omega(q));
23     xp=Mp\rp;
24     xtilda=V*xp;
25     currp(q)=xtilda(1);
26     zp(q)=20/currp(q);
27
28 end
29
30 % prendo il modulo dei vettori complessi dell'impedenza vista dal
31 % generatore
32
33 absz=abs(z);
34 abszp=abs(zp);
35
36 % plot dei due risultati con i due metodi per vederne l'effettivo
37     errore
38
39 figure(1)
40 loglog(omega,absz,"m-","LineWidth",1)
41 axis([10 2000 -100 1000])
42 xlabel('valore della pulsazione')
43 ylabel('modulo impedenza')
44 title('IMPEDENZA VISTA DAL GENERATORE')
45 grid on
46 hold on
47 plot(omega,abszp,"g--","LineWidth",1);
48 hold off
49
50 % plotto la fase dell'impedenza vista dal generatore al variare
51     di omega
52
53 phiz=angle(z)*360/pi;
54 phizp=angle(zp)*360/pi;
55
56 figure(2)
57 semilogx(omega,phiz,"r-","LineWidth",0.8)
```

```

56 axis([10 2000 -180 +180])
57 xlabel('valore della pulsazione')
58 ylabel('fase impedenza')
59 title('IMPEDENZA VISTA DAL GENERATORE')
60 grid on
61 hold on
62 plot(omega, phiz, "g--", "LineWidth", 0.5)
63 hold off

```

**Listing 3.1:** Script per il plotting dei risultati

Il codice restituisce due grafici:

- nella figura 3.2 si trova l'andamento della fase dell'impedenza vista dal generatore per ogni valore di  $\omega$ , con l'utilizzo del MOR in verde e con l'utilizzo del procedimento di Tableau Analysis standard in rosso.
- nella figura 3.3 si trova l'andamento del modulo dell'impedenza vista dal generatore per ogni valore di  $\omega$ , con l'utilizzo del MOR in verde e con l'utilizzo del procedimento di Tableau Analysis standard in magenta.

Si possono notare due punti in cui si evidenziano effetti di risonanza tra gli elementi del circuito. In riferimento al grafico 3.1, tra  $L_1$  e  $C_1$  ci sarà una risonanza serie per:

$$\omega_1 = \frac{1}{\sqrt{L_1 \cdot C_1}} \simeq 707 \frac{rad}{s} \quad (3.5)$$

mentre prendendo ad esame le coppie  $L_2$  con  $C_3$  e  $L_5$  con  $C_6$ , si avrà come è possibile notare dal circuito un valore di pulsazione che corrisponderà ad una doppia risonanza parallelo: questo fenomeno, mostrato chiaramente nelle due figure successive, darà luogo ad un punto di discontinuità poichè nell'intorno della pulsazione  $\omega_2$  le due coppie di bipoli in parallelo si assimilano ad un circuito aperto. La pulsazione in particolare sarà:

$$\omega_2 = \frac{1}{\sqrt{L_2 \cdot C_3}} = \frac{1}{\sqrt{L_5 \cdot C_6}} \simeq 182 \frac{rad}{s} \quad (3.6)$$

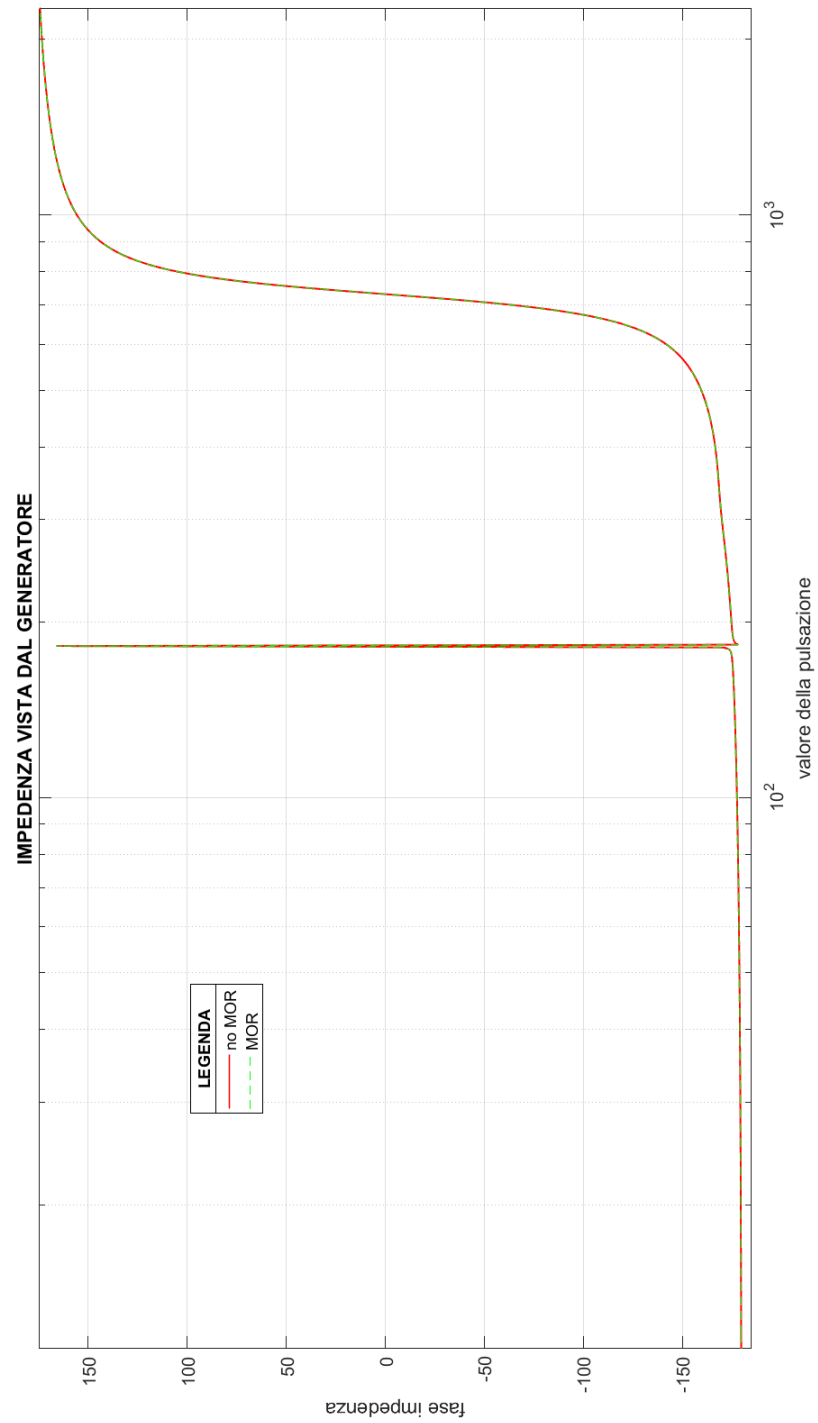


Figura 3.2: Fase impedenza vista dal generatore

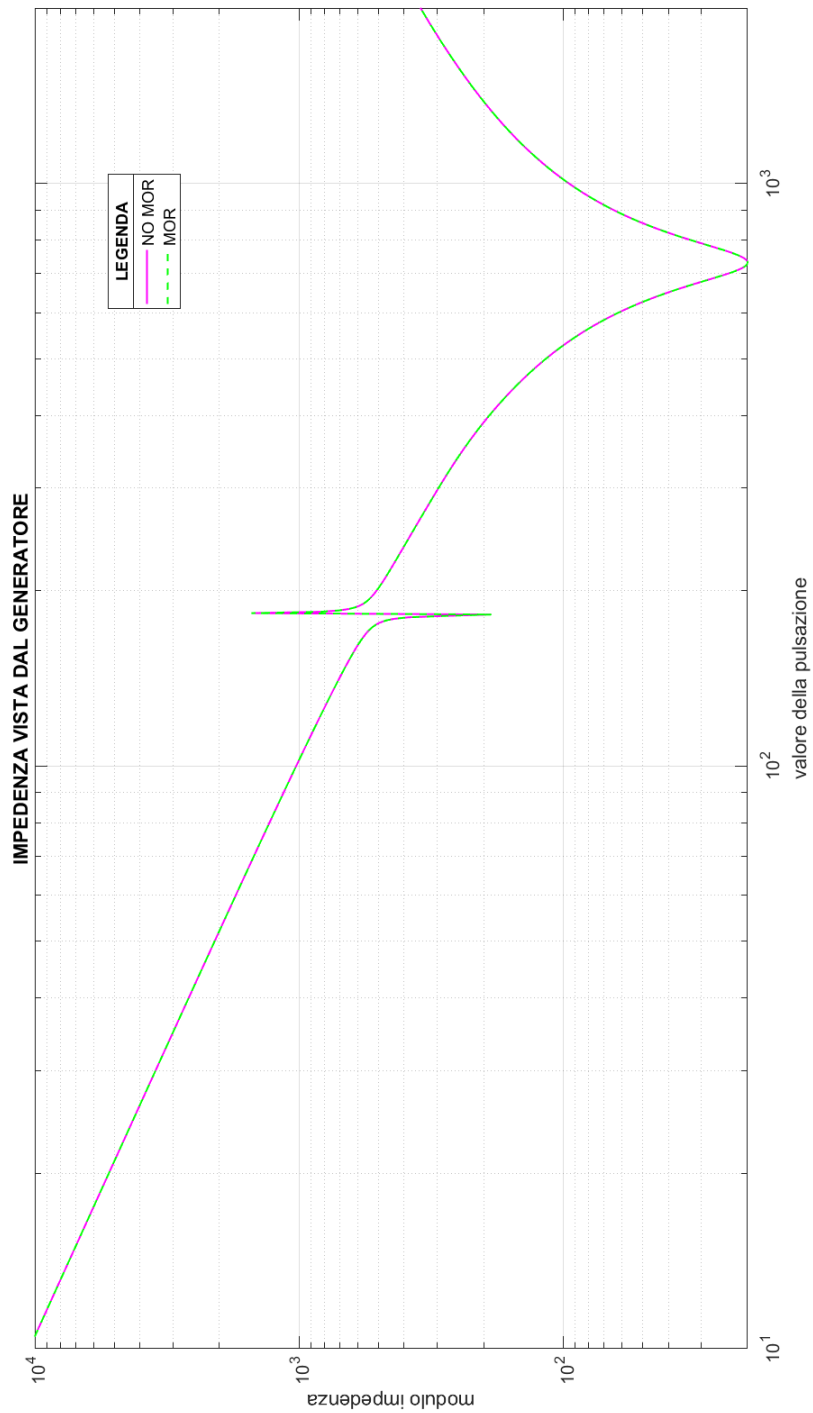


Figura 3.3: Modulo impedenza vista dal generatore

L'implementazione dei due metodi allo stesso caso di studio e gli andamenti nelle due figure sopra riportate, portano a fare un'osservazione importante: i due grafici, con e senza utilizzo del metodo di riduzione d'ordine, sono pressochè sovrapposti sia nel modulo che nella fase. Questo è dovuto al fatto che la semplicità del circuito è alta, per cui il modello d'ordine ridotto ottenuto è molto preciso, soprattutto nel range di pulsazioni con le quali è stato costruito (algoritmo ??).

A prova di quanto detto si trova che lo scarto medio che si ha tra le due soluzioni, chiamato qui  $\sigma$ , è molto piccolo, tale da non essere percepito visivamente nel grafico:

$$\sigma_Z = |z - z_p| \approx 10^{-8} \div 10^{-14} \quad (3.7)$$

## 3.2 Analisi linea di trasmissione elettrica a costanti concentrate

Per evidenziare in modo più chiaro i vantaggi dell'utilizzo del codice di riduzione d'ordine del modello, è necessario prendere in considerazione un caso di studio il cui un sistema di equazioni sia molto più grande del circuito precedentemente visto; maggiore è la quantità di elementi, nodi e lato del circuito, maggiore sarà l'utilità del MOR per la risoluzione.

Un potenziale caso di studio ci viene dallo studio delle linee di trasmissione elettrica. Tuttavia, bisogna fare delle considerazioni [13] prima di procedere alla stesura degli script su cui operare con il MOR [11].

### 3.2.1 Modello a costanti concentrate

Per quanto riguarda lo studio delle linee elettriche, in questo caso aeree, da letteratura si possono utilizzare due metodi:

- MODELLO A COSTANTI DISTRIBUITE

Questo metodo è il più utilizzato poichè è il più preciso e attendibile dal punto di vista teorico e matematico. Consiste nello studio di un sistema di equazioni differenziali che tengono conto anche delle condizioni al contorno (come carichi o generatori presenti in linea).

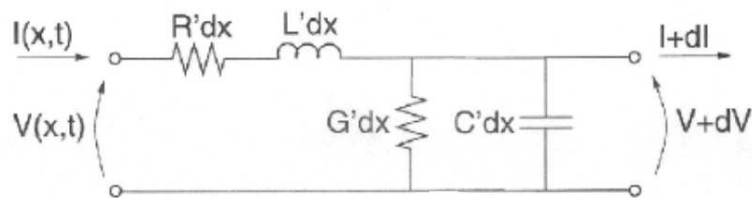


Figura 3.4: Blocco tipico modello a costanti distribuite

Le tensioni e le correnti sono qui in funzione del tempo e dello spazio, mentre le costanti di rete quali resistenze, conduttanze, induttanze e capacità sono calcolate e studiate per unità di lunghezza. Questo modello viene studiato tramite le equazioni dei telegrafisti:

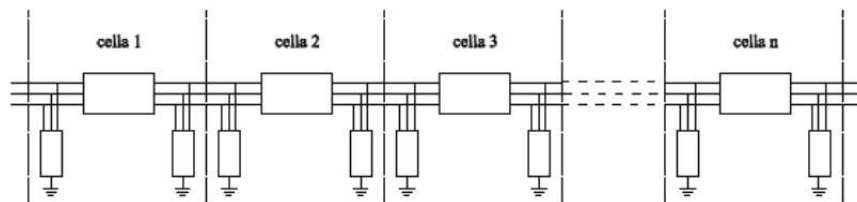
$$\begin{cases} \frac{\partial V^2(z,t)}{\partial z^2} = L \cdot C \cdot \frac{\partial V^2(z,t)}{\partial t^2} \\ \frac{\partial I^2(z,t)}{\partial z^2} = L \cdot C \cdot \frac{\partial I^2(z,t)}{\partial t^2} \end{cases} \quad (3.8)$$

dove L e C sono l'induttanza e la capacità per unità di lunghezza.

Nello studio compiuto tramite il codice MOR non viene preso in considerazione: si tratterebbe di sviluppare altrimenti un modello basato sullo sviluppo di un codice di riduzione dell'ordine ad un sistema di variabili dipendenti da più di un parametro, quando in questo lavoro di tesi si è trattata la dipendenza da un parametro solamente, ossia la pulsazione nel caso di regime sinusoidale. Per questo motivo nell'analisi di questo capitolo utilizzeremo il modello a costanti concentrate.

- MODELLO A COSTANTI CONCENTRATE

Questo modello si basa sullo studio delle grandezze di linea dipendenti solamente dal tempo. Si utilizzano qui le leggi di Kirchhoff e le leggi di Ohm, quindi in linea con lo studio della Tableau Analysis. Lo schema che viene visualizzato si basa sulla divisione della linea in "celle" come mostrato in figura che segue:



**Figura 3.5:** Modello a costanti concentrate

Sebbene nella figura 3.5 lo schema degli elementi di rete per ogni cella sia raffigurato a  $\Pi$ , si utilizzerà lo schema a T per la stesura del codice.

### 3.2.2 Script per costruzione netlist della linea

Si passa alla stesura dello script necessario per creare la netlist di una linea elettrica a costanti concentrate.

Come detto precedentemente, la netlist deve essere in grado di descrivere il circuito equivalente della linea a prescindere dal numero di "celle" in cui si voglia suddividere la linea. Questa implica la necessità dello script di creazione netlist di essere flessibile.

Lo schema di riferimento su cui lavorare per costruirlo è il seguente:

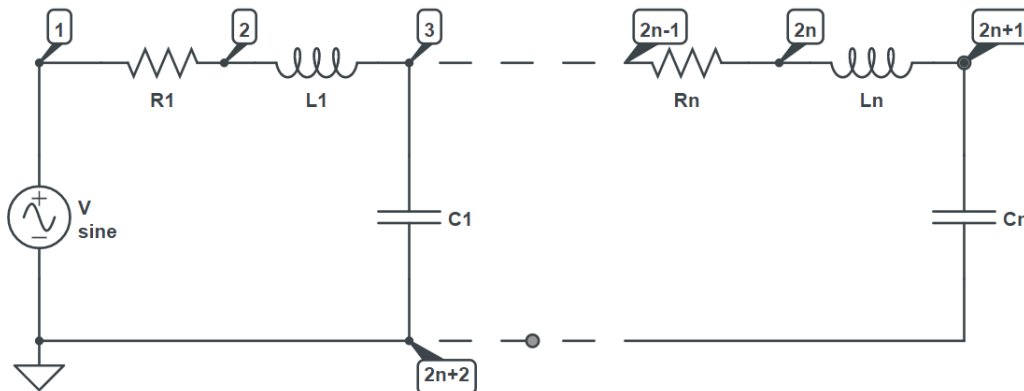


Figura 3.6: Schema di riferimento per costruzione netlist linea elettrica

Per semplificazione e per necessità, utilizzando l'algoritmo 2.3, si è considerata trascurabile la conduttanza trasversale lungo la linea.

Seguendo la logica di numerazione dei nodi, come mostrato in figura 3.6, si scrive lo script in Matlab per la generazione della netlist:

```

1 fid=fopen('netlist4.txt','w');
2 A=[];
3
4 % chiedo in input il numero di blocchi rlc in cui voglio
   suddividere la mia
5 % linea di trasmissione (es.) da modellizzare
6
7 n=input(' numero di blocchi rlc = ');
8
9 % penso di dividere la netlist che si otterrà in due blocchi

```



```
10 % un primo avente 2n righe che corrispondono agli elementi in
    serie
11 % un secondo avente (3n+1)-(2n)=n+1 righe corrispondenti agli
    elementi
12 % parallelo + alimentazione
13 % eventuale valore del carico verr aggiunto successivamente
14
15 % setto i valori degli elementi del circuito che si ripeteranno
    per ogni
16 % blocco
17
18 res=input('valore resistenza serie = ');
19 indut=input('valore impedenza induttiva serie = ');
20 capac=input('valore impedenza capacitiva parallelo = ');
21 alim=input('setto valore di tensione alimentazione = ');
22
23 % inizio a creare la matrice
24
25 for k=1:2*n
26     A(k,1)=k;
27     A(k,2)=k+1;
28     if mod(k,2)==0
29         A(k,3)=2;
30         A(k,4)=indut;
31     else
32         A(k,3)=1;
33         A(k,4)=res;
34     end
35     fprintf(fid, '%u %u %u %u \n', A(k,1), A(k,2), A(k,3), A(k,4));
36 end
37
38 % creo le variabili d'appoggio che mi serviranno per il secondo
    blocco
39
40 len=2*n+n+1;
41 nodi=2*n+2;
42 appo=1;
43
44 for k=2*n+1:len
45     if appo==1
46         A(k,1)=nodi;
47         A(k,2)=1;
48         A(k,3)=4;
49         A(k,4)= alim;
```

```

50     fprintf(fid, '%u %u %u %u \n', A(k,1), A(k,2), A(k,3), A(k,4))
    ;
51     else
52         A(k,1)=appo;
53         A(k,2)=nodi;
54         A(k,3)=3;
55         A(k,4)= capac;
56         fprintf(fid, '%u %u %u %u \n', A(k,1), A(k,2), A(k,3), A(k,4))
    ;
57     end
58
59 appo=appo+2;
60
61 end
62
63 fclose=(fid);

```

**Listing 3.2:** Script per creazione netlist linea elettrica a costanti concentrate.

Come dati di input vengono richiesti il numero di blocchi in cui si vuole suddividere la linea, i valori di impedenza resistiva, induttiva e capacitiva lungo la linea, e il valore della tensione di alimentazione.

A scopo esemplificativo, si prendano i seguenti dati:

- $R = 20\Omega$
- $L = 2 \cdot 10^{-8}$
- $C = 4 \cdot 10^{-11}$
- $V_a = 20000$
- numero di celle a  $T = 10$

Il codice, con questi dati, produrrà la seguente netlist:

starting node	ending node	type	value
1	2	$R_1$	20
2	3	$L_1$	$2 \cdot 10^{-3}$
3	4	$R_2$	20
4	5	$L_2$	$2 \cdot 10^{-3}$
5	6	$R_3$	20
6	7	$L_3$	$2 \cdot 10^{-3}$
7	8	$R_4$	20
8	9	$L_4$	$2 \cdot 10^{-3}$
10	1	V	20000
3	10	$C_1$	$4 \cdot 10^{-5}$
5	10	$C_2$	$4 \cdot 10^{-5}$
7	10	$C_3$	$4 \cdot 10^{-5}$
9	10	$C_4$	$4 \cdot 10^{-5}$

(3.9)

Si nota che la netlist acquisisce una lunghezza importante per una suddivisione in molte celle; in virtù di ciò si affronta nel prossimo step l'applicazione del MOR al codice di Analisi di Tableau in vari casi. Ciò permetterà di dimostrare l'utilità di avere un modello di ordine ridotto per lavorare con esempi di circuiti complessi e composti da molti elementi, sia in termini di memoria di archiviazione, sia in termini di tempistiche di calcolo.

### 3.2.3 Applicazione Tableau Analysis con MOR

I casi esposti di seguito riguarderanno rispettivamente la suddivisione della linea elettrica in:

1. 10 celle
2. 100 celle
3. 1000 celle

e se ne discutono i risultati. I dati delle impedenza e alimentazione precedentemente visti verranno utilizzati egualmente in ogni casistica sebbene non abbiano un effettivo riscontro reale, al fine di confrontare al meglio i risultati.

Si specifica che il codice MOR utilizzato è lo stesso dei capitoli precedenti, quindi viene fatta una valutazione nel range di  $\omega$  che va da 1 a 1000.

---

### caso con 10 celle

...

Le dimensioni, trovate utilizzando la funzione *size* in Matlab, delle matrici quadrate di riferimento del metodo standard ottenute sono le seguenti:

$$\dim M_1 = \dim M_2 = \dim M_3 = \dim M = 52 \times 52$$

mentre le dimensioni delle matrici quadrate di riferimento con il modello d'ordine ridotto sono:

$$\dim M_{1p} = \dim M_{2p} = \dim M_{3p} = \dim M_p = 5 \times 5$$

Si utilizza ora un semplice script per valutare le tempistiche necessarie per ottenere lo stesso vettore soluzione, utilizzando i due metodi. L'esempio viene eseguito con  $\omega = 50$ .

```

1 % tempo impiegato per trovare vettore soluzione
2 % utilizzando il modello standard
3
4 tic
5 M=M1+2*pi*50*M2+M3./(2*pi*50);
6 x=M\r;
7 toc
8
9 % tempo impiegato per trovare vettore soluzione
10 % utilizzando il modello ad ordine ridotto (MOR)
11

```

```

12 tic
13 Mp=M1p+2*pi*50*M2p+M3p./(2*pi*50);
14 xp=V*(Mp\rp);
15 toc

```

**Listing 3.3:** Script calcolo tempi con e senza modello ridotto.

Lo script restituisce in output:

$T_s$  di calcolo (senza MOR)  $\approx T_s$  di calcolo (con MOR)

---

### caso con 100 celle

...

Le dimensioni, trovate utilizzando la funzione *size* in Matlab, delle matrici quadrate di riferimento del metodo standard ottenute sono le seguenti:

$$\dim M_1 = \dim M_2 = \dim M_3 = \dim M = 502 \times 502$$

mentre le dimensioni delle matrici quadrate di riferimento con il modello d'ordine ridotto sono:

$$\dim M_{1p} = \dim M_{2p} = \dim M_{3p} = \dim Mp = 7 \times 7$$

Si utilizza ora lo script 3.3 per valutare le tempistiche necessarie per ottenere lo stesso vettore soluzione, utilizzando i due metodi. L'esempio viene eseguito con  $\omega = 50$ .

Lo script restituisce in output:

$T_s$  di calcolo (senza MOR) = 0.014718 secondi.

$T_s$  di calcolo (con MOR) = 0.000767 secondi.

---

### caso con 1000 celle

...

Le dimensioni, trovate utilizzando la funzione *size* in Matlab, delle matrici quadrate di riferimento del metodo standard ottenute sono le seguenti:

$$\dim M_1 = \dim M_2 = \dim M_3 = \dim M = 5002 \times 5002$$

mentre le dimensioni delle matrici quadrate di riferimento con il modello d'ordine ridotto sono:

$$\dim M_{1p} = \dim M_{2p} = \dim M_{3p} = \dim Mp = 8 \times 8$$

Si utilizza ora lo script 3.3 per valutare le tempistiche necessarie per ottenere lo stesso vettore soluzione, utilizzando i due metodi. L'esempio viene eseguito con  $\omega = 50$ .

Lo script restituisce in output:

$T_s$  di calcolo (senza MOR) = 4.070365 secondi.

$T_s$  di calcolo (con MOR) = 0.004188 secondi.

Si osserva che, soprattutto quando in cui le matrici nel caso standard, senza implementazione del MOR, sono di rango elevato, è evidente la convenienza di

utilizzo del modello di ordine ridotto.

Se questo dovessero essere utilizzato per uno studio e richiamato spesso per eseguire calcoli, confronti, integrazioni, la convenienza in termini di memoria di archiviazione dei risultati e soprattutto del tempo impiegato, è presto dimostrata.





# Capitolo 4

## Introduzione al metodo PEEC

In questo capitolo verrà inizialmente esposto un metodo di studio numerico per risolvere formulazioni integrali di simulazioni elettromagnetiche; si tratta del metodo PEEC, ossia *Partial Element Equivalent Circuit*.

In seguito a questo capitolo, ci si concentrerà sull'implementazione del metodo MOR sviluppato finora ad un codice di analisi elettromagnetica PEEC, già precedentemente sviluppato, ed infine ad un'analisi dei risultati ottenuti.

### 4.1 Metodi di risoluzione per problemi elettromagnetici

Negli ultimi decenni, la componentistica di apparecchi elettronici ed elettrici ha subito certamente un sostanziale cambiamento: le dimensioni si sono ridotte sempre più garantendo allo stesso tempo efficienza e prestazioni al passo con le esigenze.

Questi fattori hanno portato ad un aumento della complessità elettromagnetica, dovuta anche al maggiore effetto che l'elaborazione di segnali ad alte frequenze, come è per la quasi totalità dei casi nel settore elettronica al giorno d'oggi, ha sulle caratteristiche dei componenti. A pari passo, il tema della compatibilità elettromagnetica è divenuto ad essere focale.

Ciò ha portato all'elaborazione di metodi di calcolo per problemi elettromagnetici sempre più innovativi, discussi brevemente in questo capitolo, tra cui vi è il metodo PEEC, di cui si tratteranno gli aspetti principali in seguito.

### 4.1.1 FDM: metodo alle differenze finite

Il metodo alle differenze finite (FDM) [20], [3] si basa sull'analisi di problemi elettromagnetici entro un determinato dominio  $\Omega$ . Essa viene compiuta approssimando le equazioni in forma differenziale di Maxwell,

$$\begin{cases} \nabla \times E = -\frac{\partial B}{\partial t} \\ \nabla \times H = J + \frac{\partial D}{\partial t} \\ \nabla \cdot D = \rho \\ \nabla \cdot B = 0 \end{cases} \quad (4.1)$$

in differenze finite, discretizzando il problema.

Ciò ha sicuramente dei benefici, ma si tratta in ogni caso di discretizzare un problema di natura continua.

Esso consiste nel creare un reticolato (*mesh*) nel dominio  $\Omega$  considerato parallelo agli assi coordinati, in cui punto per punto la funzione presa in esame, ad esempio il potenziale elettrico  $U$ , viene sviluppato in serie di Taylor, in modo da poter consentire la formulazione alle differenze finite.

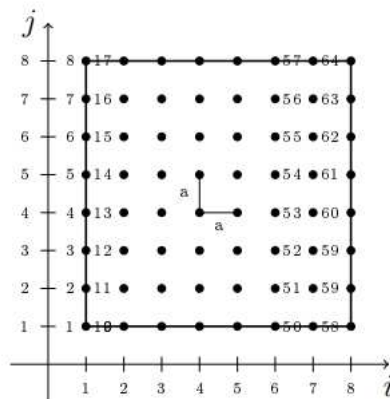


Figura 4.1: Mesh FDM

Resta inoltre necessario conoscere le condizioni al contorno che possono essere fornite sotto forme di 3 condizioni:

- $\frac{\partial U}{\partial h}|_{\Sigma} = g(P)$       Condizione di Neumann

- $U|_{\Sigma} = f(P)$                       Condizione di Dirichlet
- condizione mista Neumann e Dirichlet

### 4.1.2 FEM: metodi agli elementi finiti

Il metodo agli elementi finiti (FEM) [20], [3] si basa sullo stesso principio del metodo alle differenze finite, ossia sulla discretizzazione delle leggi di Maxwell.

A differenza del metodo precedentemente esposto però, in cui la *mesh* che veniva presa in considerazione stabiliva una geometria di analisi ben definita e non sempre fedele al caso reale, il reticolato ora è definito da nodi distribuiti in modo tale da infittirlo; tali nodi collegati tra loro creano delle forme che saranno suddivise in 3 categorie a seconda delle dimensionalità del problema:

- **caso 1D**: segmenti
- **caso 2D**: poligoni quali triangoli e quadrilateri prevalentemente
- **caso 3D**: tetraedri e altre forme tridimensionali. In questo caso la mesh sarà molto più difficile comporla rispetto al caso monodimensionale e bidimensionale

Le funzione che si stanno studiando, come ad esempio il potenziale  $U$ , vengono poi interpolate in questi elementi attraverso l'utilizzo di funzioni di base, solitamente polinomiali, per "pesare" le variabili in gioco negli elementi della *mesh*.

Ciò significa che il grado di accuratezza sarà tanto maggiore quanto maggiore sarà il grado delle funzioni polinomiali interpolanti, con, dall'altra faccia della medaglia, un maggiore dispendio computazionale.

Si presentano in seguito due esempi grafici di quella che può essere una mesh generata per lo studio di due elementi, uno bidimensionale e uno tridimensionale [16].

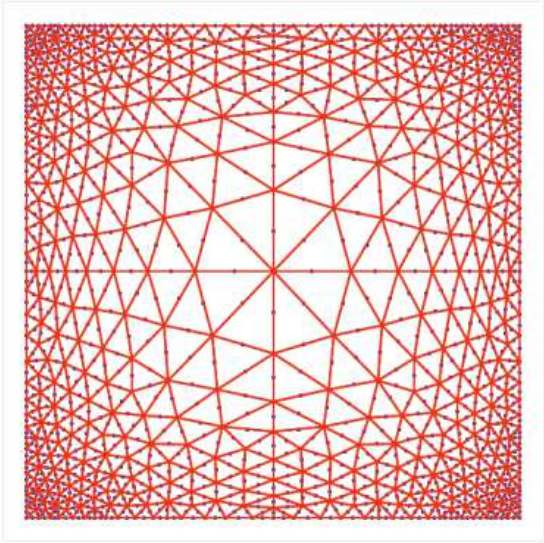


Figura 4.2: Mesh 2D

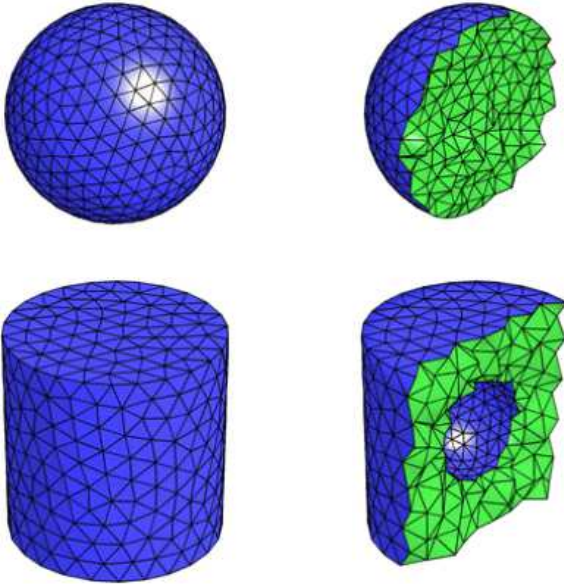


Figura 4.3: Mesh 3D

### 4.1.3 IEM: metodi alle equazioni integrali

A differenza dei metodi precedentemente visti (metodi differenziali), un altro tipo di formulazioni per la risoluzione di problemi di elettromagnetismo sono dei metodi che sfruttano la risoluzione di equazioni integrali.

In questo caso la funzione incognita, il campo elettrico o il potenziale che sia, è approssimata da una serie di funzioni che sono soluzione esatta del problema all'interno del dominio  $\Omega$  di studio, hanno quindi un supporto globale [20]. Un esempio di equazione generale a cui si riferisce la teoria dei metodi IEM, [20] è la seguente:

$$g(x) = \beta(x) \cdot u(x) + \lambda \int_a^b K(x, x') \cdot u(x') \cdot dx' \quad (4.2)$$

dove  $\beta, g, K, \lambda$  sono conosciute ed  $u$  è la funzione incognita.

Le formulazioni IEM, a differenza del FEM (o FDM) che hanno soluzione al contorno ben definite, hanno una soluzione approssimata al contorno del dominio di studio  $\Omega$ . Viceversa le soluzioni all'interno del dominio per il FEM e per l'FDM sono approssimate da funzioni polinomiali.

Per quanto utile sia il FEM se implementato ad analisi elettromagnetiche (o termodinamiche), piccoli elementi del dominio che vanno discretizzati [21], e che quindi portano ad un addensamento della mesh in alcuni punti, rischiano di produrre elevatissimi costi computazionali.

Come accennato precedentemente, le IEM sfruttano le equazioni di Maxwell in forma differenziale:

$$\left\{ \begin{array}{l} \oint_{\partial S} E \cdot dl = - \int_S \frac{\partial B}{\partial t} \cdot dS \\ \oint_{\partial S} H \cdot dl = \int_S (J + \frac{\partial D}{\partial t}) \cdot dS \\ \oint_{\partial V} D \cdot dS = \oint_V \rho \cdot dV \\ \oint_{\partial V} B \cdot dS = 0 \end{array} \right. \quad (4.3)$$

Le equazioni integrali si possono suddividere quindi in due grandi famiglie: le VIE, ossia le equazioni integrali di volume, e le SIE, ossia le equazioni integrali di superficie.

Tra i metodi risolutivi che applicano le VIE vi si trova per l'appunto il metodo PEEC (*Partial Element Equivalent Circuit*), del quale se ne discutono gli aspetti

costitutivi nel capitolo successivo.

## 4.2 PEEC

Il metodo PEEC [17], o *Partial Element Equivalent Circuit*, è stato introdotto per la prima volta da A.Ruehli negli anni '70, per la risoluzione di problemi di natura elettromagnetica. Con la crescente evoluzione dei circuiti elettrici ma soprattutto elettronici, a pari passo con la crescente complessità circuitale e di compatibilità elettromagnetica, si è visto crescere l'interesse verso la metodologia PEEC.

Il vantaggio principale di tale metodo [20], [21] è la possibilità di modellare il problema elettromagnetico come un circuito equivalente a parametri concentrati, con la possibilità di collegare circuiti esterni ad esso e studiarne le interazioni che ne scaturirebbero in termini di variabili di stato quali correnti, tensioni e potenziali.

Gli elementi della mesh [14] sono assimilati a dei circuiti RLC passivi che possono quindi essere in seguito trattati come tali nel sistema di equazioni che ne consente la risoluzione del problema all'origine.

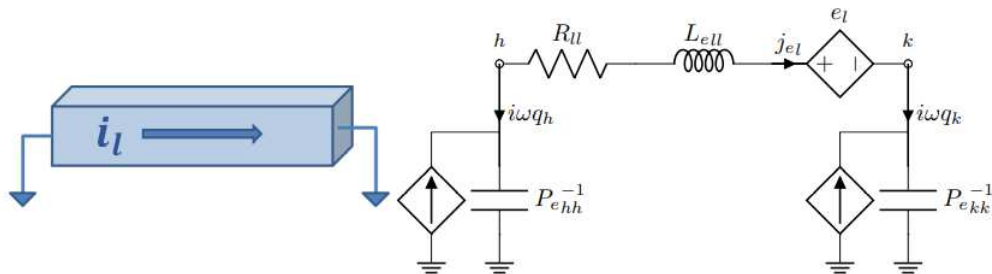


Figura 4.4: Passaggio da elemento discretizzato a circuito equivalente [21]

Il PEEC è una particolare forma delle VIE; in riferimento a ciò va specificato quindi che gli elementi che vengono discretizzati nello studio sono solo quelli facenti parte del dominio attivo del problema, ossia mezzi conduttivi, dielettrici e conduttivi. Le interazioni elettromagnetiche vengono modellate come accoppiamenti induttivi e capacitivi.

Il punto di partenza da cui comprendere la metodologia utilizzata nel metodo PEEC è lo studio del vettore del campo elettrico. Questo può essere scritto in funzione del potenziale vettoriale magnetico e del potenziale scalare elettrico:

$$\vec{E}(\vec{r}, \omega) = -j\omega\vec{A}(\vec{r}, \omega) - \nabla\Phi(\vec{r}, \omega) \quad (4.4)$$

Il potenziale vettoriale magnetico può essere considerato come:

$$\vec{A}(\vec{r}, \omega) = \mu \int_{v'} (\vec{G}(\vec{r}, \vec{r}') \vec{J}(\vec{r}', \omega) dv' \quad (4.5)$$

in cui  $\vec{J}$  è il vettore densità di corrente elettrica considerato in un certo punto  $r'$ , e  $\vec{G}$  è la funzione di Green libera:

$$\vec{G}(\vec{r}, \vec{r}') = \frac{e^{-j\beta R}}{4\pi R} \quad (4.6)$$

dove  $R = |r - r'|$ . Il vettore scalare elettrico invece:

$$\Phi(\vec{r}, \omega) = \frac{1}{\epsilon} \int_{v'} (\vec{G}(\vec{r}, \vec{r}') \vec{q}(\vec{r}', \omega) dv' \quad (4.7)$$

dove  $v'$  è considerato come il volume dell'elemento conduttore considerato e  $q$  è la densità di carica ivi presente.

Riarrangiando l'equazione 4.4, sostituendo i termini appena trovati si ottiene:

$$\vec{E}(\vec{r}, \omega) = -j\omega\mu \int_{v'} (\vec{G}(\vec{r}, \vec{r}') \vec{J}(\vec{r}', \omega) dv' - \frac{\nabla}{\epsilon} \int_{v'} (\vec{G}(\vec{r}, \vec{r}') \vec{q}(\vec{r}', \omega) dv' \quad (4.8)$$

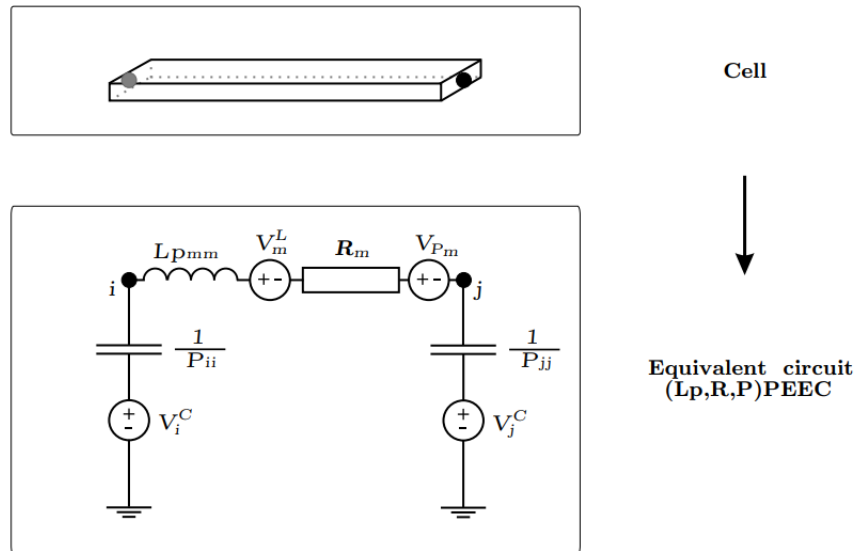
I termini dell'equazione possono essere interpretati come [19] mostrato di seguito:

- il termine a sinistra può essere considerato come la differenza di potenziale che caratterizza l'elemento del dominio considerato
- il primo termine a destra dell'uguale può essere considerato come una differenza di potenziale dovuta al carattere induttivo del conduttore
- il secondo termine a destra dell'uguale può essere considerato come una differenza di potenziale dovuta al carattere capacitivo del conduttore

Graficamente, quanto detto, si può raffigurare come segue, mostrando il passaggio da elemento volumetrico del dominio di analisi al circuito equivalente che lo



rappresenta:



**Figura 4.5:** Dall'elemento volumetrico al circuito equivalente

Essendo questo lavoro di tesi non incentrato sull'esposizione completa del metodo PEEC, ma sull'implementazione del MOR ad un suo script di analisi in Matlab, si rimanda alla letteratura del metodo l'analisi matematico-fisica che lo costituisce.

Per il caso che si è preso in considerazione, una versione 1D del PEEC [18] è utilizzato per lo studio in atto. In particolare, solo il carattere conduttivo del dominio è preso in considerazione; a questo è possibile abbinare altri elementi circuitali a parametri concentrati esterni in caso fosse richiesto.

Similmente a quanto visto per la Tableau Analysis, l'equazione che rappresenta il circuito completo è la seguente:

$$\begin{bmatrix} R + i\omega L & A^T & 0 & 0 \\ P \cdot A & -i\omega \mathbb{1} & P \cdot A_e & 0 \\ 0 & A_e^T & R_{app} + i\omega L_{app} & A_n^T \\ 0 & 0 & A_n & 0 \end{bmatrix} \cdot \begin{bmatrix} j \\ \phi \\ j_{app} \\ \phi_{app} \end{bmatrix} = \begin{bmatrix} E \\ P \cdot I \\ E_{app} \\ I_{app} \end{bmatrix} \quad (4.9)$$

dove  $A_n$  rappresenta la matrice d'incidenza di connessione tra i lati e i nodi del dominio del problema elettromagnetico,  $A_e$  è la matrice che collega i nodi del circuito principale a quelli del circuito esterno,  $P$  viene ricavata dalla matrice inversa delle capacità,  $R_{app} + i\omega L_{app}$  rappresenta la matrice d'impedenza concentrata,  $I_{app}$  rappresenta il vettore delle sorgenti di corrente di origine esterna che si iniettano nei nodi principali ed infine  $E_{app}$  rappresenta il vettore delle tensioni ai lati del circuito principale.

Il sistema di equazioni rappresentato da 4.9 è quello che dovrà essere ridotto d'ordine tramite implementazione del MOR allo script del programma Matlab *MAIN\_PEEC\_1D.m* a cui si rimanda [18].

## Capitolo 5

# Applicazione del MOR al metodo PEEC

Si affronta ora il procedimento atto ad implementare il MOR al programma basato sulla risoluzione tramite metodo PEEC di un caso di studio monodimensionale, ed analisi dei risultati ottenuti.

### 5.1 Utilità del modello di riduzione d'ordine nel metodo PEEC

Il metodo PEEC è stato considerato dagli ingegneri negli ultimi decenni profondamente versatile, per la sua capacità di trasformare problemi elettromagnetici in circuiti RLC passivi [15].

Quando si tratta di progetti di ottimizzazione EM od esplorazione dello spazio di un progetto, numerose simulazioni vengono fatte per studiarne caratteristiche ed efficacia; utilizzando il metodo PEEC spesso vengono coinvolte moli di elementi circuitali tali per cui lo studio reiterato con il modello "standard", in cui le matrici di calcolo sono di conseguenza di elevato rango, diviene molto complesso e richiede uno sforzo computazionale non di poco conto, se non a volte inattuabile.

A questo proposito metodi di riduzione d'ordine sono stati proposti [19], ed in

particolare le tecniche pMOR aventi come variabile quindi la frequenza, sono state implementate, consentendo da una parte di ridurre grandi sistemi di equazioni rappresentanti i casi di studio, dall'altra di mantenere la stabilità e la robustezza di calcolo all'interno dello spazio di progetto.

Il metodo proposto finora in questo lavoro di tesi si propone di rientrare in questa ultima categoria. Viene di seguito presentata una variante dello script 2.3 applicabile allo script PEEC-1D [18] e successivamente eseguita in ambiente Matlab.

## 5.2 Modifica script del MOR per programma PEEC 1D

Facendo riferimento alla matrice 4.9 e congiuntamente allo script del codice MOR creato in precedenza (script 2.3), si espone di seguito come viene implementato al programma che esegue un metodo PEEC monodimensionale [18]. La matrice rappresentante il caso di studio viene suddivisa in modo simile a quanto fatto nei capitoli precedenti con la Tableau Analysis, in modo da poter isolare la pulsazione  $\omega$  e renderla un parametro isolato su cui il codice possa operare.

Di seguito viene riportata la divisione:

$$\begin{bmatrix} R + i\omega L & A^T & 0 & 0 \\ P \cdot A & -i\omega \mathbf{1} & P \cdot A_e & 0 \\ 0 & A_e^T & R_{app} + i\omega L_{app} & A_n^T \\ 0 & 0 & A_n & 0 \end{bmatrix} = \quad (5.1)$$

$$= \begin{bmatrix} R & A^T & 0 & 0 \\ P \cdot A & 0 & P \cdot A_e & 0 \\ 0 & A_e^T & R_{app} & A_n^T \\ 0 & 0 & A_n & 0 \end{bmatrix} + \omega \cdot \begin{bmatrix} iL & 0 & 0 & 0 \\ 0 & -i\mathbf{1} & 0 & 0 \\ 0 & 0 & iL_{app} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Di conseguenza l'equazione 4.9 riscritta diviene:

$$[M_1 + \omega M_2] \cdot x = b \quad (5.2)$$

dove:

$$M_1 = \begin{bmatrix} R & A^T & 0 & 0 \\ P \cdot A & 0 & P \cdot A_e & 0 \\ 0 & A_e^T & R_{app} & A_n^T \\ 0 & 0 & A_n & 0 \end{bmatrix} \quad (5.3)$$

$$M_2 = \begin{bmatrix} iL & 0 & 0 & 0 \\ 0 & -i\mathbb{1} & 0 & 0 \\ 0 & 0 & iL_{app} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.4)$$

$$x = \begin{bmatrix} j \\ \phi \\ j_{app} \\ \phi_{app} \end{bmatrix} \quad (5.5)$$

$$b = \begin{bmatrix} E \\ P \cdot I \\ E_{app} \\ I_{app} \end{bmatrix} \quad (5.6)$$

Questa nuova equazione si inserisce nello script del MOR, che ora ha come input le due nuove matrici  $M_1$  ed  $M_2$  ottenute dalla matrice  $M$  originale, la quale viene fornita dal programma esterno che implementa il metodo PEEC monodimensionale:

```

1 function [M1p,M2p,Mp,bp,V,c,omegav] = pMORPEEC(M1,M2,b,omega)
2
3 wmax=omega+1e6;           % valore massimo di pulsazione per
                           % la validita' del MOR
4 wmin=omega-1e6;         % valore minimo di pulsazione per
                           % la validita' del MOR
5 Nrand=10000;           % numero di valori di frequenza
                           % testati per ogni iterazione
6 V=[];                   % inizializzazione vettore V
7 c=1;                   % counter
8 puls=1e9;              % pulsazione di incipt per le
                           % iterazioni
9 re=inf;                 % residuale del problema
                           % approssimato (inizializzato a
                           % +inf)
10 eta=1e-3;              % errore accettabile dal problema
11
12
13 % inizio ciclo while che da soluzioni finch la soluzione
   % approssimata del MOR ha
14 % un errore apprezzabile
15
16 while re>eta
17
18     % calcolo in primis il risultato x con le matrici "normali"
   % del problema
19
20     M=M1+puls*M2;
21     x=M\b;
22
23     % ortonormalizzo il risultato x in nuove basi ortonormali
24     V=mgson([V,x]);
25
26     % cambiamento di basi di M1,M2 e b
27
28     M1p=V'*M1*V;
29     M2p=V'*M2*V;
30     bp=V'*b;
31
32     % genero Nrand valori casuali di omega e ne creo un vettore
   % in questo caso da 0 a 10000
33     omegav=wmin+(wmax-wmin)*rand(Nrand,1);
34
35     % inizializzo vettore dei residuali che andr a calcolare

```

```

36 % e xp vettore della soluzione con basi ortonormali
37 rh=zeros(Nrand,1);
38
39 for h=1:Nrand
40
41 % h-esimo valore di omega generato
42 puls=omegav(h);
43
44 % soluzione in pMOR per ogni h
45
46 Mp=M1p+puls*M2p;
47 xp=Mp\bp;
48 M=M1+puls*M2;
49
50 % soluzione approssimata di dimensioni (2nb+n-1,1)
51 xh=V*xp;
52
53 % vettore dei residuali per ogni soluzione approssimata
54 rh(h)=(norm(b))\((norm(M*xh-b));
55
56 end
57
58 % residuale maggiore tra quelli ottenuti per la success
59 iterazione
60 [rM, posh]=max(rh);
61 re=rM
62
63 % omega del residuale maggiore
64 puls=omegav(posh);
65
66 % verifico se il residuale preso ancora maggiore dell'
67 incertezza massima desiderata
68 % e se lo , aumento di 1 il counter c per la success
69 iterazione
70
71 if re>eta
72     c=c+1
73 end
74 end

```

Listing 5.1: Script MOR per metodo PEEC

### 5.3 Simulazione PEEC 1D con metodo MOR: risultati ed osservazioni

Si inserisce quindi lo script del MOR nel programma utilizzando il metodo PEEC-1D al fine di creare il modello ridotto del problema.

C'è da precisare che la simulazione viene eseguita considerando il dominio di studio solamente conduttivo e per frequenze di funzionamento tali per cui i potenziali non debbano considerarsi ritardati.

In riferimento allo script del MOR 2.3 riportato sopra, che è stato adattato alle variabili del programma che utilizza il metodo PEEC, si fanno le seguenti considerazioni:

- l'intervallo di frequenza entro il quale viene implementato il MOR varia tra un minimo ed un massimo, ed il modello generato avrà validità in quell'intervallo. Si avrà quindi:

$$\omega - \omega_0 < \omega_t < \omega + \omega_0 \quad (5.7)$$

dove  $\omega_t$  è la pulsazione casuale presa ad ogni iterazione  $N_{rand}$  volte, e  $\omega$  s'intende essere la pulsazione per la quale il programma PEEC-1D calcola la soluzione del caso di studio.

- $N_{rand}$  viene preso con valore 10000, in questo modo la validità sarà più fine.
- la tolleranza viene modificata in base alla complessità del caso di studio.

Si esegue il programma con implementazione del MOR adattato a due casi di studio. Il primo sarà un segmento di linea in **cavo bifilare** lunga  $1m$ , in cui ogni cavo è alimentato da un lato da un generatore di corrente e dall'altro lasciato aperto. Nel secondo studio viene analizzata un' **antenna NFC** eccitata ai due nodi estremi da un generatore di corrente. Si esegue il programma con

$$f = 3 \cdot 10^8 \text{ Hz}$$



$$\omega = 2 \cdot \pi \cdot f = 1.885 \cdot 10^9 \frac{rad}{s}$$

inoltre il raggio dei cavi in entrambi i casi sarà  $r = 0.1 \cdot 10^{-3} m$  con un resistività  $\rho = 1.754 \cdot 10^{-8} \frac{\Omega}{m}$ . La corrente circolante nei conduttori è stata ipotizzata essere di 1 A.

Per ogni caso di studio vengono analizzati i risultati, ossia considerato il modello ridotto ottenuto e il confronto con il modello base per verificarne l'efficacia effettiva.

---

### 5.3.1 Cavo bifilare

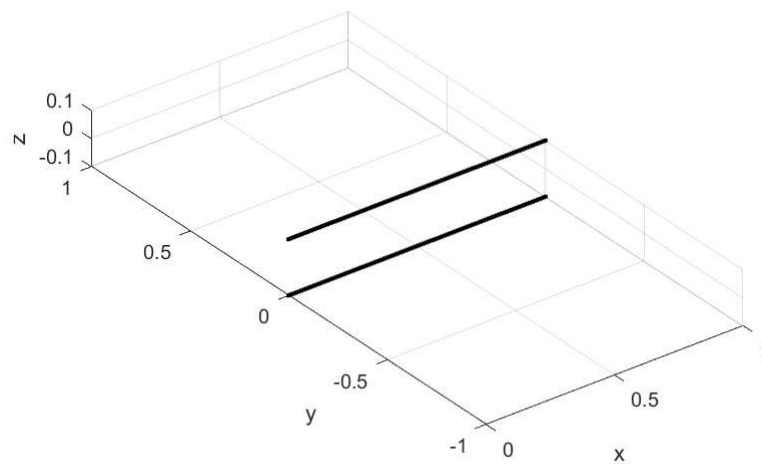


Figura 5.1: Cavo bifilare.

Per il caso di studio del cavo bifilare vengono utilizzati i seguenti dati per il MOR:

- $\omega_0 = 10^6$
- $\eta = 10^{-3}$

Il modello ridotto restituisce i seguenti risultati:

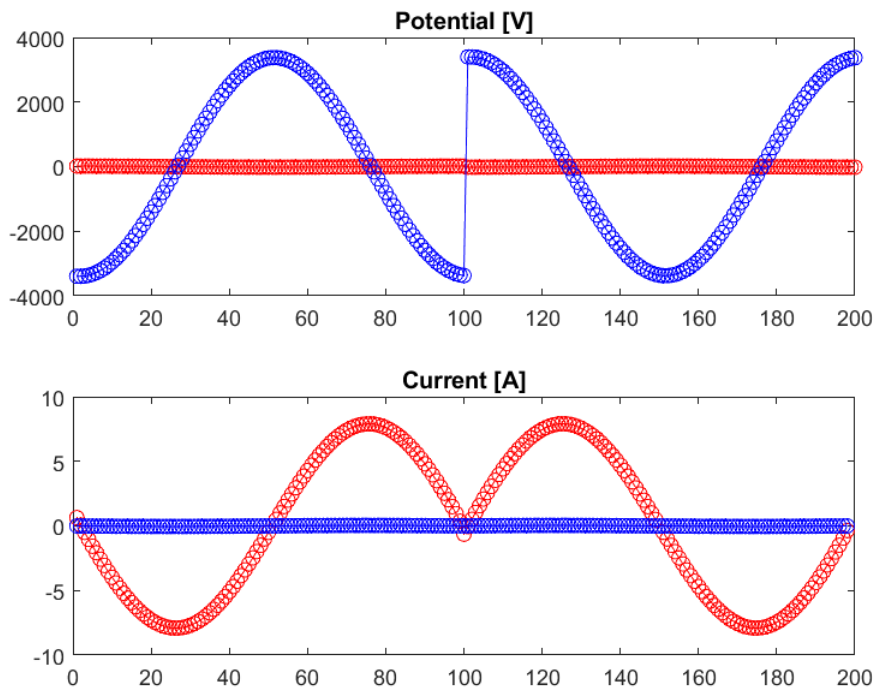
$$\dim M_1 = \dim M_2 = \dim M = 398 \times 398 \quad (5.8)$$

mentre per il modello ad ordine ridotto ottenuto:

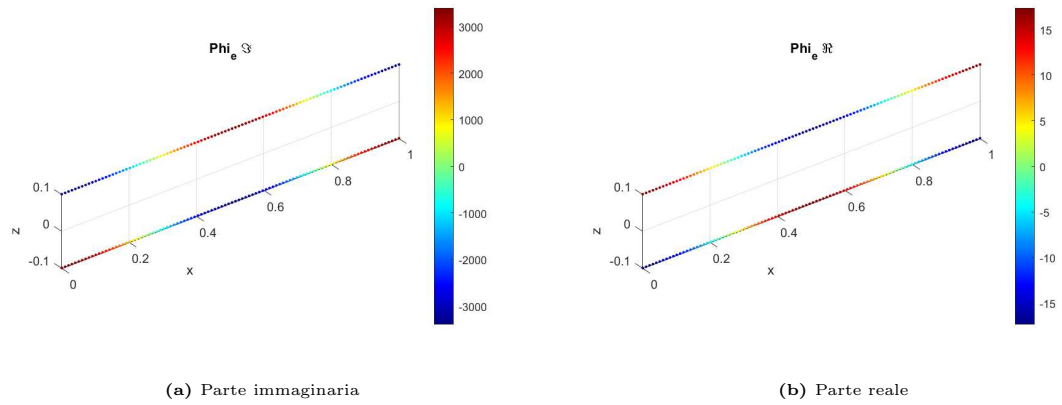
$$\dim M_{1p} = \dim M_{2p} = \dim M_p = 3 \times 3 \quad (5.9)$$

Si confrontano ora graficamente alcuni risultati del vettore  $x$  soluzione nel caso del modello senza implementazione del codice di riduzione d'ordine e nel caso di utilizzo del modello di ordine ridotto:

### modello senza MOR

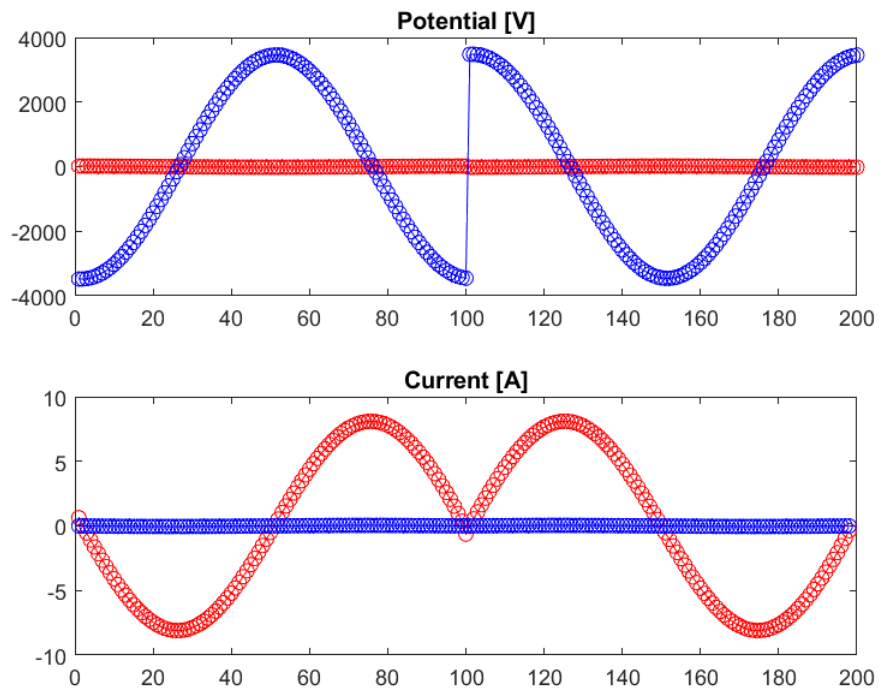


**Figura 5.2:** tensione e corrente lungo i due cavi (in blu la parte immaginaria e in rosso la parte reale).

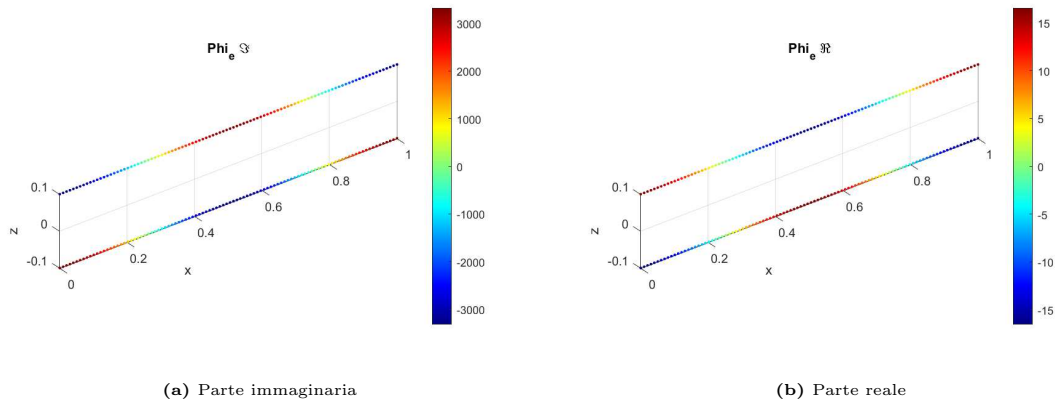


**Figura 5.3:** Potenziale scalare elettrico  $\phi_e$

### modello con MOR

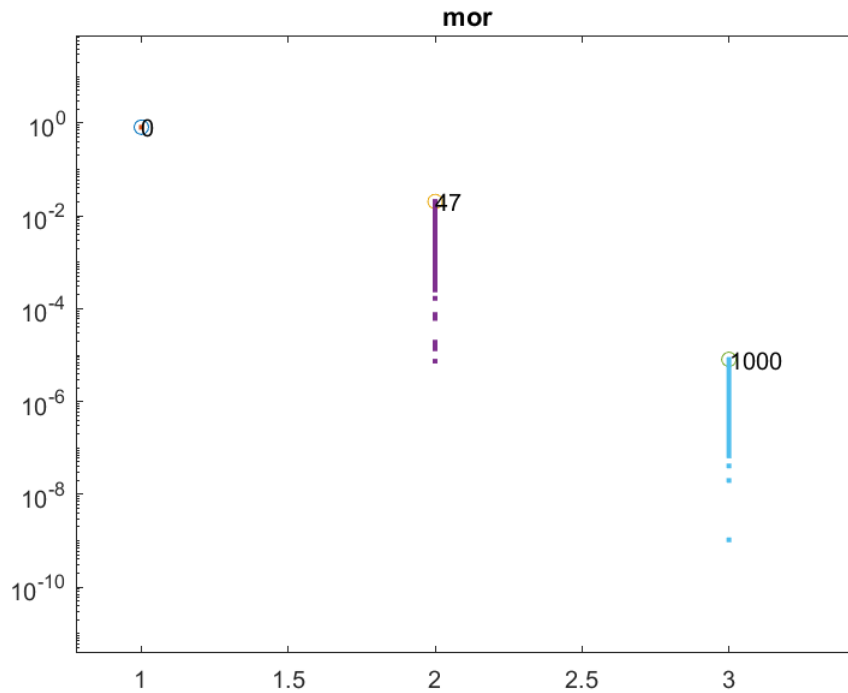


**Figura 5.4:** tensione e corrente lungo i due cavi (in blu la parte immaginaria e in rosso la parte reale).



**Figura 5.5:** Potenziale scalare elettrico  $\phi_e$

Si plotta ora la convergenza del metodo in un semplice grafico, avente come ordinata il valore del residuale massimo ad ogni iterazione, e ad ascissa il valore dell'iterazione  $c$ .



**Figura 5.6:** Convergenza del metodo (3 step)

### 5.3.2 Antenna NFC

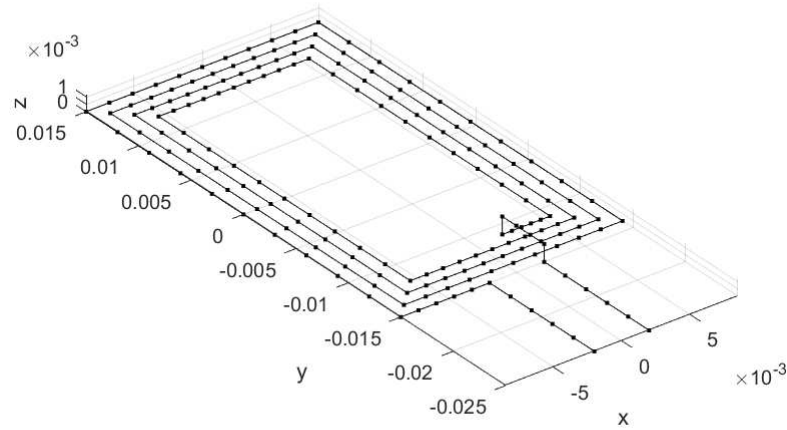


Figura 5.7: Antenna NFC.

Per il caso di studio dell'antenna NFC si prende un range di frequenza più ampio e si richiede una tolleranza maggiore dal problema, con conseguente maggiore precisione del modello ridotto:

- $\omega_0 = 10^8$
- $\eta = 10^{-8}$

Per quanto concerne le matrici senza riduzione d'ordine si ha:

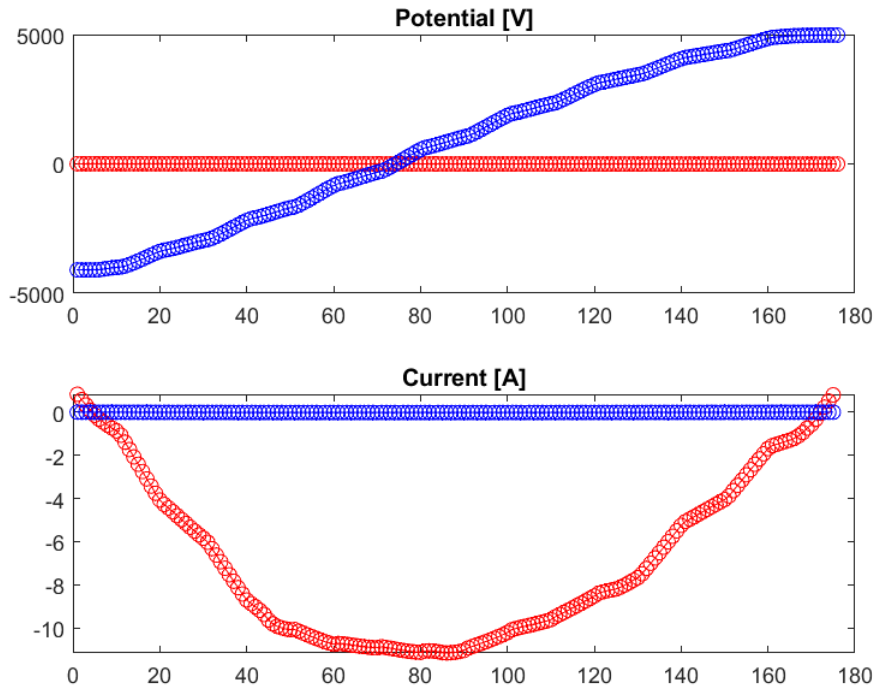
$$\dim M_1 = \dim M_2 = \dim M = 2101 \times 2101 \quad (5.10)$$

mentre per il modello ad ordine ridotto ottenuto:

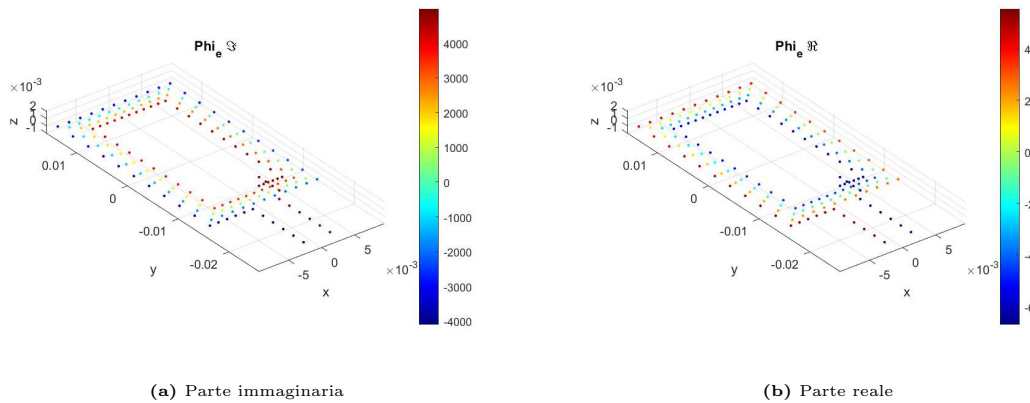
$$\dim M_{1p} = \dim M_{2p} = \dim M_p = 5 \times 5 \quad (5.11)$$

Si confrontano ora graficamente alcuni risultati del vettore  $x$  soluzione nel caso del modello senza implementazione del codice di riduzione d'ordine e nel caso di utilizzo del modello di ordine ridotto:

### modello senza MOR

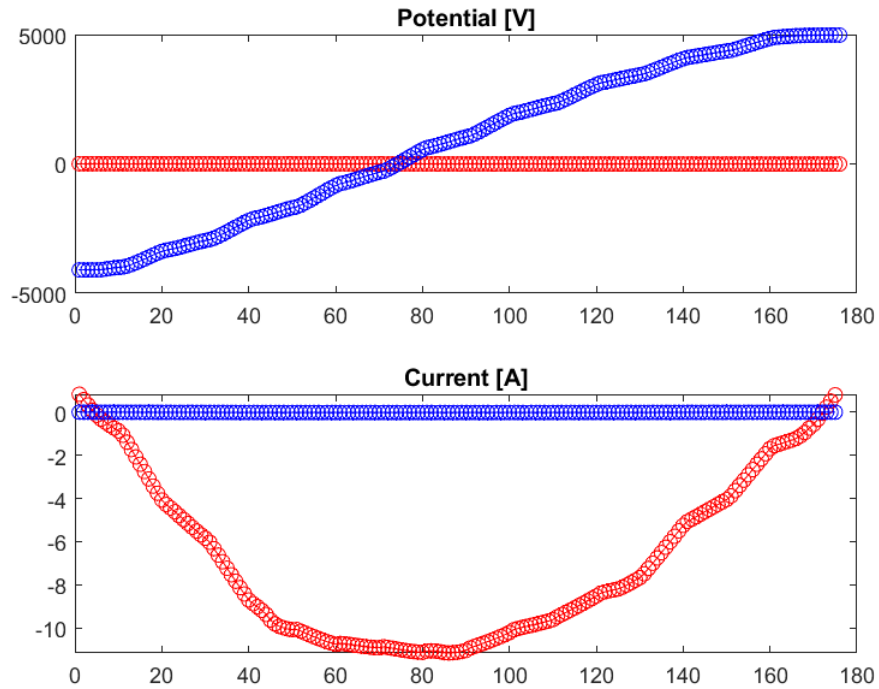


**Figura 5.8:** tensione e corrente lungo il percorso dell'antenna NFC (in blu la parte immaginaria e in rosso la parte reale).

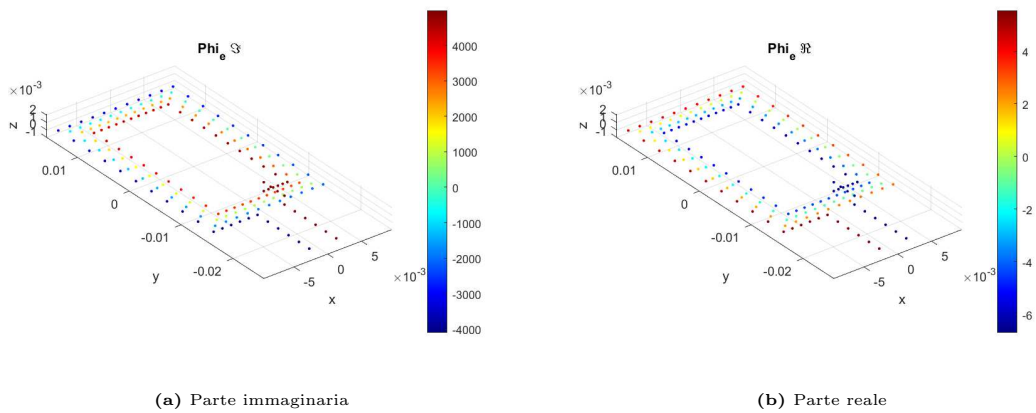


**Figura 5.9:** Potenziale scalare elettrico  $\phi_e$

### modello con MOR

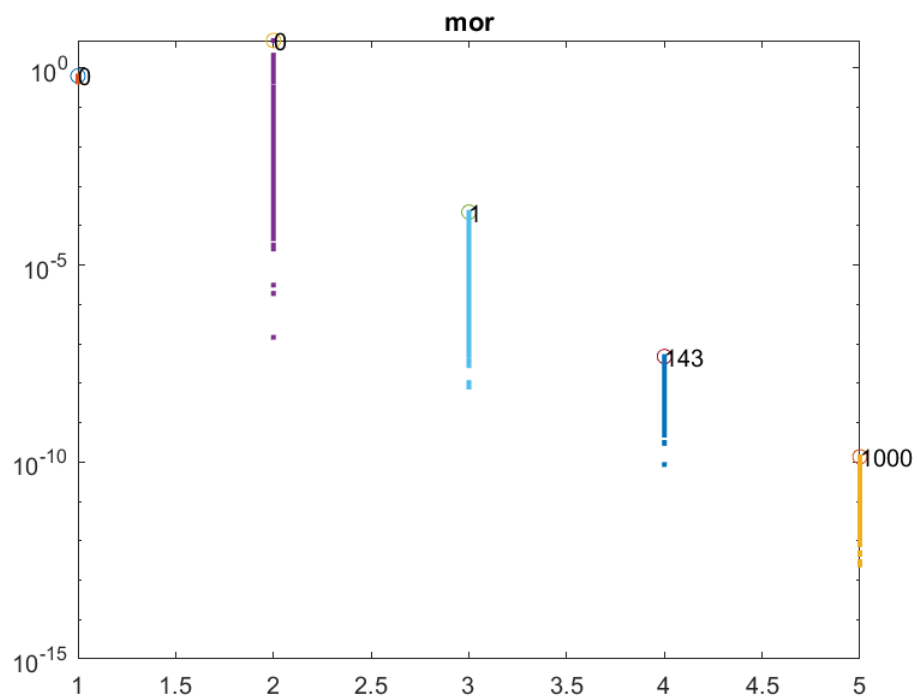


**Figura 5.10:** tensione e corrente lungo il percorso dell'antenna NFC (in blu la parte immaginaria e in rosso la parte reale).



**Figura 5.11:** Potenziale scalare elettrico  $\phi_e$

Si plotta ora la convergenza del metodo in un semplice grafico, avente come ordinata il valore del residuo massimo ad ogni iterazione, e ad ascissa il valore dell'iterazione  $c$ .



**Figura 5.12:** Convergenza del metodo (5 step)



Per entrambi i casi di studio la soluzione ricavata utilizzando il metodo MOR corrisponde alla soluzione standard, ottenuta utilizzando la matrice  $M$  e il vettore  $b$ :

$$x_{MOR} \approx x_{standard} \quad (5.12)$$

ciò è garantito, in rigoroso riferimento ai dati di input che sono stati dati, a meno di una tolleranza di  $\eta = 10^{-3}$  per il cavo bifilare e di  $\eta = 10^{-8}$  per l'antenna NFC (ma l'errore può rivelarsi anche inferiore!).

Dai grafici arriva un'ulteriore conferma di quanto detto, essendo i plot ,ottenuti con e senza utilizzo del MOR, evidentemente uguali.

Se si richiedesse una tolleranza più stringente, un numero di pulsazioni generate  $N_{rand}$  diverso e/o un range di frequenze più ampio, si avrebbero più iterazione nel MOR per arrivare alla convergenza, come notato precedentemente tra i due casi in cui il secondo richiedeva una precisione; di conseguenza il modello di ordine ridotto successivamente ottenuto avrebbe dimensione più elevate, ma allo stesso tempo sarebbe molto più preciso.



# Conclusioni

Alla luce di quanto si è visto in questo lavoro di tesi, il metodo di riduzione d'ordine ha dimostrato la sua efficacia. Partendo dall'analisi di Tableau si è costruito un codice di riduzione d'ordine (MOR) potenzialmente applicabile, con alcune modifiche, a molti programmi di simulazione elettromagnetica: in particolare, si è applicato ad un programma che implementa una particolare forma delle VIE (*Volume Integral Equation*), e se ne sono analizzati i risultati. Da subito si è riscontrato come l'ordine del modello di studio nei casi analizzati sia stato ridotto a dimensioni molto inferiori, consentendo un più agevole ed efficiente utilizzo qualora quei modelli fossero stati utilizzati per altre simulazioni e analisi.

Metodi di riduzione dell'ordine sono sempre più ricercati dagli ingegneri per la loro utilità, versatilità ed efficienza in termini di risparmio di memoria e velocità di computazione. Con il progresso scientifico caratterizzato da un'evoluzione sempre più esponenziale, i dispositivi elettrici e magnetici divengono sempre più complessi e disporre di metodologie che vadano a semplificare quella che è l'analisi dei modelli diviene necessario.

Questo lavoro di tesi può essere uno spunto per evolvere il codice MOR fino ad arrivare ad un'analisi di modelli dipendenti da più variabili; in alternativa può essere studiato per essere applicato a diversi casi di studio che non siano prettamente elettromagnetici, come ad esempio analisi nel campo della termodinamica o del settore gestionale.



# Bibliografia

- [1] Robert R. Schaller  
"MOORE'S LAW: past, present, and future"  
George Mason University  
06.1997
  
- [2] Christopher Griffin con la contribuzione di Elena Kosygina, Suraj Shekhar  
"Graph Theory: Penn State Math 485 Lecture Notes"  
2011-2021
  
- [3] Fiorindo Matteo, Sempredoni Nicola, Tollardo Michele Tassinato Anna,  
Amatruda Mattia Giovanni Beggiato, Marco Francesco Grolla, Nicolò Paolo  
"Computational Electrical Engineering XXX"  
22.03.2019
  
- [4] M. Vansáč, I. Tomčíková  
"Sparse Tableau Analysis used in Matlab for simulation of electric circuits"  
Technical University in Košice, Park Komenského 3, Košice, 04200, Slovakia  
27.03.2013
  
- [5] Jacek Klucznik  
"Earth wires currents calculation by tableau analysis"  
Gdansk University of Technology, Faculty of Electrical and Control Engineering,  
Narutowicza 11/12, 80-233, Poland  
15.06.2017
  
- [6] Riccardo Torchio, Lorenzo codecasa (Member, IEEE), Luca di Rienzo (Senior Member, IEEE) e Federico Moro (Member, IEEE)  
"Fast Uncertainty Quantification in Low Frequency Electromagnetic Problems by an Integral Equation Method Based on Hierarchical Matrix Compression";  
Università di Padova, pubblicato: 11.10.2019

- [7] Chengdong Yuan, Dennis Hohlfeld, Tamara Bechtold  
*"Design optimization of a miniaturized thermoelectric generator via parametric model order reduction"*  
Department of Engineering, Jade University of Applied Sciences, Wilhelmshaven, Germany  
Institute for Electronic Appliances and Circuits, University of Rostock, Rostock, Germany  
8.03.2021
- [8] Lihong Feng , Peter Benner  
*"A robust algorithm for parametric model order reduction"*  
Mathematics in Industry and Technology, Faculty of Mathematics Chemnitz University of Technology, D-09107 Chemnitz, Germany  
2007
- [9] Matthias Geuss, Heiko Panzer and Boris Lohmann  
*"On Parametric Model Order Reduction by Matrix Interpolation"*  
2013 European Control Conference (ECC), Zürich, Switzerland  
17-19.07.2013
- [10] Alfred Tze-Mun Leung, Roni Khazaka  
*"Parametric Model Order Reduction Technique For Design Optimization"*  
Department of Electrical and Computer Engineering McGill University Montreal, Quebec, Canada  
Department of Electrical and Computer Engineering McGill University Montreal, Quebec, Canada  
2005
- [11] Ajay Venkateswaran  
*"Analysis of Planar EBG Structures Using Transmission Line Models"*  
Master of Engineering - Department of Electrical Computer Engineering University, Montreal, Canada  
01.2009
- [12] Dario A. Bini  
*"Aspetti computazionali dei metodi di Gauss e di Householder"*  
Università di Pisa  
1.11.2011

- [13] Andrea Chinello  
*"Studio del comportamento a regime di linee elettriche di trasmissione dissimetriche tradizionali e innovative"*  
Università degli studi di Padova  
2015-2016
- [14] Francesco Ferranti, Member, IEEE, Giulio Antonini, Senior Member, IEEE, Tom Dhaene, Senior Member, IEEE and Luc Knockaert, Senior Member IEEE  
*"Guaranteed Passive Parameterized Model Order Reduction of the Partial Element Equivalent Circuit (PEEC) Method"*  
IEEE Transactions on Electromagnetic compatibility  
Vol. 52, NO. 4  
11.2010
- [15] Francesco Ferranti, Giulio Antonini  
*"Riduzione d'ordine parametrica per sistemi elettromagnetici analizzati tramite il metodo PEEC"*  
Dip. of Information Technology (INTEC), Ghent University-IBBT, Belgium, UAq EMC Laboratory  
Dip. di Ingegneria Elettrica e dell'Informazione, Università degli Studi dell'Aquila
- [16] John Burkardt  
*"Meshing for the Finite Element Method"*  
Department of Scientific Computing  
Florida State University  
04.11.2011
- [17] Albert Ruehli  
*"Partial Element Equivalent Circuit (PEEC) Method and its Application in the Frequency and Time Domain"*  
IBM Research Division, T. J. Watson Research Center Yorktown Heights, NY 10598  
1996
- [18] <https://github.com/UniPD-DII-ETCOMP/PEEC-1Dù>
- [19] Jonas Ekman  
*"Electromagnetic Modeling Using the Partial Element Equivalent Circuit Me-*

*thod*”

EISLAB, Dept. of Computer Science and Electrical Engineering, Lulea University of Technology, Lulea, Sweden

03.2003

[20] Tafano Luigi

*”Il metodo “Partial Element Equivalent Circuit” per la modellizzazione elettromagnetica di interconnessioni e packaging elettronico”*

Università degli studi di Napoli ”Federico II”

2003

[21] Riccardo Torchio

*”Extending the Unstructured PEEC Method to Magnetic, Transient, and Stochastic Electromagnetic Problems”*

, Università degli Studi di Padova, ‘Communauté Université Grenoble Alpes

12.12.2019