

2 Esercizio 2: Metodo dei Momenti

Il metodo dei momenti è una tecnica numerica che viene utilizzata per la soluzione delle equazioni di Maxwell: nella fattispecie esso è adoperato per risolvere l'equazione di Hallén. L'equazione di Hallén è un'equazione integrale appartenente alla famiglia delle equazioni di Fredholm di prima specie, non risolubili analiticamente, ma solo per via numerica:

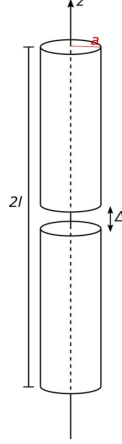


Figura 7: Schematizzazione dell'antenna filiforme in esame

$$\frac{\mu}{4\pi} \int_{-l}^l \frac{I(z')}{\sqrt{a^2 + (z - z')^2}} e^{-jk\sqrt{a^2 + (z - z')^2}} dz' = c \cos(kz) - j \frac{\omega \varepsilon \mu V_g}{2k} \sin(k|z|)$$

L'incognita nell'equazione di Hallén è la corrente $I(z')$.

La costante c può essere ricavata imponendo come condizioni al contorno per il problema l'annullamento della corrente agli estremi dell'antenna. Per risolvere l'equazione in questione con l'ausilio di un calcolatore è necessario *discretizzare* sia l'incognita che il termine noto in quanto il calcolatore può gestire solo un numero discreto di parametri.

Il metodo dei momenti viene applicato all'equazione di Fredholm di prima specie nella forma generale:

$$\int_0^1 k(z, z') I(z') dz' = f(z)$$

2.1 Discretizzazione dell'incognita

È possibile pensare ad una rappresentazione delle $I(z')$ come somma di funzioni di base, scelte in maniera arbitraria, pesate con dei coefficienti I_n :

$$I(z') = \sum_{n=1}^N I_n \varphi_n(z')$$

In queste condizioni, i coefficienti I_n diventano le incognite.

La scelta delle funzioni di base non è scontata in quanto da essa dipende l'accuratezza dell'approssimazione della funzione $I(z')$.

Una scelta semplice potrebbe essere quella di utilizzare funzioni *rect* di ampiezza unitaria, i cui supporti sono tutti adiacenti tra loro. Si ottengono così funzioni φ_n tra loro ortogonali in L^2 :

$$\int_0^1 \varphi_i(z) \varphi_j^*(z) dz = 0 \quad \text{se} \quad i \neq j$$

Nel momento in cui la funzione da approssimare varia velocemente rispetto alla variabile indipendente, la rappresentazione mediante funzioni rettangolari potrebbe non fornire più un risultato ragionevolmente accurato.

Una soluzione a questo problema è quella di approssimare $I(z')$ con una funzione lineare a tratti. Un'approssimazione di questo tipo consiste nello scegliere degli impulsi triangolari come funzioni di base. Tuttavia tale scelta implica un aumento della complessità computazionale, a causa del fatto che funzioni di questo tipo sono più complicate da trattare numericamente.

Per approssimazioni più accurate è possibile scegliere come funzioni di base:

- Polinomi di ordine superiore
- Polinomi di Bessel
- Sinusoidi a frequenza variabile (quando la funzione da approssimare ha una velocità di variazione elevata)

Appare chiaro dunque che, scegliendo funzioni di base particolarmente complicate, migliora l'accuratezza dell'approssimazione ma aumenta la complessità computazionale.

Discretizzate le incognite si ottiene:

$$\sum_{n=1}^N \int_0^1 k(z, z') \varphi_n(z') dz' = \sum_{n=1}^N G_n(z) I_n = f(z)$$

2.2 Discretizzazione delle funzioni G_n ed f

Per poter risolvere l'equazione al calcolatore, anche le funzioni $G_n(z)$ ed $f(z)$ vanno discretizzate.

Si impone l'uguaglianza tra i coefficienti delle funzioni in una base, calcolati sfruttando il prodotto scalare sullo spazio funzionale L^2 :

$$\left\langle \sum_{n=1}^N I_n G_n(z), \psi_k(z) \right\rangle = \langle f(z), \psi_k(z) \rangle$$

In principio lo spazio funzionale di appartenenza delle due funzioni è a dimensione infinita, ma per la discretizzazione si considera un numero finito di vettori di base. Si perviene dunque ad un sistema di N equazioni in N incognite, risolubile tramite algoritmi con un'accettabile complessità computazionale, come *l'eliminazione gaussiana*. Molto spesso come funzioni ψ si scelgono le Delta di Dirac, in tal caso si parla di *point matching*:

$$\psi_k(z) = \delta(z - z_k)$$

2.3 Implementazione dell'algoritmo in MATLAB™

Definizione dei parametri del problema Si inizia col definire i dati del problema in esame:

```
%% Definizione parametri elettromagnetici
freq=300E6;
c=3E8;
lambda=c/freq;
mi=4*pi*1E-7;
k=2*pi/lambda;
epsilon=8.854E-12;
omega=2*pi*freq;
%% Definizione dimensioni dell'antenna
l=lambda/4;
a=lambda/100;
%% Alimentazione
Vg=1;
```

Tra i parametri sono definiti la frequenza di lavoro, le costanti elettromagnetiche e le dimensioni elettriche dell'antenna da studiare ($2l = \lambda/2$, $a = \lambda/100$).

Discretizzazione delle incognite Un modo per discretizzare la variabile z' è quello di utilizzare delle funzioni costanti a tratti. Per fare ciò si divide il dominio di integrazione in segmenti di lunghezza pari a $\delta z = \lambda/30$. Si definisce un vettore zm di elementi equispaziati di $\lambda/30$ che vanno da $-l$ a l , il cui generico elemento rappresenta l'estremo sinistro di ogni intervallo.

```
%% Discretizzazione incognite
delta_z = lambda/30;
N = 2*l/delta_z;
zm = linspace(-l, l-delta_z, N);
```

Discretizzazione del termine noto Il termine noto può essere discretizzato utilizzando il metodo di *point-matching*, che impone l'uguaglianza puntuale tra i due membri dell'equazione, valutati, in questo caso, nei punti medi di ciascun intervallo. Tali punti sono ottenuti sommando $\delta z/2$ agli elementi del vettore zm , definito in precedenza:

```
%% Discretizzazione termine noto
z_mp = zm + delta_z/2;
```

L'uguaglianza da imporre diviene:

$$\sum_{n=0}^{N-1} I_n \int_{zm_n}^{zm_{n+1}} \frac{\mu}{4\pi} \frac{e^{-jk\sqrt{a^2+(zm_p-z')^2}}}{\sqrt{a^2+(zm_p-z')^2}} dz' = c \cos kz_{mp} - j \frac{\omega \epsilon \mu V_g}{2k} \sin kz_{mp}$$

I singoli risultati degli integrali al variare di n possono essere collezionati in una matrice $\underline{\underline{A}}$. Si è ricondotta così l'equazione integrale ad un sistema lineare di N equazioni nelle N incognite I_n . La risoluzione dell'integrale è affidata ad un metodo di quadratura, in particolare si impiega il metodo di *quadratura gaussiana*. Tale metodo permette di integrare in maniera esatta dei polinomi che soddisfino tale condizione:

$$M \geq \frac{p+1}{2}$$

in cui M è il numero di nodi utilizzato per la quadratura e p è il grado del polinomio. Si impone la convergenza dell'integrale al valore:

$$\int_{-1}^1 f(x)dx = \sum_{i=0}^{M-1} w_i f(x_i)$$

dove i coefficienti w_i da determinare e i punti x_i in cui valutare la funzione f sono definiti a partire dal numero di nodi e dal grado del polinomio. Per il problema in questione, nell'applicare tale metodo, si considera un numero M di nodi pari a 2, e i seguenti valori:

```
%% Valori per la quadratura
w1 = 0.5;
w2 = w1;
x1 = (-1/sqrt(3) + 1)*0.5;
x2 = (1/sqrt(3) + 1)*0.5;
```

Prima di procedere con l'integrazione si può far vedere che gli elementi della diagonale sono uguali ad un certo valore, che può essere escluso dalla quadratura per risparmiare conti:

$$A_{i,i} = \frac{\mu}{4\pi} \left[\log \frac{\sqrt{1 + \left(\frac{4a^2}{\Delta z^2}\right)} + 1}{\sqrt{1 + \left(\frac{4a^2}{\Delta z^2}\right)} - 1} - \frac{jk\Delta z}{4\pi} \right]$$

Successivamente si passa al riempimento della matrice $\underline{\underline{A}}$:

```
%% Fill di A
A=zeros(N);
for zi=1:N
    for zj=1:N
        if zj==zi
            A(zi,zj)= (mi/(4*pi))*log((sqrt(1+((4*a^2)/delta_z^2))
+1)/(sqrt(1+(4*a^2)/delta_z^2)-1))-((1i*k*delta_z)/(4*pi));
        else
            A(zi,zj) =mi/(4*pi)*((w1*delta_z*integrando(delta_z,k,a
,z_mp(zj),zm(zi),x1))+ w2*delta_z*(integrando(delta_z,k,a,z_mp(
zj),zm(zi),x2)));
        end
    end
end
```

in cui la funzione `integrando()`, è così definita:

```
function f_int=integrando(delta_z,k,a,z_mp,zm,x)
f_int=(exp(-1i*k*sqrt(a.^2+(z_mp-x*delta_z-zm)^2))/(sqrt(a.^2+(
z_mp-x*delta_z-zm)^2)))
end
```

Trovata la matrice $\underline{\underline{A}}$, si può proseguire alla sua inversione per determinare prima il valore di c e poi il vettore incognito $\underline{\underline{I}}$:

```
%% Calcolo della C
u = zeros(1, N);
u(1) = 1;
u(N) = 1;
c=(cos(k*z_mp))';
s=(-1i*((2*pi)*freq)*epsilon*Vg)/(2*k)*sin(k*abs(zm))';
C=-(u*inv(A)*s)/(u*inv(A)*c)
```

```
%% Calcolo di I
I=inv(A)*C*c+inv(A)*s
```

Determinati i valori di \underline{I} , questi possono essere rappresentati graficamente:

```
%% Plot
plot(zm/lambda, abs(I))
xlabel('$l (\lambda)$', 'Interpreter', 'latex', 'FontSize', 15);
ylabel('$|I(l)|$', 'Interpreter', 'latex', 'FontSize', 15);
```

Si ottiene il seguente andamento:

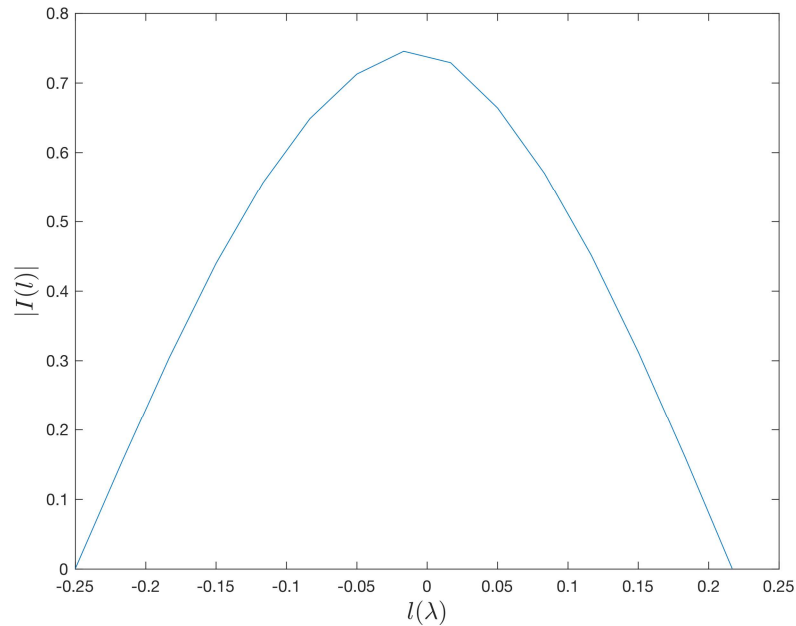


Figura 8: Andamento del modulo della corrente lungo l'antenna

2.4 Ottimizzazione dell'algoritmo

La matrice \underline{A} risulta essere una matrice di *Toeplitz*.

Una matrice \underline{A} $n \times n$ si definisce di *Toeplitz* se l'elemento:

$$a_{i,j} = a_{i-1,j-1} \quad (1)$$

Dunque un esempio può essere la matrice così composta:

$$\underline{A} = \begin{pmatrix} a & b & c \\ b & a & b \\ c & b & a \end{pmatrix}$$

È chiaro a questo punto che, nota la prima riga, è possibile costruire l'intera matrice senza effettuare ulteriori calcoli per gli elementi delle righe successive. In particolare *MATLAB*[™], mette a disposizione una funzione apposita che costruisce la matrice di Toeplitz dato un vettore riga:

```
A=toeplitz(T);
```

Perciò, nel caso in analisi, è possibile ottimizzare l'algoritmo riempiendo la prima riga della matrice $\underline{\underline{A}}$, richiamando in seguito `toeplitz()`:

```
for zj=1:N
    if zj==1
        T(1,zj)= (mi/(4*pi))*log((sqrt(1+((4*a^2)/delta_z^2))
+1)/(sqrt(1+(4*a^2)/delta_z^2)-1))-((1i*k*delta_z)/(4*pi));
    else
        T(1,zj) =mi/(4*pi)*((w1*delta_z*integrando(delta_z,k,a
,z_mp(zj),zm(1),x1))+ w2*delta_z*(integrando(delta_z,k,a,z_mp(
zj),zm(1),x2)));
    end
end
A=toeplitz(T);
```

Mediante l'utilizzo del timer di MATLAB `tic toc`, si può osservare una diminuzione dei tempi di riempimento della matrice di quasi un ordine di grandezza già per matrici relativamente piccole.

Ad esempio nel caso in cui $\underline{\underline{A}}$ sia una matrice 50×50 :

```
>> t_toeplitz = 0.01042
>> t_notoeplitz = 0.0780
```

Lo scarto tra i due tempi cresce all'aumentare delle dimensioni delle matrici.