

Animazione dell'andamento della tensione e della corrente lungo la linea

La simulazione prevedeva la creazione di un'animazione riguardante la propagazione della tensione e della corrente lungo una linea di trasmissione nel dominio del tempo applicando la tecnica **FDTD**. Bisognava prendere in considerazione un cavo coassiale di lunghezza 3m, le cui dimensioni erano:

- Raggio interno $a = 2.8\text{mm}$;
- Raggio esterno $b = 10\text{mm}$;
- Dielettrico interno che ha una $\epsilon_{\text{psr}} = 2.3$;

Bisognava considerare due tipologie di segnali con un generatore la cui resistenza interna è adattata alla linea:

1. Segnale gaussiano di lunghezza $\sigma = 300\text{ps}$ (frequenza massima circa 1.3GHz), ritardo di t_0 .
2. Segnale sinusoidale di frequenza 1.4GHz .

Infine bisognava considerare quattro tipologie di carichi:

1. Resistenza adattata;
2. Resistenza disadattata;
3. Carico capacitivo;
4. Carico induttivo.

Tecnica FDTD

La tecnica FDTD è stata proposta da Yee, nel 1966. È attualmente una delle tecniche più utilizzate per la soluzione di problemi elettromagnetici.

Può essere applicata ad una ampia varietà di problemi complessi: radiazione di antenne in ambienti complessi, problemi biomedici, modellizzazione di circuiti a microonde, problemi di compatibilità elettromagnetica ecc.

Vantaggi FDTD

1. È una soluzione diretta delle equazioni di Maxwell, non richiede l'inversione di matrici.
2. L'implementazione del metodo è semplice ed intuitiva.
3. È applicabile a qualsiasi geometria 3D.
4. Può essere applicata a materiali dielettrici reali di qualunque tipo.

Svantaggi FDTD

1. Richiede che venga definito sia l'oggetto che lo spazio circostante.
2. Le superfici curve vengono simulate con delle approssimazioni in quanto la griglia di discretizzazione è rettangolare.
3. Richiede che vengano definite le condizioni di assorbimento che simulano la radiazione all'infinito.
4. Il codice finale è complesso.

Applicazione della tecnica FDTD

Un punto nello spazio discretizzato viene indicato in questo modo:

$$(i, j, k) = (i\Delta x, j\Delta y, k\Delta z)$$

Ogni grandezza calcolata in un punto dello spazio discretizzato e del tempo discretizzato viene indicato come:

$$u(i\Delta x, j\Delta y, k\Delta z, n\Delta t) = u_{i,j,k}^n$$

Differenze finite:

$$\frac{df(x_0)}{dx} = f'(x_0) \cong \frac{f(x_0 + \Delta x/2) - f(x_0 - \Delta x/2)}{\Delta x}$$

Per calcolare la derivata di una funzione, con la tecnica FDTD, si utilizzano le **Differenze finite centrali** sia nel tempo che nello spazio:

$$\frac{\partial u}{\partial t}(i\Delta x, j\Delta y, k\Delta z, n\Delta t) = \frac{u_{i,j,k}^{n+1/2} - u_{i,j,k}^{n-1/2}}{\Delta t} + O\left[(\Delta t)^2\right]$$

$$\frac{\partial u}{\partial x}(i\Delta x, j\Delta y, k\Delta z, n\Delta t) = \frac{u_{i+1/2,j,k}^n - u_{i-1/2,j,k}^n}{\Delta x} + O\left((\Delta x)^2\right)$$

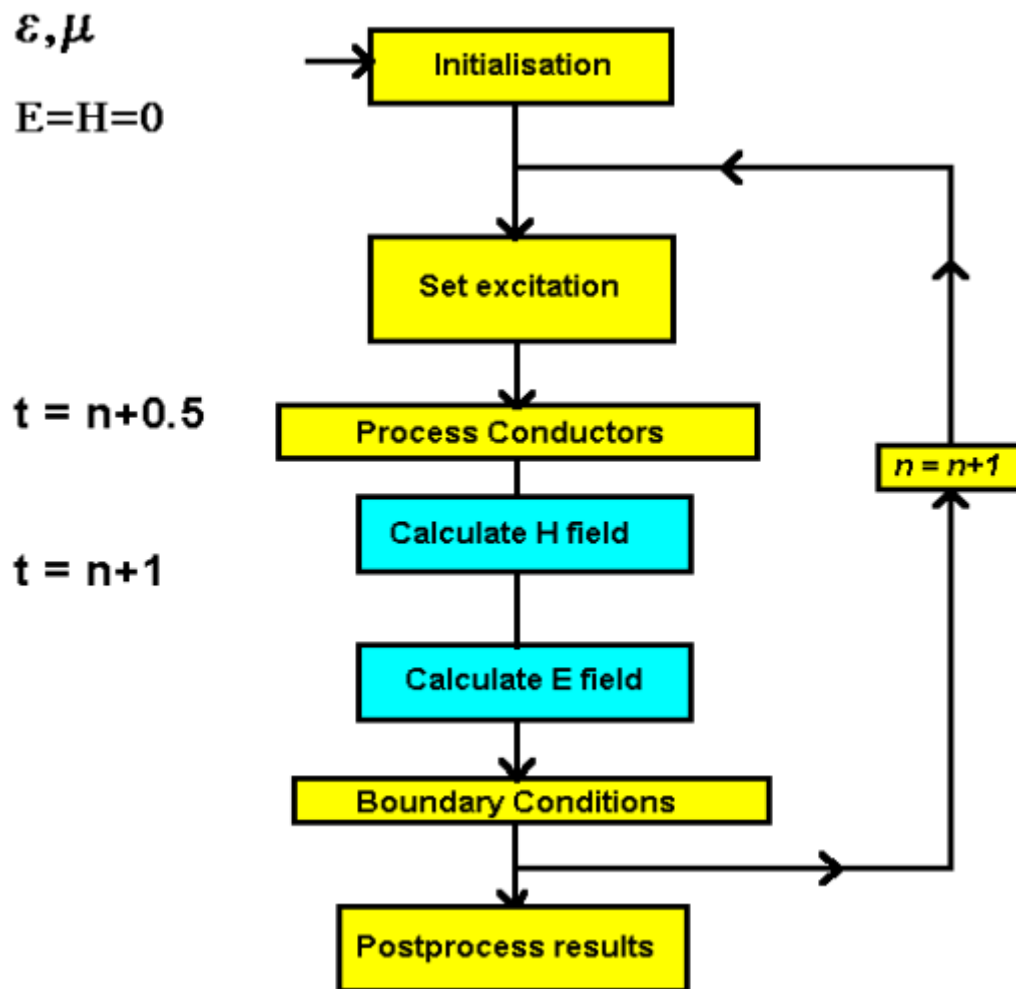
CAMPO ELETTRICO

$$\left. \begin{aligned} \frac{\partial E_x}{\partial t} &= \frac{1}{\varepsilon} \left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} - \sigma E_x \right) \\ \frac{\partial E_y}{\partial t} &= \frac{1}{\varepsilon} \left(\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} - \sigma E_y \right) \\ \frac{\partial E_z}{\partial t} &= \frac{1}{\varepsilon} \left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} - \sigma E_z \right) \end{aligned} \right\}$$

CAMPO MAGNETICO

$$\left. \begin{aligned} \frac{\partial H_x}{\partial t} &= \frac{1}{\mu} \left(\frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y} \right) \\ \frac{\partial H_y}{\partial t} &= \frac{1}{\mu} \left(\frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} \right) \\ \frac{\partial H_z}{\partial t} &= \frac{1}{\mu} \left(\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} \right) \end{aligned} \right\}$$

Come implementare l'FDTD



L'algoritmo YEE restituisce una situazione stabile solo se

$$\Delta t \leq \frac{1}{c} \frac{1}{\sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}}}$$

Per ottenere dei risultati accurati occorre che

$$\Delta x_{\max}, \Delta y_{\max} \text{ and } \Delta z_{\max} \leq \frac{1}{20} \lambda_{\min} \quad \lambda_{\min} = \frac{v_c}{f_{\max} \sqrt{\mu_{r\max} \epsilon_{r\max}}}$$

Le condizioni al contorno della linea sono rappresentate dalle equazioni caratteristiche dei carichi e dal generatore

Carico resistivo $V = RI \quad V(i, k) = RI \left(i + \frac{1}{2}, k - \frac{1}{2} \right)$

Carico induttivo $V = L \frac{\partial I}{\partial t}$

$$V(i, k) = L \frac{I \left(i + \frac{1}{2}, k + \frac{1}{2} \right) - I \left(i + \frac{1}{2}, k - \frac{1}{2} \right)}{\Delta t}$$

Carico Capacitivo $I = C \frac{\partial V}{\partial t}$

$$I \left(i + \frac{1}{2}, k + \frac{1}{2} \right) = C \frac{V(i, k + 1) - V(i, k)}{\Delta z}$$

Descrizione codice

Il nostro codice è composto principalmente da tre sezioni. Nella prima parte del codice sono stati inseriti i vari dati e definite le variabili del programma, nella seconda parte tramite un ciclo for vengono specificate le condizioni al contorno della linea e, nell'ultima parte, vengono descritte due funzioni che vengono richiamate all'interno del ciclo.

Nella **prima parte** del codice abbiamo definito, oltre ai dati forniti dalla richiesta, dei valori che ci sono stati utili nelle successive parti del codice: *epsylon0*, *mu0*, *C* (capacità), *L* (induttanza).

```
8 -     epsylon0 = 8.85e-12;
9 -     mu0 = 4 * pi * 1e-7;
10 -    C = (epsylon0 * 2.3 * 2 * pi) / log(10/2.8);
11 -    L = mu0 / (2 * pi) * log(10/2.8);
```

Nella **seconda parte** del codice abbiamo implementato la **tecnica FDTD**, inserendo il numero di suddivisioni per lo spazio ed inserendo il tempo di simulazione e la discretizzazione del tempo per il tempo. Successivamente abbiamo definito, come richiesto la **resistenza disadattata** e la **resistenza adattata**.

```
29 - R_load = 5e1;  
30 - R_load = sqrt(L/C);
```

Il **ciclo for** svolge, in ordine, la funzione di calcolo dell'equazione caratteristica dei **carichi** (sia **induttivo**, che **capacitivo**) e del **generatore** (sia in caso di **induttore**, che di **condensatore**); successivamente salva gli andamenti della linea di trasmissione.

```
60 - V(1) = Vin(t);  
61 - I(end) = (V(end)* dt) / L; % induttore, carico induttivo  
62 - [V, I] = my_step(V, I, L, C, dt, dz);
```

```
60 - V(1) = Vin(t);  
61 - V(end) = (I(end)* dt) / C; % conduttore, carico capacitivo  
62 - [V, I] = my_step(V, I, L, C, dt, dz);
```

Nella terza ed ultima parte del codice sono state implementate due funzioni: **tensioni_correnti** e **grafico_tensioni_correnti**.

La funzione **tensioni_correnti** analizza l'evoluzione del sistema e restituisce i valori di I e di V .

```
91 - function [V, I] = tensioni_correnti(V, I, L, C, dt, dz)  
92 -     N = numel(V);  
93 -  
94 -     dv_dz = diff(V) / dz;  
95 -     di_dz = diff(I) / dz;  
96 -  
97 -     I(1:end-1) = I(1:end-1) - (dv_dz / L) * dt;  
98 -     V(2:end) = V(2:end) - (di_dz / C) * dt;  
99 - end
```

Mentre la funzione ***grafico_tensioni_correnti*** si occupa di assimilare i valori di V e di I e riportarli all'interno di un grafico che descrive l'andamento delle linee.

```
101 - function [] = grafico_tensioni_correnti(xx, V, I)
102 -     xlabel("spazio [m]")
103 -     yyaxis left
104 -     plot(xx, V);
105 -     ylabel("tensione [V]")
106 -     yyaxis right
107 -     plot(xx, I);
108 -     ylabel("corrente [A]")
109 -     yyaxis left
110 -     %grid();
111 - end
```