



ECOLE  
POLYTECHNIQUE  
DE BRUXELLES

2024-2025

ELEC-304 : Physique des Télécommunications

---

## Résolution des équations de Maxwell par FDTD

---

Henry Chou

Philippe De Doncker



## Table des matières

<b>1</b>	<b>FDTD 1D</b>	<b>1</b>
1.1	Etape 1 : Espace Libre . . . . .	1
1.2	Etape 2 : diélectriques . . . . .	2
1.2.1	Cas de la permittivité relative . . . . .	2
1.2.2	Cas des milieux diélectriques avec pertes . . . . .	2
<b>2</b>	<b>FDTD 2D</b>	<b>3</b>
2.1	Etape 3 : Espace Libre . . . . .	3
2.1.1	Algorithme de Yee . . . . .	3
2.1.2	Amplitude d'oscillation du champ électrique . . . . .	5
2.1.3	Tension et puissance reçue par une antenne . . . . .	5
2.2	Etape 4 : Obstacles diélectriques et parfaitement conducteurs . . . . .	6
<b>3</b>	<b>Simulations</b>	<b>7</b>
3.1	Chapitre 7 . . . . .	7
3.2	Chapitre 8 . . . . .	8
3.3	Chapitre 9 . . . . .	10
<b>4</b>	<b>code</b>	<b>12</b>

## 1.1 Etape 1 : Espace Libre

Pour cette étape, les développements théoriques étant fournis dans l'énoncé du projet, ils ne sont pas répétés ici. Voici les résultats obtenus pour les 2 sources :

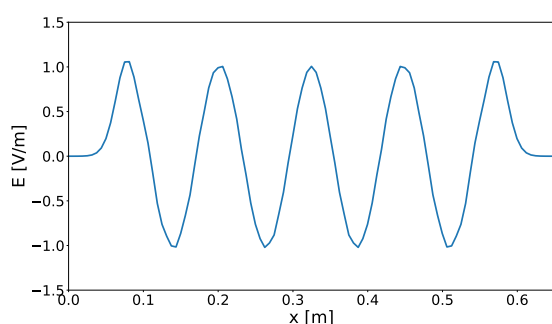


FIGURE 1.1 :  $E_z$  pour une source sinusoïdale

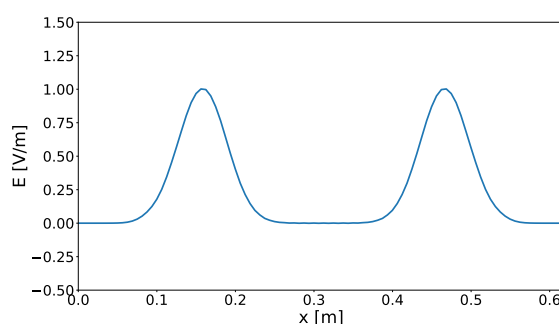


FIGURE 1.2 :  $E_z$  pour une source gaussienne

Pour la source sinusoïdale, on évalue l'amplitude du champ électrique en simulant sur une durée suffisamment longue pour que le régime soit établi, c'est-à-dire que l'onde se soit propagée dans tout le domaine. Il suffit de prendre la valeur maximale, en soustraire la valeur minimale et diviser par 2 pour chaque point de la grille. On obtient la figure suivante :

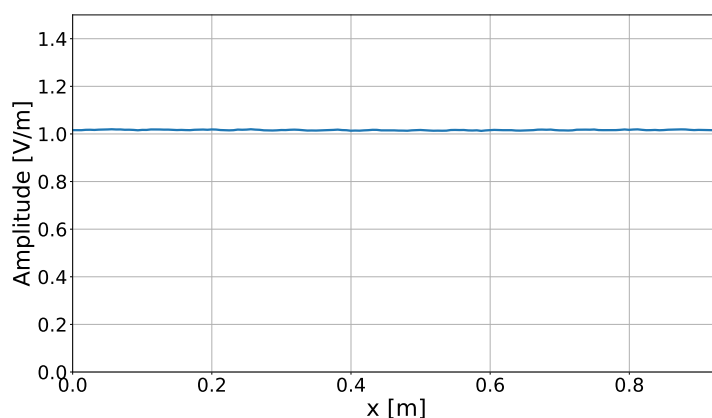


FIGURE 1.3 : Amplitude de  $E_z$

L'amplitude est constante sur toute la grille, ce qui est cohérent puisque nous sommes en espace libre avec des ondes planes, il n'y a pas d'atténuation. Les petites variations sont causées par des erreurs numériques.

Si l'on impose des conditions limites nulles, cela fait rebondir l'onde aux limites du domaine. En effet, l'onde réfléchie d'amplitude opposée interfère avec l'onde incidente, ce qui permet de respecter les conditions aux bords nulles. Physiquement, cela se produit si l'on place un conducteur parfait aux limites du domaine, ses charges se mettent en oscillation pour créer une onde parfaitement opposée à l'onde incidente. Ce phénomène s'observe plus simplement avec l'impulsion gaussienne, on voit les fronts d'ondes qui reviennent et qui interfèrent lorsqu'ils se croisent tout en continuant leur chemin. Dans le cas de la source sinusoïdale le comportement est un peu plus chaotique étant donné que la source continue à envoyer des ondes.

## 1.2 Etape 2 : diélectriques

### 1.2.1 Cas de la permittivité relative

Il suffit de remplacer  $\varepsilon_0$  par  $\varepsilon$  dans les équations de Maxwell, on obtient :

$$\begin{cases} \vec{\nabla} \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \\ \vec{\nabla} \times \vec{B} = \varepsilon \mu_0 \frac{\partial \vec{E}}{\partial t} + \mu_0 \vec{J} \end{cases} \quad (1.1)$$

Seule la constante  $\varepsilon_0$  a changé, on peut donc reprendre les résultats de l'étape 1 et ajuster les équations.

$$E_z^{q+1}[m] = E_z^q[m] + \frac{1}{\varepsilon[m]\mu_0} \frac{\Delta t}{c\Delta x} \left( \tilde{B}_y^{q+1/2}[m+1/2] - \tilde{B}_y^{q+1/2}[m-1/2] \right) - \frac{\Delta t}{\varepsilon[m]} J_z^{q+1/2}[m] \quad (1.2)$$

$$\tilde{B}_y^{q+1/2}[m+1/2] = \tilde{B}_y^{q-1/2}[m+1/2] + \frac{c\Delta t}{\Delta x} (E_z^q[m+1] - E_z^q[m]) \quad (1.3)$$

Sachant que  $c = \frac{1}{\sqrt{\varepsilon_0\mu_0}}$ ,  $\varepsilon = \varepsilon_r\varepsilon_0$  et  $\Delta t = \frac{\Delta x}{2c}$ , on peut simplifier :

$$E_z^{q+1}[m] = E_z^q[m] + \frac{1}{2\varepsilon_r[m]} \left( \tilde{B}_y^{q+1/2}[m+1/2] - \tilde{B}_y^{q+1/2}[m-1/2] \right) - \frac{\Delta t}{\varepsilon_r[m]\varepsilon_0} J_z^{q+1/2}[m] \quad (1.4)$$

$$\tilde{B}_y^{q+1/2}[m+1/2] = \tilde{B}_y^{q-1/2}[m+1/2] + \frac{1}{2} (E_z^q[m+1] - E_z^q[m]) \quad (1.5)$$

Si l'on considère des diélectriques sans pertes, c'est-à-dire que  $\varepsilon$  est réelle, alors les équations ci-dessus sont suffisantes. En revanche, pour prendre en compte les pertes diélectriques, comme dans le cas de la cuisson d'un aliment au micro-ondes, il faut introduire une conductivité équivalente.

### 1.2.2 Cas des milieux diélectriques avec pertes

Dans un milieu avec pertes, nous avons  $\vec{J} = \sigma \vec{E}$ . Si l'on considère  $E_z^q[m] = E_z^{q+1/2}[m]$ , on obtient en modifiant 1.4 :

$$\begin{aligned} E_z^{q+1}[m] &= E_z^q[m] + \frac{1}{2\varepsilon_r[m]} \left( \tilde{B}_y^{q+1/2}[m+1/2] - \tilde{B}_y^{q+1/2}[m-1/2] \right) - \frac{\Delta t}{\varepsilon_r[m]\varepsilon_0} \sigma E_z^q[m] \\ &= \left( 1 - \frac{\sigma \Delta t}{\varepsilon_r[m]\varepsilon_0} \right) E_z^q[m] + \frac{1}{2\varepsilon_r[m]} \left( \tilde{B}_y^{q+1/2}[m+1/2] - \tilde{B}_y^{q+1/2}[m-1/2] \right) \end{aligned} \quad (1.6)$$

## 2.1 Etape 3 : Espace Libre

Partons des équations de Maxwell en rotationnel dans le vide :

$$\begin{cases} \vec{\nabla} \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \\ \vec{\nabla} \times \vec{B} = \varepsilon_0 \mu_0 \frac{\partial \vec{E}}{\partial t} + \mu_0 \vec{J} \end{cases} \quad (2.1)$$

Nous cherchons les solutions pour lesquelles le champ électrique est polarisé selon  $\vec{I}_z$  :  $\vec{E} = E_z(\vec{x}, t) \vec{I}_z$ .  
Calculons son rotationnel :

$$\vec{\nabla} \times \vec{E} = \frac{\partial E_z}{\partial y} \vec{I}_x - \frac{\partial E_z}{\partial x} \vec{I}_y \quad (2.2)$$

On peut donc en déduire que  $\vec{B} = B_x(\vec{x}, t) \vec{I}_x + B_y(\vec{x}, t) \vec{I}_y$  et calculer son rotationnel :

$$\vec{\nabla} \times \vec{B} = -\frac{\partial B_y}{\partial z} \vec{I}_x + \frac{\partial B_x}{\partial z} \vec{I}_y + \left( \frac{\partial B_y}{\partial x} - \frac{\partial B_x}{\partial y} \right) \vec{I}_z \quad (2.3)$$

Comme  $\vec{E}$  est polarisé selon  $\vec{I}_z$ , le rotationnel de  $\vec{B}$  l'est aussi. Les composantes selon  $\vec{I}_x$  et  $\vec{I}_y$  sont donc nulles. On a alors :

$$\begin{cases} \frac{\partial E_z}{\partial y} = -\frac{\partial B_x}{\partial t} \\ \frac{\partial E_z}{\partial x} = \frac{\partial B_y}{\partial t} \\ \frac{\partial B_y}{\partial x} - \frac{\partial B_x}{\partial y} = \varepsilon_0 \mu_0 \frac{\partial E_z}{\partial t} + \mu_0 J_z \end{cases} \quad (2.4)$$

### 2.1.1 Algorithme de Yee

Soit la notation discrète<sup>1</sup> :

$$E_z^q[m, n] = E_z(m\Delta x, n\Delta y, q\Delta t) \quad (2.5)$$

Evaluons la troisième équation de 2.4 en  $(m\Delta x, n\Delta y)$  à l'instant  $(q + 1/2)\Delta t$  :

$$\left. \frac{\partial B_y}{\partial x} \right|_{m\Delta x, n\Delta y, (q+1/2)\Delta t} - \left. \frac{\partial B_x}{\partial y} \right|_{m\Delta x, n\Delta y, (q+1/2)\Delta t} = \varepsilon_0 \mu_0 \left. \frac{\partial E_z}{\partial t} \right|_{m\Delta x, n\Delta y, (q+1/2)\Delta t} + \mu_0 J_z|_{m\Delta x, n\Delta y, (q+1/2)\Delta t} \quad (2.6)$$

1. Dans le code, les indices sont inversés pour correspondre aux lignes et colonnes d'une matrice

Appliquons la différence finie centrée et la notation discrète :

$$\begin{aligned} & \frac{B_y^{q+1/2}[m+1/2, n] - B_y^{q+1/2}[m-1/2, n]}{\Delta x} - \frac{B_x^{q+1/2}[m, n+1/2] - B_x^{q+1/2}[m, n-1/2]}{\Delta y} \\ &= \varepsilon_0 \mu_0 \frac{E_z^{q+1}[m, n] - E_z^q[m, n]}{\Delta t} + \mu_0 J_z^{q+1/2}[m, n] \end{aligned} \quad (2.7)$$

En isolant  $E_z^{q+1}[m, n]$  :

$$\begin{aligned} E_z^{q+1}[m, n] &= E_z^q[m, n] - \frac{\Delta t}{\varepsilon_0} J_z^{q+1/2}[m, n] \\ &+ \frac{1}{\varepsilon_0 \mu_0} \frac{\Delta t}{\Delta x} (B_y^{q+1/2}[m+1/2, n] - B_y^{q+1/2}[m-1/2, n]) \\ &- \frac{1}{\varepsilon_0 \mu_0} \frac{\Delta t}{\Delta y} (B_x^{q+1/2}[m, n+1/2] - B_x^{q+1/2}[m, n-1/2]) \end{aligned} \quad (2.8)$$

De manière équivalente, considérons les 2 premières équations de 2.4 :

$$\left. \frac{\partial E_z}{\partial y} \right|_{m\Delta x, (n+1/2)\Delta y, q\Delta t} = - \left. \frac{\partial B_x}{\partial t} \right|_{m\Delta x, (n+1/2)\Delta y, q\Delta t} \quad (2.9)$$

$$\left. \frac{\partial E_z}{\partial x} \right|_{(m+1/2)\Delta x, n\Delta y, q\Delta t} = \left. \frac{\partial B_y}{\partial t} \right|_{(m+1/2)\Delta x, n\Delta y, q\Delta t} \quad (2.10)$$

En appliquant les différences finies :

$$\frac{E_z^q[m, n+1] - E_z^q[m, n]}{\Delta y} = - \frac{B_x^{q+1/2}[m, n+1/2] - B_x^{q-1/2}[m, n+1/2]}{\Delta t} \quad (2.11)$$

$$\frac{E_z^q[m+1, n] - E_z^q[m, n]}{\Delta x} = \frac{B_y^{q+1/2}[m+1/2, n] - B_y^{q-1/2}[m+1/2, n]}{\Delta t} \quad (2.12)$$

Isolons  $B_y$  et  $B_x$  :

$$B_x^{q+1/2}[m, n+1/2] = B_x^{q-1/2}[m, n+1/2] - \frac{\Delta t}{\Delta y} (E_z^q[m, n+1] - E_z^q[m, n]) \quad (2.13)$$

$$B_y^{q+1/2}[m+1/2, n] = B_y^{q-1/2}[m+1/2, n] + \frac{\Delta t}{\Delta x} (E_z^q[m+1, n] - E_z^q[m, n]) \quad (2.14)$$

Finalement, sachant que  $\tilde{B} = cB$ ,  $c = \frac{1}{\sqrt{\varepsilon_0 \mu_0}}$  et  $\Delta t = \frac{\Delta x}{2c} = \frac{\Delta y}{2c}$ , on peut simplifier :

$$\begin{aligned} E_z^{q+1}[m, n] &= E_z^q[m, n] - \frac{\Delta t}{\varepsilon_0} J_z^{q+1/2}[m, n] \\ &+ \frac{1}{2} (\tilde{B}_y^{q+1/2}[m+1/2, n] - \tilde{B}_y^{q+1/2}[m-1/2, n]) \\ &- \frac{1}{2} (\tilde{B}_x^{q+1/2}[m, n+1/2] - \tilde{B}_x^{q+1/2}[m, n-1/2]) \end{aligned} \quad (2.15)$$

$$\tilde{B}_x^{q+1/2}[m, n+1/2] = \tilde{B}_x^{q-1/2}[m, n+1/2] - \frac{1}{2} (E_z^q[m, n+1] - E_z^q[m, n]) \quad (2.16)$$

$$\tilde{B}_y^{q+1/2}[m+1/2, n] = \tilde{B}_y^{q-1/2}[m+1/2, n] + \frac{1}{2} (E_z^q[m+1, n] - E_z^q[m, n]) \quad (2.17)$$

### 2.1.2 Amplitude d'oscillation du champ électrique

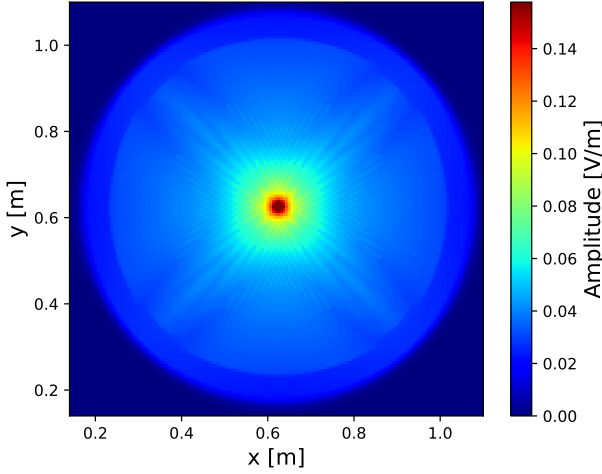


FIGURE 2.1 : Ensemble de la grille

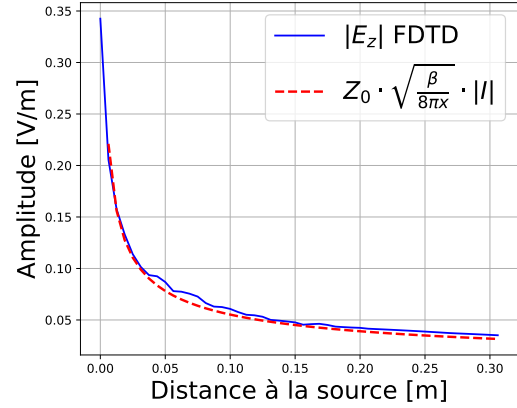


FIGURE 2.2 : Vue en coupe

On observe une décroissance de l'amplitude en fonction de la distance à la source. Une coupe du champ permet de vérifier que le profil obtenu en simulation correspond au profil théorique d'une ligne de courant infinie. L'expression théorique est obtenue en prenant le module de  $\vec{E}_z$ , dont la forme est donnée dans le cours.

$$|E_z| = Z_0 \sqrt{\frac{\beta}{8\pi}} \frac{I}{\sqrt{d}} \quad (2.18)$$

Dans la simulation, on effectue une coupe selon  $x$ , la coordonnée radiale  $d$  devient  $x$ . Comme on fixe  $J$ , on trouve  $I$  en multipliant par la surface d'un élément de la grille discrète :  $I = J \cdot \Delta x \Delta y$ . La décroissance en  $\sqrt{1/d}$  est nécessaire afin de satisfaire la conservation de l'énergie. En effet, le vecteur de Poynting est proportionnel à  $|E_z|^2$ , il a donc une dépendance en  $1/d$ . Son intégrale (puissance rayonnée par unité de longueur) est donc indépendante du cercle d'intégration.

### 2.1.3 Tension et puissance reçue par une antenne

Considérons une antenne dipôle demi-longueur d'onde coupée en son centre par le plan de la simulation. On peut calculer la tension induite et la puissance reçue avec les formules suivantes :

$$h_{e\theta} = \frac{-\lambda \cos(\frac{\pi}{2} \cos \theta)}{\pi \sin \theta} \quad (2.19)$$

$$\underline{V}_{oc}(\vec{r}) = -\vec{h}_{e\perp}(\theta, \phi) \cdot \vec{E}(\vec{r}) \quad (2.20)$$

$$P_L = \frac{1}{2} \frac{R_L}{(R_a + R_L)^2 + (X_a + X_L)^2} |\underline{V}_{oc}|^2 \quad (2.21)$$

Dans le plan  $(x, y)$ , on a  $\theta = 90^\circ$ , ce qui donne  $h_{e\theta} = \frac{-\lambda}{\pi} = -h_{ez}$ . Considérons  $R_a = 73\Omega$ ,  $R_L = 50\Omega$  et  $X_a + X_L = 0$ . On a finalement :

$$|\underline{V}_{oc}| = \frac{\lambda}{\pi} \cdot |\underline{E}| \quad (2.22)$$

$$P_L = \frac{50}{2 \cdot 123^2} |\underline{V}_{oc}|^2 \quad (2.23)$$

L'amplitude de la tension a exactement la même allure que celle du champ électrique, puisqu'elle en dépend à une constante près. Au niveau de la puissance, on observe une décroissance plus marquée ( $1/r$ ), car le terme est pris au carré 2.3.

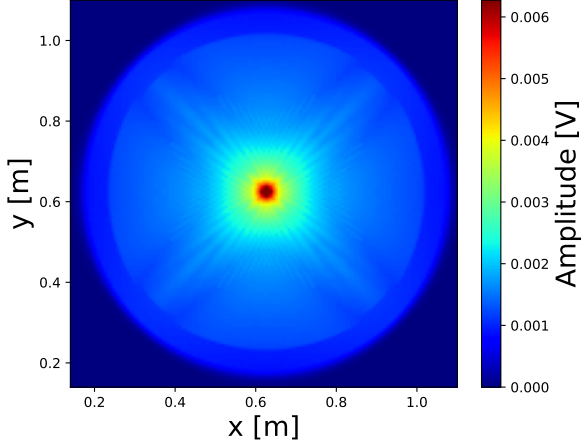


FIGURE 2.3 : Amplitude de la tension

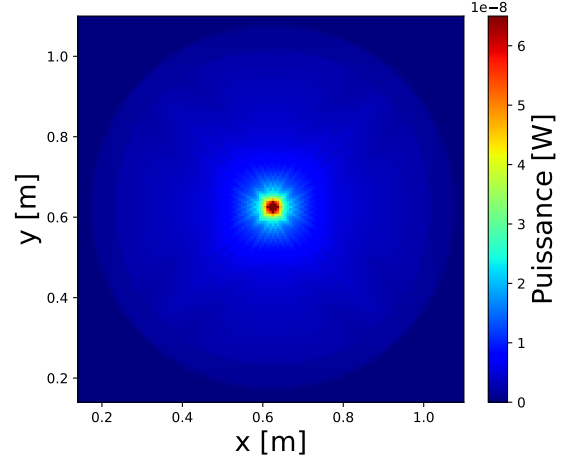


FIGURE 2.4 : Puissance reçue

## 2.2 Etape 4 : Obstacles diélectriques et parfaitement conducteurs

Pour implémenter des obstacles parfaitement conducteurs, il suffit d'imposer  $E_z = 0$  aux endroits où le conducteur est présent. Pour les milieux présentant une permittivité relative  $\varepsilon_r(x, y)$  et une conductivité  $\sigma(x, y)$ , on effectue une démarche similaire à l'étape 2 1.2 :

$$\begin{aligned}
 E_z^{q+1}[m, n] = & \left(1 - \frac{\sigma \Delta t}{\varepsilon_r[m, n] \varepsilon_0}\right) E_z^q[m, n] \\
 & + \frac{1}{2\varepsilon_r[m, n]} (\tilde{B}_y^{q+1/2}[m + 1/2, n] - \tilde{B}_y^{q+1/2}[m - 1/2, n]) \\
 & - \frac{1}{2\varepsilon_r[m, n]} (\tilde{B}_x^{q+1/2}[m, n + 1/2] - \tilde{B}_x^{q+1/2}[m, n - 1/2])
 \end{aligned} \tag{2.24}$$



### 3.1 Chapitre 7

Commençons par une situation simple afin de valider les performances de la simulation. Nous utilisons la FDTD 1D afin de vérifier la décroissance exponentielle dans les milieux avec pertes :

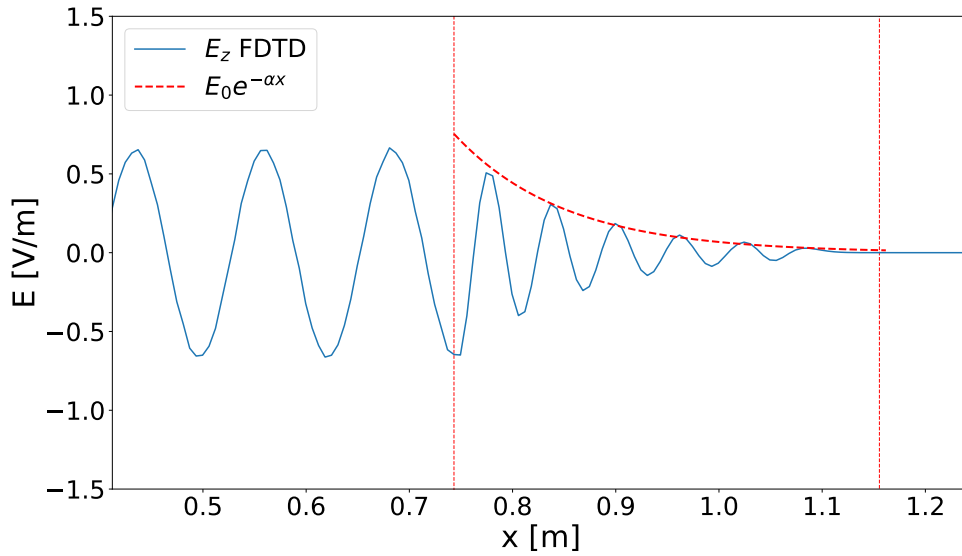


FIGURE 3.1 : Mur diélectrique  $\varepsilon_r = 4$  ;  $\sigma = 0.1S/m$

La décroissance simulée correspond à la décroissance théorique.  $E_0$  est l'amplitude du champ au début du mur. On trouve le facteur  $\alpha$  avec la formule suivante :

$$\alpha = \omega \sqrt{\frac{\mu_0 \varepsilon}{2}} \left[ \sqrt{1 + \left( \frac{\sigma}{\omega \varepsilon} \right)^2} - 1 \right]^{1/2} \quad (3.1)$$

On remarque également que l'onde se resserre, la longueur d'onde diminue dans le diélectrique. En effet,  $\lambda = 2\pi/\beta$ , mais  $\beta$  change en fonction du milieu :

$$\beta = \omega \sqrt{\frac{\mu_0 \varepsilon}{2}} \left[ \sqrt{1 + \left( \frac{\sigma}{\omega \varepsilon} \right)^2} + 1 \right]^{1/2} \quad (3.2)$$

Pour  $f = 2.4GHz$ , on obtient  $\lambda = 0.062m$ , ce qui correspond à la longueur d'onde mesurée sur la figure. Lorsque l'onde arrive sur le diélectrique, ses charges se mettent en oscillation et rayonnent des ondes. Il y a donc à chaque interface des réflexions qui modifient le profil du champ. La figure a donc été prise avant que l'onde n'atteigne la limite du mur afin de ne pas s'éloigner du profil théorique qui suppose un mur infini. Ce n'est pas visible sur la figure, mais on peut voir dans la simulation que l'amplitude du champ avant le début du mur varie à cause des interférences avec les réflexions.

## 3.2 Chapitre 8

Illustrons les lois de Snell dans des milieux non dissipatifs. On utilise une onde plane qui se propage selon  $-\vec{1}_y$  et un obstacle diélectrique incliné de sorte que  $\theta_i = 15^\circ$ .

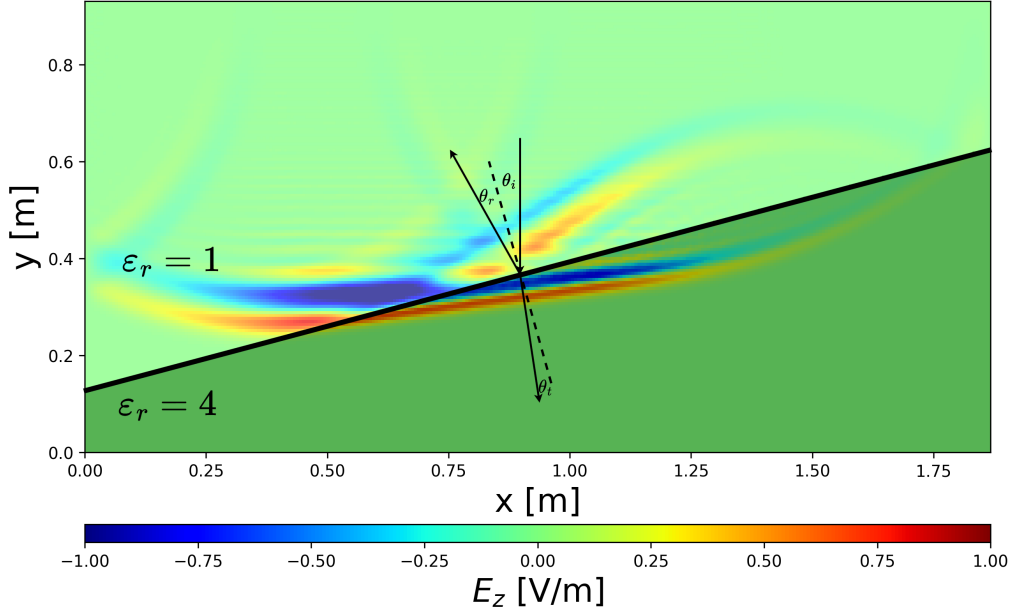


FIGURE 3.2 : Réflexion et transmission milieux diélectriques sans pertes

L'onde plane est créée dans le vide ( $\varepsilon_r = 1$ ) en alignant plusieurs sources sinusoïdales. Les bords ne sont donc pas parfaits, mais on peut se concentrer sur le milieu du front d'onde. Lorsque l'onde arrive à l'interface, une partie est réfléchiée selon un angle  $\theta_r = \theta_i$  et le reste est transmis selon l'angle  $\theta_t = 7^\circ$ . Calculons cet angle à l'aide de la loi de Snell :

$$\begin{aligned} \sqrt{\varepsilon_1} \sin \theta_i &= \sqrt{\varepsilon_2} \sin \theta_t \\ \Leftrightarrow \theta_t &= \arcsin\left(\sqrt{\frac{1}{4}} \sin \theta_i\right) = 7.43^\circ \end{aligned} \quad (3.3)$$

On obtient une valeur cohérente avec la théorie.

Si l'onde plane provient cette fois-ci du milieu de permittivité plus élevée et que  $\theta_i \geq \arcsin\left(\sqrt{\frac{\varepsilon_2}{\varepsilon_1}}\right) = 30^\circ$ , il y a réflexion totale, l'onde ne se propage pas dans le second milieu (figure 3.3). En effet, cette condition sur l'angle d'incidence implique  $\sin \theta_t > 1$ , ce qui n'est possible que si  $\theta_t$  est complexe. Cela se traduit par une exponentielle décroissante dans l'expression du champ transmis.

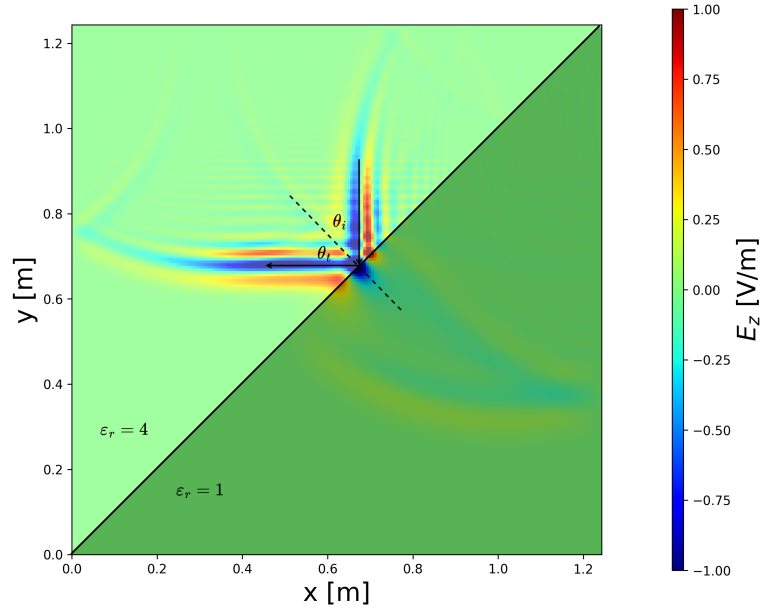


FIGURE 3.3 : Réflexion totale

Par rapport à la simulation précédente, les permittivités ont été inversées et l'angle de l'interface est désormais de  $45^\circ$ . On observe que l'onde incidente est entièrement réfléchiée, sans se propager dans le second milieu. L'atténuation exponentielle est clairement visible. Toutefois, une faible propagation dans le second milieu semble subsister ; il s'agit en réalité d'une imperfection de la simulation. En effet, l'onde incidente n'est pas parfaitement plane : ses bords sont arrondis, ce qui réduit localement l'angle d'incidence en dessous de l'angle critique.

### 3.3 Chapitre 9

Nous allons considérer la propagation entre deux plaques conductrices. Comme la simulation est réalisée en 2D, le champ électrique est polarisé selon  $\vec{I}_z$ . Dès lors, les modes TEM et TM ne peuvent pas exister, car ils nécessitent un champ polarisé dans le plan  $(x, y)$ . On se concentre donc sur le mode TE. Les conditions pour que les modes  $TE_n$  soient admissibles sont les suivantes :

$$\theta_n = \arccos \frac{n\pi}{\beta b} \quad f \geq \frac{nc}{2b} \quad \text{pour } n = 1, 2, \dots \quad (3.4)$$

La source utilisée étant une ligne de courant émettant dans toutes les directions, les conditions angulaires sont automatiquement satisfaites. On choisit donc le mode  $TE_n$ , en imposant la condition sur la fréquence. En posant  $b = M \cdot dy$  avec  $dy = \lambda/20$ , la condition s'écrit alors  $M \geq 10n$ . Il suffit simplement de changer le nombre de pas spatiaux entre les 2 plaques pour visualiser les différents modes.

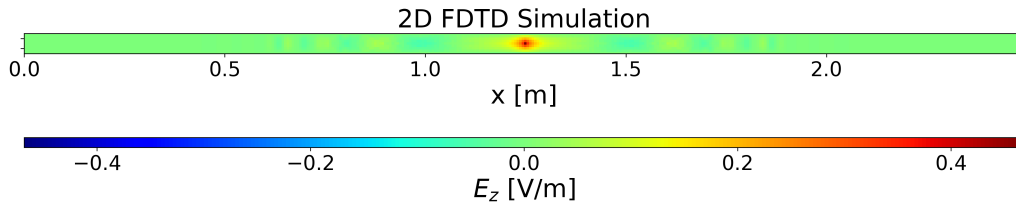


FIGURE 3.4 : Mode évanescent :  $0 < f < f_{c1}$

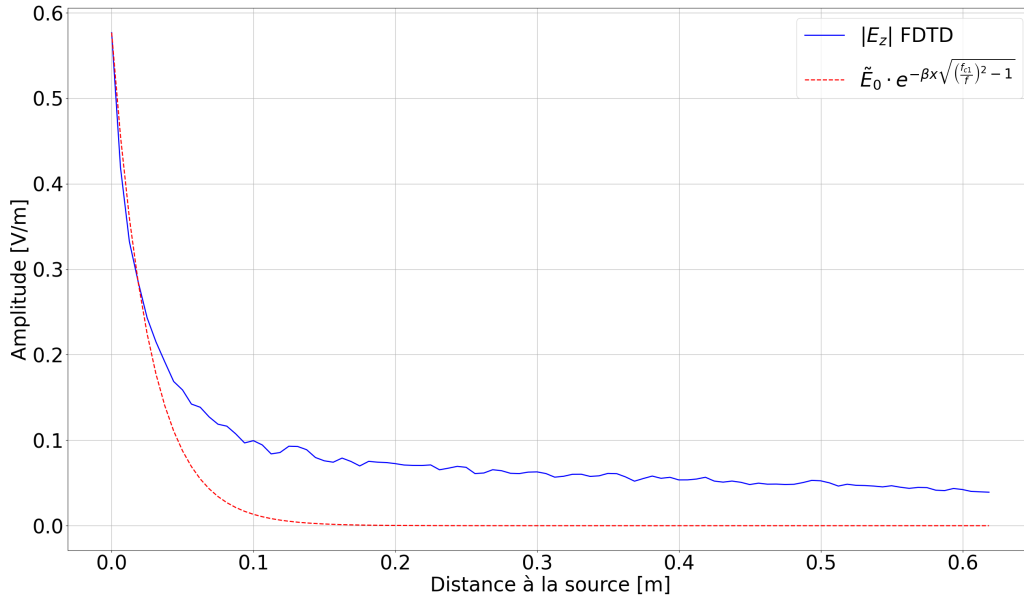


FIGURE 3.5 : Décroissance exponentielle de  $|E_z|$

Comme le mode TEM ne peut pas exister et que  $f < f_{c1}$ , tous les modes  $TE_n$  sont exponentiellement atténués. On observe que la courbe simulée ne correspond pas exactement à la courbe théorique, en particulier à grande distance. Cette différence provient du fait que l'expression théorique est incomplète : elle est valable uniquement pour une onde plane incidente selon un angle  $\theta_1$ . Or, la simulation utilise une ligne de courant, ce qui implique que tous les modes sont excités. Leur atténuation étant plus ou moins forte selon le mode, le mode  $TE_1$  domine à courte distance. Cependant, à mesure que la distance augmente, tous les modes atteignent une amplitude proche de zéro. Néanmoins, comme leur nombre est infini, leur somme ralentit la décroissance

globale, ce qui explique la divergence par rapport à l'expression théorique.

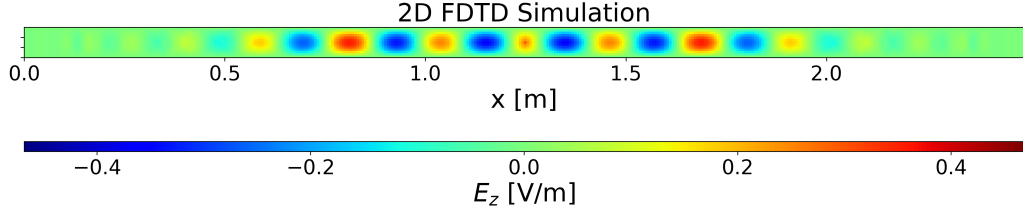


FIGURE 3.6 : Mode  $TE_1$  :  $f_{c1} < f < f_{c2}$

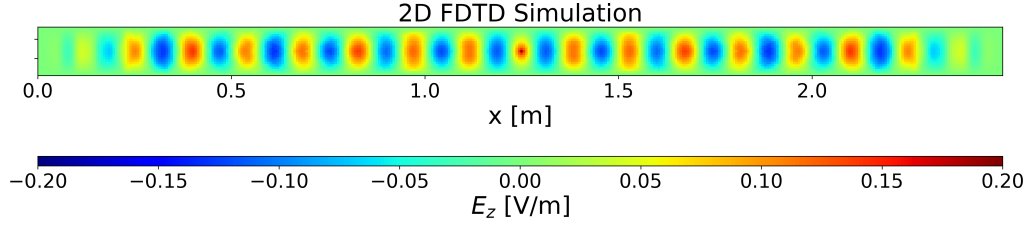


FIGURE 3.7 : Mode  $TE_2$  :  $f_{c2} < f < f_{c3}$

Les résultats des simulations sont bien cohérents avec la théorie, la longueur d'onde du motif associé au mode  $TE_2$  est inférieure à celle du mode  $TE_1$ . On peut calculer  $\lambda_{gn}$  avec la formule suivante :

$$\lambda_{gn} = \frac{2\pi}{\beta \sqrt{\left(1 - \frac{f_{cn}}{f}\right)^2}} \quad (3.5)$$

Pour  $f = 2.4GHz$ , on obtient  $\lambda_{g1} = 0.226m$  et  $\lambda_{g2} = 0.144m$ , ce qui correspond aux valeurs que l'on peut mesurer sur les figures. Les 2 figures sont prises au même échantillon temporel, on voit que le motif  $TE_2$  s'étend plus loin dans le domaine. La vitesse de phase de ce mode est donc plus élevée. Lorsque  $f > f_{c2}$ ,  $f > f_{c1}$  est aussi vérifiée. On a donc une combinaison des modes  $TE_1$  et  $TE_2$ . Ceci mène à un phénomène de battement, car les  $\lambda_{gn}$  sont différents.

Voici un exemple de code minimal pour la FDTD 1D dans les milieux :

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from matplotlib.animation import FuncAnimation
4 from scipy.constants import c
5 from scipy.constants import epsilon_0
6 from matplotlib.patches import Rectangle
7
8 # Parametres
9 f = 2.4e9 # Frequence (Hz)
10 Lambda = c / f # Longueur d'onde (m)
11 dx = Lambda / 20 # Pas spatial (m)
12 a = 2 # constante qui permet de changer le pas temporel (non utilisee finalement)
13 dt = dx / (a * c) # Pas temporel (s)
14 e_r = 4 # Permittivite relative du sol
15 M = 200 # Nombre de points spatiaux
16 Q = 800 # Nombre de pas de temps
17 x = np.linspace(0, (M - 1) * dx, M) # Grille spatiale
18 t = np.linspace(0, (Q - 1) * dt, Q) # Grille temporelle
19 # Grille de permittivite
20 wall_start = 3 * M // 5 # Debut du mur
21 wall_width = M // 4 # Largeur du mur
22 epsilon_r = np.ones((1, M))
23 epsilon_r[0, wall_start : wall_start + wall_width] = e_r
24 sigma = np.zeros((1, M)) # Grille de conductivite
25 sigma[0, wall_start : wall_start + wall_width] = 0.1 # Conductivite du sol
26
27 # Creation de Jz
28 omega = 2 * np.pi * f # Pulsation
29 J = np.zeros((Q, M)) # Densite de courant
30 source_position = (M // 2) # Position de la source (int)
31 J[:, source_position] = -np.sin(omega * t) # Source sinusoidale
32
33 # Creation des champs
34 E = np.zeros((1, M)) # Champ electrique, dernier echantillon (M-1)
35 B = np.zeros((1, M-1)) # Champ magnetique, dernier echantillon (M-2)
36 E_boundary = np.zeros((2, 2)) # Conditions aux limites pour E

```

```

37
38 # Initialisation de la figure
39 fig, ax = plt.subplots()
40 animated_source, = ax.plot([], []) # E
41 ax.set_xlim(0, (M - 1) * dx)
42 ax.set_ylim(-2, 2)
43 ax.set_xlabel("x [m]")
44 ax.set_ylabel("E [V/m]")
45 ax.set_title("FDTD 1D")
46
47 # Fonction de reinitialisation en fin d'animation
48 def reset():
49     global E, B, E_boundary
50     print("Reinitialisation de tous les vecteurs a zero...")
51     E[:] = 0 # Reinitialise le champ electrique
52     B[:] = 0 # Reinitialise le champ magnetique
53     E_boundary[:] = 0 # Reinitialise les conditions aux limites
54
55 # Fonction qui permet de calculer les champs et envoyer les donnees a l'animation
56 def update(frame):
57
58     if frame == 0: # Reinitialisation au debut de l'animation
59         reset()
60     # Conditions aux limites
61     if frame > 0:
62         E_boundary[0, 0] = E_boundary[1, 0] # Mise a jour de la condition limite gauche
63         E_boundary[1, 0] = E[0, 1] # Stocke la prochaine condition limite gauche
64         E_boundary[0, 1] = E_boundary[1, 1] # Mise a jour de la condition limite droite
65         E_boundary[1, 1] = E[0, M-2] # Stocke la prochaine condition limite droite
66     # Calculs FDTD
67     # NB : On ne sait pas faire de demi-indices pour B donc B[q + 1/2, m + 1/2] devient B[q, m]
68     # Comme on ne stocke pas la matrice, q n est pas utilise, on met simplement a jour un vecteur
        ligne
69     for m in range(1, M - 1): #1 compris, M-1 exclu
70         if sigma[0, m] == 0:
71             E[0, m] = E[0, m] + 1/a/epsilon_r[0,m] * (B[0, m]- B[0, m-1]) -
                (J[frame,m])/epsilon_r[0,m] #J est normalise par epsilon_r
72         else:
73             E[0, m] = (1- sigma[0,m]/epsilon_r[0,m])*E[0, m] + 1/a/epsilon_r[0,m] * (B[0, m]- B[0,
                m-1])
74
75     # # Application des conditions aux limites
76     if frame > a - 1 : # Passe la premiere iteration
77         E[0, 0] = E_boundary[0, 0]

```

```

78     E[0, M-1] = E_boundary[0,1]
79
80     if frame < Q-1: #Ne calcule pas la derniere ligne de B (pas utile pour trouver E)
81         for m in range(0, M - 1): #0 compris, M-1 exclu
82             B[0, m] = B[0,m] + 1/a *(E[0, m+1] - E[0, m])
83
84
85     animated_source.set_data(x, E[0, :]) # Met a jour le champ E
86     return animated_source, ax.title
87
88 # Create the animation
89 ani = FuncAnimation(fig, update, frames=Q, interval=15, blit=False, repeat=True)
90
91 # Save the animation
92
93 ani.save("1D_sine_source.mp4", fps=60)
94
95 # Show the animation
96 plt.show()

```

Listing 4.1 : Code FDTD 1D (Python)

Dans le cas 2D, la structure pour réaliser les animations est assez similaire, voici donc uniquement le calcul des champs :

```

1 def update(frame):
2     if frame == 0: # Reinitialisation au debut de l'animation
3         reset()
4     if frame > 0:
5         J[source_position_y, source_position_x] = -np.sin(omega * frame * dt) # Source sinusoidale
6         # Si la source est a l'interieur du conducteur parfait, on fixe E au lieu de J
7
8         # Calculs FDTD
9         for n in range(1, N - 1):
10             for m in range(1, M - 1): #1 compris, M-1 exclu
11                 if sigma[n, m] == -1: # -1 est un flag pour les conducteurs parfaits
12                     E[n, m] = 0
13                 elif sigma[n, m] == 0:
14                     E[n, m] = E[n, m] + 1/a/epsilon_r[n,m] * (-Bx[n, m] + Bx[n-1, m] + By[n , m] -
15                         By[n, m-1]) - (J[n,m])/epsilon_r[n,m] #Normalized J
16
17                 else:
18                     E[n, m] = (1- sigma[n,m]/epsilon_r[n,m])*E[n, m] + 1/a/epsilon_r[n,m] * (-Bx[n, m]
19                         + Bx[n-1, m] + By[n , m] - By[n, m-1])

```



```

19     if frame < Q-1: #Ne calcule pas la derniere ligne de B (pas utile pour trouver E)
20         for n in range(0, N - 1):
21             for m in range(1, M - 1): #0 compris, M-1 exclu
22                 Bx[n, m] = Bx[n,m] - 1/a *(E[n+1, m] - E[n, m])
23
24         for n in range(1, N - 1):
25             for m in range(0, M - 1): #0 compris, M-1 exclu
26                 By[n, m] = By[n,m] + 1/a *(E[n, m + 1] - E[n, m])
27
28     im.set_array(E) # Mise a jour du graphique 2D
29     return [im], ax.title

```

Listing 4.2 : Calcul des champs en 2D