



UNIVERSITÉ
LAVAL

Rapport de laboratoire 2

Transmission des ondes électromagnétiques

présenté à

M. Dominique Grenier

<i>matricule</i>	<i>nom</i>
910 055 897	Daniel Thibodeau
910 097 879	Francis Valois

Université Laval
16 octobre 2012

Chapitre 1

Laboratoire 2

1.1 Projet 1

L'objectif fondamental de ce laboratoire est de réaliser un programme informatique qui aura pour fonction de calculer numériquement les paramètres d'une ligne microruban. Le premier paramètre à trouver est l'impédance caractéristique de ligne : Z_0 . Le second paramètre à déterminer est la vitesse de propagation v_p .

On cherche à identifier les concepts de base théoriques utiles à ce laboratoire, soit :

- la version numérique de l'intégrale de Gauss nécessaire pour obtenir les capacitances \mathcal{C} et \mathcal{C}_0 ;
- l'obtention de Z_0 et v_p en mode quasi-TEM.

1.1.1 Passage de l'analytique au numérique

Les équations de Maxwell permettent, à partir d'une distribution de potentiel qui possède une géométrie simple et d'une distribution d'un champ électrique associé (\vec{E}), de déterminer la charge emmagasinée dans une distribution donnée. On observe que le champ électrique d'une distribution est lié au potentiel électrique de la manière suivante :

$$\vec{E}(x,y) = -\Delta V(x,y) \quad (1.1)$$

Aussi, on sait que la charge électrique dans un milieu donné s'exprime comme :

$$Q = \oint_S \epsilon \vec{E} \, dS \quad (1.2)$$

En combinant les deux expressions précédentes, on obtient :

$$Q = \oint_S -\epsilon \Delta V(x,y) \, dS \quad (1.3)$$

On note tout de suite que tout dépendant de l'orientation des surfaces respectives, l'intégrale deviendra rapidement infaisable à la main, d'ailleurs dans un laps de temps court. Il va de soit qu'il est possible d'exprimer ce calcul sous la forme de somme discrète (ce qui est en fait une intégrale) et qu'en minimisant les distances successives d'évaluation, on obtiendra une évaluation de plus en plus précise. On peut penser à une somme de Riemann classique, dans laquelle si on diminuait le pas entre chaque rectangle, on convergerait vers la définition

de l'intégrale. En reportant ce résultat dans le cas étudié, on obtient une version discrète de l'équation 1.3.

1.1.2 Détermination de \mathcal{C}

Afin de déterminer la capacité \mathcal{C} , on utilise le résultat qui nous donne une version discrète de la charge électrique de Q en chaque point. Sachant la charge électrique, il suffit de diviser par le potentiel en chaque point. Par la suite, on divise par le pas utilisé (il est intéressant de remarquer que dans le cas où il est uniforme et égal à 1, soit dans notre cas, cette étape n'affecte pas le résultat) et on obtient la capacité totale dans la géométrie étudiée.

1.1.3 Détermination de $\epsilon_{r_{interface}}$

Pour les points placés aux interfaces entre les deux diélectriques, on peut approximer la permittivité à l'interface ($\epsilon_{r_{interface}}$) par :

$$\epsilon_{r_{interface}} \approx \frac{(\epsilon_{r1} + \epsilon_{r2})}{2} \quad (1.4)$$

1.1.4 Détermination de la permittivité effective $\epsilon_{r_{eff}}$

Dans ce laboratoire, la détermination de la capacité \mathcal{C} est faite dans la section précédente. Ce faisant, on peut utiliser le résultat précédent et la relation suivante :

$$\epsilon_{r_{eff}} = \frac{\mathcal{C}}{\mathcal{C}_0} \quad (1.5)$$

Connaissant \mathcal{C} et \mathcal{C}_0 , trouver $\epsilon_{r_{eff}}$ devient trivial.

1.1.5 Détermination de Z_0 et de v_p

Comme on connaît \mathcal{C} et \mathcal{C}_0 des étapes précédentes, il suffit d'appliquer les relations suivantes pour trouver les deux paramètres demandés :

$$V_p \approx c \sqrt{\frac{\mathcal{C}_0}{\mathcal{C}}} \quad (1.6)$$

$$Z_0 \approx \frac{1}{c \sqrt{\mathcal{C}_0 \mathcal{C}}} \quad (1.7)$$

1.2 Projet 2

Il est important de noter que dans un soucis de lisibilité, le code Matlab de la fonction MDF n'est pas reporté dans le présent rapport. Celui-ci est toutefois disponible dans le rapport du laboratoire 1.

1.2.1 Algorithme

1.2.1.1 Fonction IGauss

```

1 function C = IGauss(V, er1, er2, d, w)
2     %Largeur et hauteur de la matrice avec correction
3     [n,m] = size(V);
4     n = n-1;
5     m = m-1;
6
7     %Milieu du systeme pour le calcul de Vab
8     if mod(m+1,2) == 0
9         halfM = (m+1)/2;
10    else
11        halfM = floor((m+1)/2);
12    end
13
14    %Definition des constantes
15    e0 = 8.854*(10^(-12));
16    Vab = V(n+1-d,halfM);
17
18    %Remplissage des permitivites pour simplifier les boucles
19    permitivityMatrix = zeros(n+1, m+1);
20    if d == 0;
21        permitivityMatrix(2:n, 2:m) = er2;
22    else
23        permitivityMatrix(2:n-d, 2:m) = er2;
24        permitivityMatrix(n+1-d, 2:m) = (er1+er2)/2;
25        permitivityMatrix(n+2-d:n, 2:m) = er1;
26    end
27
28    %Calcul du potentiel pour le pourtour du systeme
29    tourPotential = 0;
30
31    %Nord et Sud
32    for i=2:m
33        tourPotential = tourPotential + permitivityMatrix(n,i)*V(n,i);
34        tourPotential = tourPotential + permitivityMatrix(2,i)*V(2,i);
35    end
36
37    %Est et Ouest
38    for i=2:n

```

```

39         tourPotential = tourPotential + permitivityMatrix(i,2)*V(i,2);
40         tourPotential = tourPotential + permitivityMatrix(i,m)*V(i,m);
41     end
42
43     %Calcul de la capacite
44     C = e0/Vab*tourPotential;
45 end

```

1.2.1.2 Fonction MicroPar

```

1     function [Z_o, v_p] = MicroPar(m, n , er1, er2, d, w, tol)
2         %Declaration des constantes
3         c = 3*(10^8);
4
5         %Creation des deux matrices de potentiel. La premiere avec une
6         %permutivite dans le vide et l'autre avec les permutivites ...
7         %originales
8         V_vide = mdf(m,n,1,1,d,w,tol);
9         Vp_orig = mdf(m,n,er1,er2,d,w,tol);
10
11         %Calcul des deux differentes capacites
12         C0 = IGauss(V_vide,1,1,d,w);
13         C = IGauss(Vp_orig,er1,er2,d,w);
14
15         %Calcul des valeurs demandees selon les equation 1.6 et 1.7 ...
16         %dans le
17         %rapport
18         Z_o = 1/(c*sqrt(C0*C));
19         v_p = c*sqrt(C0/C);
20     end

```

1.2.2 Guide d'utilisation

Les fichiers nommés *mdf.m*, *IGauss.m* et *MicroPar.m* sont disponibles dans le répertoire remis sur pixel.

1. Copier les fichiers dans un répertoire connu du logiciel Matlab ;
2. Dans l'interpréteur, fixer les paramètres d'appel de la fonction selon la même nomenclature qu'utilisée dans l'énoncé de laboratoire. Pour se faire, utiliser une syntaxe du genre : $m=10$; etc.
 - Exemple : $[Z0, vp] = \text{MicroPar}(10,10, 1.10,4,3,1 * 10^{(-9)});$
3. Interpréter la ligne (appuyer sur entrer) ;
4. Le "workspace" de Matlab devrait contenir les deux réponses voulues.

1.3 Reproduction des exemples

Sachant que dans le précédent rapport, les deux matrices obtenues étaient en tout point semblables à celles fournies, il ne sert à rien de réeffectuer la comparaison.

1.3.1 Exemple 7.5

La matrice obtenu par la fonction mdF avec un seuil de tolérance de 10^{-9} par notre programme est la suivante :

$$V_0 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.07043930 & 0.12599118 & 0.14570741 & 0.12599118 & 0.07043930 & 0 \\ 0 & 0.15576603 & 0.28781803 & 0.33084727 & 0.28781803 & 0.15576603 & 0 \\ 0 & 0.26480681 & 0.53866761 & 0.60204562 & 0.53866761 & 0.26480681 & 0 \\ 0 & 0.36479358 & 1 & 1 & 1 & 0.36479358 & 0 \\ 0 & 0.19436750 & 0.41267643 & 0.45633821 & 0.41267643 & 0.19436750 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

En utilisant cette matrice dans la fonction IGauss ainsi que les paramètres suivants :

- $\epsilon_{r1} = \epsilon_{r2} = 1$;
- $\omega = 2$;
- $d = 2$.

nous donne la capacité suivante :

$$C_0 \approx 38.1549044906185[pF/m] \quad (1.8)$$

Cette valeur est très proche de la valeur fournie qui est la suivante :

$$C_{0_{prof}} \approx 38.15571464671[pF/m] \quad (1.9)$$

La différence entre les deux valeurs est de moins de 0.002% ce qui semble est un taux d'erreur très acceptable.

1.3.2 Exemple 7.6

La matrice obtenu avec un seuil de tolérance de 10^{-9} par notre programme est la suivante :

$$V = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.06983505 & 0.12534299 & 0.14509465 & 0.12534299 & 0.06983505 & 0 \\ 0 & 0.15399721 & 0.28644227 & 0.32969262 & 0.28644227 & 0.15399721 & 0 \\ 0 & 0.25971150 & 0.53673627 & 0.60079129 & 0.53673627 & 0.25971150 & 0 \\ 0 & 0.34811255 & 1 & 1 & 1 & 0.34811255 & 0 \\ 0 & 0.18987646 & 0.41139327 & 0.45569664 & 0.41139327 & 0.18987646 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

En utilisant cette matrice dans la fonction IGauss ainsi que les paramètres suivants :

- $\epsilon_{r1} = 10$
- $\epsilon_{r2} = 1$;
- $\omega = 2$;
- $d = 2$.

nous donne la capacité suivante :

$$C_0 \approx 227.651093653445[pF/m] \quad (1.10)$$

Cette valeur est très proche de la valeur fournie qui est la suivante :

$$C_{0_{prof}} \approx 227.6559274467[pF/m] \quad (1.11)$$

Encore une fois, la différence entre les deux valeurs est de moins de 0.002% ce qui tend à prouver que la fonction fait bien ce qui lui est demandé.

1.4 Programme selon la géométrie et les diélectriques