

# Exercices corrigés d'analyse tensorielle des réseaux

*De l'électronique à la multiphysique*

Olivier MAURICE

13 octobre 2008

# Table des matières

1. Rappel des principales méthodes topologiques.....	3
1.1 Méthode nodale.....	3
1.2 Méthode nodale différentielle.....	5
1.3 Méthode des mailles.....	7
1.4 Espace des moments.....	9
1.5 Méthode BLT généralisée.....	9
2. Filtres de premier ordre.....	11
2.1 Filtre RC.....	11
2.2 Filtre R-C: opérations sur les matrices d'ordre 2.....	14
2.3 Filtre RL et thermique.....	22
3. Un problème de perturbation par détection hyperfréquence dans un boîtier électronique	26
Circuit électrique.....	27
Circuit thermique.....	31
Vibrations.....	33
Connexions des différents réseaux.....	34
4. Méthode nodale différentielle.....	38

# 1. Rappel des principales méthodes topologiques

Nous rappelons rapidement les principes de base des techniques de description dans les différents espaces des nœuds, paires de nœuds, mailles et moments. Les lecteurs désirant revoir en détail ces techniques peuvent essayer de se procurer pour les relire les ouvrages donnés en référence<sup>1</sup>. Les applications données par la suite ne sont pas présentées sous forme d'énoncés – corrigés. Chaque enseignant pourra imaginer et adapter le niveau de difficulté des questions à ses auditeurs en profitant du détail de développement comme d'une correction prête à l'emploi. La difficulté des applications augmente au fur et à mesure de l'avancée dans le polycopie. Des précisions particulières sont données parfois sur certains thèmes ou le développement de certains programmes.

## 1.1 Méthode nodale

La méthode nodale est une application directe des lois de Kirchhoff. On définit les potentiels aux nœuds dans un vecteur  $\mathbf{vn}$ , ces potentiels étant référencés par rapport à un nœud désigné comme référence.

$$\mathbf{vn}_n = (\mathbf{vn}_1, \mathbf{vn}_2, \mathbf{vn}_3, \dots, \mathbf{vn}_N)$$

équation 1

Entre les tensions de branche  $\mathbf{vb}$  et le vecteur des potentiels de nœuds  $\mathbf{vn}$  on établit une incidence  $\mathbf{A}$  suivant:

$$\mathbf{vb}_i = \mathbf{A}_i^j \mathbf{vn}_j$$

équation 2

Les tensions de branches peuvent s'écrire, suivant la loi de Kirchhoff, en fonction des tensions développées aux bornes des charges  $\mathbf{v}$  et des sources  $\mathbf{vs}$ :

$$\mathbf{v}_i - \mathbf{vs}_i = \mathbf{A}_i^j \mathbf{vn}_j$$

équation 3

La branche de Kirchhoff, présentée figure 1, montre que les sources de courant  $\mathbf{is}$  sont opposées aux courants de branches  $\mathbf{i}$ . On a donc:

---

<sup>1</sup> PEIKARI, « Fundamentals of network analysis and synthesis ». Prentice hall series.  
PAPIN-KAUFMANN, « cours de calcul tensoriel ». Editions Albin Michel. MAURICE,  
« Compatibilité électromagnétique des systèmes complexes ». Edition Hermès sciences.

$$\mathbf{A}_q^m \mathbf{i}^q = -\mathbf{A}_q^m \mathbf{i}s^q$$

équation 4

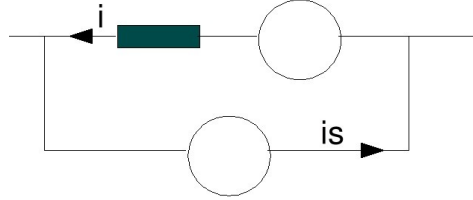


Figure 1

Les courants de branches  $\mathbf{i}$  peuvent être exprimés en fonction des tensions de branches  $\mathbf{v}$  par le biais de l'admittance  $\mathbf{Y}$  pour écrire:

$$\mathbf{A}_m^q \mathbf{Y}^{mp} \mathbf{v}_p = -\mathbf{A}_m^q \mathbf{i}s^m$$

équation 5

En utilisant l'équation 3 on obtient:

$$\mathbf{A}_m^q \mathbf{Y}^{mp} \mathbf{A}_f^p \mathbf{v}_{n_f} = -\mathbf{A}_m^q \mathbf{Y}^{mp} \mathbf{v}_{s_p} - \mathbf{A}_m^q \mathbf{i}s^m$$

équation 6

On reconnaît dans le premier produit triple la transformation de la métrique inverse de l'espace des branches à l'espace des nœuds où l'incidence est vue comme une connectivité avec  $\mathbf{Y}_n$  métrique inverse dans l'espace des nœuds:

$$\mathbf{Y}_n^{qm} = \mathbf{A}_i^q \mathbf{Y}^{ij} \mathbf{A}_j^m$$

équation 7

Cette relation permet de réécrire l'équation 6 sous la forme:

$$\mathbf{Y}_n^{qf} \mathbf{v}_{n_f} = -\mathbf{A}_m^q \mathbf{Y}^{mp} \mathbf{v}_{s_p} - \mathbf{A}_m^q \mathbf{i}s^m$$

équation 8

Si la métrique déduite de  $\mathbf{Y}_n$  existe, on peut multiplier les deux membres de l'équation 8 par cette métrique  $\mathbf{Z}_n$  pour obtenir:

$$\mathbf{v}_{n_q} = -\mathbf{Z}_{n_{qm}} \mathbf{A}_p^m \mathbf{Y}^{pf} \mathbf{v}_{s_f} - \mathbf{Z}_{n_{qm}} \mathbf{A}_p^m \mathbf{i}s^p$$

équation 9

Cette équation tensorielle avec les vecteurs  $\mathbf{i}s$  et  $\mathbf{v}_s$  des sources connus, représente un ensemble de  $N$  équations dont la solution est le vecteur des potentiels de nœuds  $\mathbf{V}_n$ . Connaissant les potentiels de nœuds on en déduit par l'équation 2 les différences de potentiels de branches. Les courants de branches s'obtiennent alors par la loi de Kirchhoff:

$$\mathbf{i}b^q = \mathbf{Y}^{qm} \left( \mathbf{v}b_m + \mathbf{v}s_m \right) + \mathbf{i}s^q$$

équation 10

La description nodale est celle utilisée (directement ou sous des formes similaires) dans les logiciels de simulation électronique type SPICE.

Son intérêt est de positionner la métrique et les inconnues au niveau de l'espace des nœuds. Elle peut donc présenter une description pertinente pour des problèmes liés aux charges sur les nœuds (électrostatique), comme interface vers les fonctions d'ondes de la mécanique quantique ou encore si l'on veut travailler sur les masses, etc.

On ne détaillera pas dans ce polycopié la méthode nodale modifiée. Elle présente certes un certain intérêt pour la programmation, mais l'organisation des matrices dans cette méthode ne répond pas à une rigueur d'identification des objets avec la topologie. Elle mélange métrique, connectivité, sources, etc. Cette non homogénéité des matrices me paraît particulièrement gênante pour la compréhension des formalismes topologiques et l'introduction à l'analyse tensorielle. Par ailleurs, même si cela est un peu plus compliqué à programmer, la méthode nodale « d'origine » est tout à fait programmable.

## 1.2 Méthode nodale différentielle

Encore appelée méthode des paires de nœuds elle est introduite par Kron. C'est une première introduction à l'usage des connectivités.

Dans cette méthode, chaque branche est une branche de Kirchhoff avec une charge et une source de courant. Mais cette fois au lieu d'établir l'incidence entre nœuds et branches, on va créer une connectivité entre des différences de potentiels de nœuds référencés à un nœud arbitrairement choisi comme référence, et les différences de potentiels de branches. Le nombre de paires de nœuds est fixé par la relation topologique:

$$\mathbf{P} = \mathbf{B} - \mathbf{N} + \mathbf{S}$$

équation 11

Où B est le nombre de branches, N le nombre de nœuds et S le nombre de sous réseaux. Chaque branche est du type de celle présentée figure 2: elle comporte une impédance et un générateur de courant.

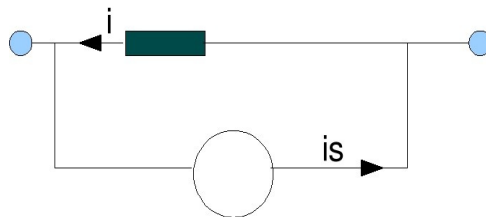


Figure 2

L'orientation de la branche est fixée par le courant de branche, L'insertion de la source de

courant pouvant être symboliquement ajouté à côté ou sur la branche. Par convention, la direction du courant source dans la charge est celle du courant de branche. Un graphe quelconque comportant 6 branches est présenté figure 3.

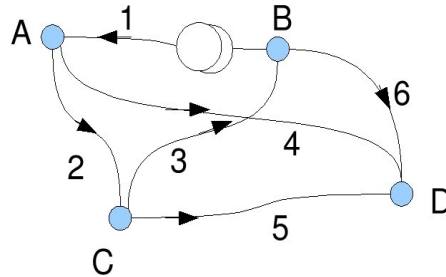


Figure 3

Chacune des branches de ce graphe est orientée. Seule la branche 1 comporte une source de courant. Chaque branche  $i$  comporte une admittance  $Y^i$ .

Comme dans la description nodale nous allons choisir un nœud comme référence. A partir de ce nœud on choisit des différences de potentiels vers tous les nœuds du graphe (figure 4).

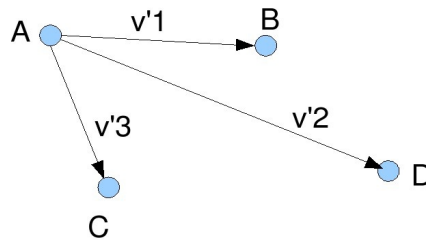


Figure 4

On note  $v_i$  le vecteur des branches. Entre les branches  $v_i$  et les différences de potentiels  $v'_i$  on peut établir une connectivité  $C$ . Les relations suivantes créent cette connectivité:

$$\begin{cases} v_1 = -v'_1 \\ v_2 = v'_3 \\ v_3 = v'_1 - v'_3 \\ v_4 = v'_2 \\ v_5 = v'_2 - v'_3 \\ v_6 = -v'_1 + v'_2 \end{cases} \Rightarrow C = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 1 & -1 \\ -1 & 1 & 0 \end{bmatrix}$$

équation 12

Le vecteur des sources  $\vec{I}$  est donné simplement par:  $[1 \ 0 \ 0 \ 0 \ 0 \ 0]$ . L'inverse de la métrique, en l'absence de couplages entre branches est donné par:

$$Y = \begin{pmatrix} Y^{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & Y^{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & Y^{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & Y^{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & Y^{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & Y^{66} \end{pmatrix}$$

équation 13

On calcule la transformation non holonomique de l'inverse de la métrique de l'espace des branches vers l'espace des différences de potentiels. On calcule:

$$Y^{qk} = C_{.i}^q \cdot Y^{ij} \cdot C_{.j}^k$$

équation 14

De la même façon on va pouvoir définir des courants de potentiels qui correspondent aux mouvements des charges accumulées sur les nœuds. On calcule:

$$I^k = C_j^k I^j$$

équation 15

L'équation du réseau est alors donnée par:

$$I^k = Y^{km} v'_m$$

équation 16

On résout les inconnues qui sont ici les différences de potentiels  $v'$ . Une fois ces ddp calculées on peut en déduire facilement les tensions de branches par l'équation des relations 12.

### 1.3 Méthode des mailles

Dans la méthode des mailles on définit une circulation de branches fermées par maille. La relation topologique de l'espace des mailles est:

$$M = B - N + R$$

équation 17

M est le nombre de mailles, B le nombre de branches, N le nombre de nœuds et R le nombre de sous-réseaux. La relation topologique doit être vérifiée à chaque nouveau problème. La figure 5 présente un graphe simple avec une description de branches et de mailles.

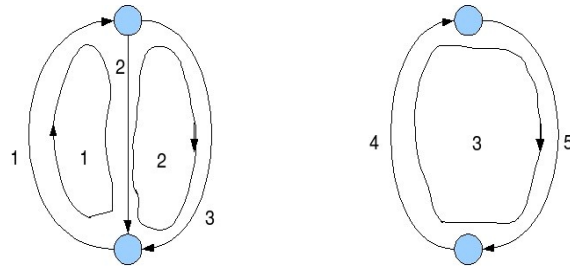


Figure 5

Dans l'espace des mailles, la connectivité est réalisée entre les flux de branches et ceux de mailles. Pour le graphe de la figure 5 nous avons:

$$\mathbf{I}^a = \mathbf{L}_\alpha^a \mathbf{I}^\alpha \Rightarrow \mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

équation 18

La métrique dans l'espace des branches est définie par les valeurs des impédances de chaque branche en termes diagonaux et par les impédances de couplage pour les termes extra diagonaux. La métrique dans l'espace des mailles est obtenue par l'opération:

$$\mathbf{Z}_{\alpha\beta} = \mathbf{L}_{\alpha.}^a \mathbf{Z}_{ab} \mathbf{L}_{.\beta}^b$$

équation 19

Pour chaque branche on peut définir une source. L'ensemble des sources dans l'espace des branches constitue un vecteur des sources (ou efforts)  $\mathbf{e}_a$ . Sa transformation dans l'espace des mailles est:

$$\mathbf{e}_\alpha = \mathbf{e}_a \mathbf{L}_\alpha^a$$

équation 20

Le système est représenté par le système d'équations tensorielles intégral-différentielle (au minimum):

$$\mathbf{e}_{\alpha.} = \mathbf{Z}_{\alpha\beta} \mathbf{I}^\beta$$

équation 21

On résout les courants de maille par inversion de la métrique.

## 1.4 Espace des moments

Lorsqu'un réseau est éloigné il ne reste de son image qu'un diagramme de rayonnement en champ lointain. On désire accéder à un niveau d'espace encore supérieur à l'espace des mailles pour obtenir ce niveau de description de champ lointain. La première opération consiste à relier les courants de mailles et les moments des mailles.

On connecte déjà les amplitudes par une relation de connectivité:

$$\mathbf{m}^y = \mathbf{S}_\alpha^y \mathbf{I}^\alpha$$

équation 22

S est une matrice de connectivité qui n'est pas une matrice d'Hadamard. Ses coefficients sont les surfaces des mailles. Pour rendre chaque moment vectoriel, on le multiplie par le vecteur normal à la maille (la notation de l'indice entre crochets indique que l'on se réfère à une composante particulière du vecteur).

$$\vec{\mathbf{m}}(\gamma) = \mathbf{m}^{[\gamma]} \vec{\mathbf{u}}_{[\gamma]}$$

équation 23

En réalisant le produit contracté des moments on obtient le moment total du réseau: on a atteint dans ce cas l'objectif de réduire le réseau à un vecteur de rayonnement unique.

$$\vec{\mathbf{M}} = \vec{\mathbf{u}}_y \mathbf{m}^y \quad \vec{\mathbf{u}}_y = \vec{\mathbf{u}}_x \mathbf{C}_y^x$$

équation 24

La matrice C réalise la connexion entre le vecteur moment à N dimensions et le vecteur des directions de l'espace à trois dimensions.

## 1.5 Méthode BLT généralisée

La méthode BLT (pour Baum, Liu, Tesche) a été généralisée par Jean-Philippe PARMANTIER et Gérard LABAUNE. Elle repose sur le concept d'ondes. Deux équations sont à la base de la méthode. La première équation donne le lien entre les ondes qui rentrent dans un noeud et les ondes qui sortent du noeud. La figure 5-2 montre un noeud où rentrent des ondes  $\Psi$  et d'où sortent des ondes  $\Phi$ .

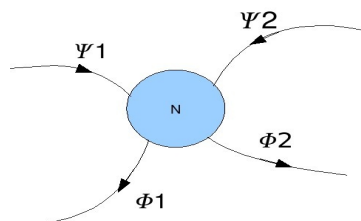


Figure 5-2

On ne détaille pas le processus qui relie les ondes entrantes aux sortantes. Elles sont toutes de même dimension (par exemple des ondes de tension, de courant, de puissance). Une matrice de diffraction – ou paramètres « S » - donnent les relations entre ces ondes:

$$\Phi_k = \Psi_m S_k^m$$

équation 24-2

Les noeuds échangent entre eux des énergies par le biais de tubes d'énergie conduite ou rayonnée. La figure 5-3 présente cet échange qui peut être symbolisé par un canal tel qu'une ligne modélisée par Branin.

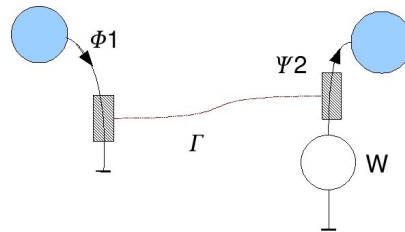


Figure 5-3

Sur un port en entrée on a la somme des sources W sur ce port et des ondes provenant d'autres noeuds.

$$\Psi_k = \Phi_m \Gamma_k^m + W_k$$

équation 24-3

En combinant les équations 24-2 et 24-3 on obtient:

$$\Phi_k \left( \mathbf{1} - S_m^k \Gamma_k^m \right) = W_m S_m^k$$

équation 24-4

C'est l'équation dite « BLT-généralisée ». Elle n'implique que des matrices et des vecteurs. On peut réécrire cette équation en choisissant de vecteurs naturels. Elle devient:

$$\left( \mathbf{1} - S_m^k \Gamma_k^m \right) \Phi^k = S_m^k W^m$$

équation 24-5

Cette écriture est quelque part plus élégante est naturelle positionnant les matrices à gauche dans les produits. De plus on peut définir les ondes de flux en les rattachant en tout points de l'espace et du temps à des vecteurs longitudinaux aux tubes de flux. Ces vecteurs eux-mêmes peuvent être définis dans un plan de polarisation. Cependant les ondes entrantes comme sortantes restent de même dimension, aucune métrique n'a lieu d'intervenir.

Dans la topologie BLT-G, on résout les ondes sortantes par inversion du terme entre accolades.

## 2. Filtres de premier ordre

Les filtres de premier ordre sont des circuits simples qui constituent des exemples idéaux pour appliquer une méthode nodale.

### 2.1 Filtre RC

La figure 6 présente le graphe d'un circuit RC. On y trouve deux nœuds et deux branches.

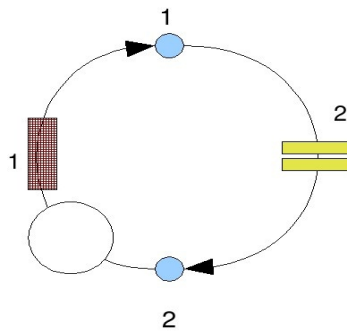


Figure 6

Les étapes de la méthode nodale sont les suivantes:

1. Choisir un nœud arbitraire comme masse et donner l'expression de la matrice d'incidence  $A$  entre potentiel de nœuds et différences de potentiels de branches;
2. trouver la matrice admittance des branches;
3. Utiliser  $A$  pour calculer l'admittance des nœuds:  $Y_n = A Y A^T$ ;
4. définir les sources de tensions et courants  $V_s$  et  $I_s$ ;
5. en déduire les potentiels de nœuds par:  $v_n = -Y_n^{-1} A Y v_s - Y_n^{-1} A I_s$ ;
6. Utiliser  $v_b = A^T v_n$  pour calculer les tensions et courants de branches.

En choisissant le nœud 2 comme nœud de masse nous pouvons écrire:

$$\begin{cases} v_1 = v n_1 \\ v_2 = -v n_1 \end{cases} \Rightarrow A^T = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

équation 25

La matrice admittance est donnée par:

$$Y = \begin{bmatrix} \frac{1}{R} & 0 \\ 0 & C \frac{d}{dt}(\cdot) \end{bmatrix}$$

équation 26

On calcule alors l'admittance du nœud 1:

$$Y_n = [1 \quad -1] \begin{bmatrix} \frac{1}{R} & 0 \\ 0 & C \frac{d}{dt}(\cdot) \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

équation 27

Le vecteur des sources vaut:  $v_s = [E \ 0]$ . On résout l'équation  $v_n = -Y_n^{-1} A Y v_s$ :

$$v_n = \frac{-1}{\left(\frac{1}{R} + C \partial_t(\cdot)\right)} [1 \quad -1] \begin{bmatrix} \frac{1}{R} & 0 \\ 0 & -C \partial_t(\cdot) \end{bmatrix} \begin{bmatrix} E \\ 0 \end{bmatrix}$$

équation 28

avec:  $\partial_t(\cdot) = \frac{d(\cdot)}{dt}$ .

$v_n$  est un vecteur réduit ici à un terme. En résolvant l'équation précédente on obtient:

$$v_n = \frac{-E}{(1 + RC \partial_t(\cdot))}$$

équation 29

Pour confirmer ce résultat nous pouvons résoudre l'équation de la maille:

$$E + R \frac{dq}{dt} + \frac{q}{C} = 0 \Rightarrow \frac{q}{C} [1 + RC \partial_t(\cdot)] = -E$$

équation 30

Comme  $v_n = \frac{q}{C}$ , l'équation 30 est bien équivalente à l'équation 29.

*Résolution temporelle de l'équation 29.*

Dans le domaine temporel, les inconnues doivent être positionnées dans le terme de gauche de l'équation. On veut résoudre en différences finies:

$$\left(1 + RC \partial_t (\cdot)\right) v_n = -E \Rightarrow v_n + RC \frac{(v_n - kn)}{\delta} = -E$$

équation 31

On a remplacé dans l'équation 30 la dérivée analytique par son équivalent en différences finies. Le terme kn représente la valeur passée de vn. En regroupant les termes on obtient:

$$v_n \left[1 + \frac{RC}{\delta}\right] - \frac{kn}{\delta} RC = -E \Rightarrow v_n = \frac{RC \frac{kn}{\delta} - E}{\left[1 + \frac{RC}{\delta}\right]}$$

équation 32

Dans la première partie de l'équation 32, le terme entre parenthèses et en facteur de vn est appelé wronksien. Nous le noterons W. Tous les autres termes regroupés forment ce que l'on nommera les termes sources, notés T. L'équation 32 peut donc se réécrire sous la forme:

$$W.v_n = T$$

équation 33

Que l'on résout en inversant W:  $v_n = W^{-1}.T$ . Cela revient à assimiler W avec Yn et T avec -AYvs-AIs avant de réexprimer ces matrices en différences finies (passer les termes en kn de W dans T).

On peut calculer la tension vn en tout instant par programmation. On donne ci-après le listing du programme écrit en python ainsi que la courbe obtenue, dessinée sous openoffice-calc (figure 7).

```
#####
# programme de calcul du circuit RC en méthode nodale #
# Auteur: Olivier MAURICE date: 2 Juillet 2008 #
#####
#
# importations
from math import *
```

```

#déclarations de variables
R=5.00
C=100E-12
d=10E-12
kn=[]
vn=[]

#initialisation
kn.append(0) #condition initiale en tension

#calcul
W=1+R*C/d
E=1
for t in range(0,500):
    vn.append((R*C/d*kn[t]-E)/W)
    kn.append(vn[t])

#sauvegarde du résultat
result_f=open('result_f.csv','w')
for t in range(0,500):
    result_f.write(str(t*d)+' '+str(-vn[t])+'\n') #-vn=v2
result_f.close()

```

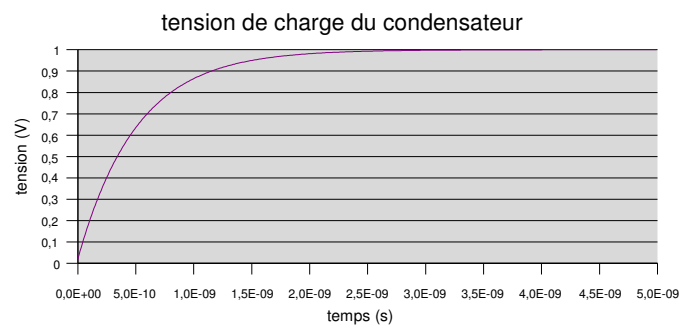


Figure 7

## 2.2 Filtre R-C: opérations sur les matrices d'ordre 2

La figure 8 présente un circuit constitué de quatre branches munies de résistances et d'une inductance. On cherche l'évolution de la tension aux bornes du condensateur.

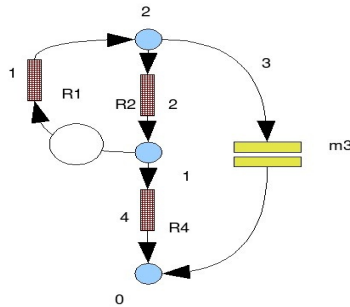


Figure 8

La connectivité et l'admittance sont données par (s est l'opérateur de Laplace):

$$A^T = \begin{bmatrix} -1 & 1 \\ 1 & -1 \\ 0 & -1 \\ 1 & 0 \end{bmatrix} \quad Y = \begin{bmatrix} R1 & 0 & 0 & 0 \\ 0 & R2 & 0 & 0 \\ 0 & 0 & s.m3 & 0 \\ 0 & 0 & 0 & R4 \end{bmatrix}$$

équation 34

L'étape suivante consiste à transformer l'admittance Y des branches dans l'espace des nœuds. Cette opération se répètera dans de nombreux cas puisqu'elle est à la base de l'usage des connectivités. On peut pour la réaliser soit la faire à la main, soit utiliser des outils de calculs formels comme Maxima, ou encore fabriquer un outil spécialisé dans cette opération. C'est l'objet du programme présenté ci-après. Ce programme n'est pas si simple que l'on pourrait le croire car il faut venir éliminer les caractères de fin de ligne, et transformer des multiplications de nombres en concaténations de mots. Il réalise en final ce qui est fait parmi les fonctions de Maxima. Mais pour ceux qui ne dispose pas de cet open source, ils peuvent ici faire la même opération naturellement sous tout environnement avec Python.

```
#####
# #
# Programme de transformation d'espace #
# #
# inputs: L connectivité #
# M matrice à transformer #
# ded dimensions de l'espace de départ #
# dea dimensions de l'espace d'arrivée #
# #
# O.Maurice 4 Juillet 2008 #
#####
#
#
```

```

# lecture des fichiers
ofiL=open('L.csv','r')
ofiM=open('M.csv','r')
ded=4
dea=2
L=[]
M=[]

tL=[]
tM=[]
for i in range(0,ded):
    tL=ofiL.readline()
    tM=ofiM.readline()
    L.append(tL.split(", "))
    M.append(tM.split(", "))

ofiL.close()
ofiM.close()

print 'L:\n'+str(L)
print 'M:\n'+str(M)

#retrait du caractère de fin de ligne
liste=[None]*dea
liste4=[None]*ded
LL=[]
MM=[]
for i in range(ded):
    LL.append(liste[:])
    MM.append(liste4[:])

for i in range(ded):
    for j in range(dea):
        LL[i][j]=L[i][j][0]
for i in range(ded):
    for j in range(ded):
        if j!=ded-1:
            MM[i][j]=M[i][j]
        if j==ded-1:
            MM[i][j]=M[i][j][:-1]

print 'LL:\n'+str(LL)
print 'MM:\n'+str(MM)

```

```

#transformation dans LL des - en -1
for i in range(ded):
    for j in range(dea):
        if LL[i][j]=='-':
            LL[i][j]='-1'

#affichage des tableaux finaux
print 'LL final:\n'+str(LL)
print 'MM final:\n'+str(MM)

# transposition de LL
liste3=[None]*ded
LLT=[]
for i in range(dea):
    LLT.append(liste3[:])

for i in range(dea):
    for j in range(ded):
        LLT[i][j]=LL[j][i]

print 'LLT:\n'+str(LLT)

#on réalise ici la tranformation
P=[]
liste2=[""]*ded
for i in range(dea):
    P.append(liste2[:])

for i in range(dea):
    for k in range(ded):
        for j in range(ded):
            if LLT[i][j]=='1':
                P[i][k]=P[i][k]+'+'+str(MM[j][k])
            if LLT[i][j]=='-1':
                P[i][k]=P[i][k]+'-'+str(MM[j][k])

print 'P: \n'+str(P)

#il faut remultiplier par L

P2=[]
liste5=[""]*dea
for i in range(dea):
    P2.append(liste5[:])

```

```

for i in range(dea):
    for k in range(dea):

        for j in range(ded):
            if LL[j][k]=='1':
                P2[i][k]=P2[i][k]+'('+str(P[i][j])+')'
            if LL[j][k]=='-1':
                P2[i][k]=P2[i][k]+'-'+str(P[i][j])+')'

print 'P2: \n'+str(P2)

#sauvegarde en .csv
ofiP=open('P.csv','w')
ofiP.write(str(P2))
ofiP.close()

```

Pour utiliser ce programme il faut écrire les matrices A et Y dans deux fichiers L.csv et M.csv respectivement. En fin de chaque ligne il faut faire <return> pour écrire un caractère de fin de ligne, y compris sur la dernière ligne. Le résultat est affiché avec des parenthèses pour les signes provenant de L. Ainsi,  $-(R1+0)$  donne  $+R1$  bien sûr. Mais c'est à l'utilisateur de traduire! On peut donc maintenant avec ce programme calculer  $Y_n$  dans l'espace des nœuds. On obtient:

$$Y_n = \begin{bmatrix} R1 + R2 + R4 & -R1 - R2 \\ -R1 - R2 & R1 + R2 + s.m3 \end{bmatrix}$$

équation 35

On veut résoudre l'équation:

$$Y_n^{uv} v_n = A_a^u Y^{ab} v_s$$

équation 36

Il n'y a pas de sources en courant. En développant les relations précédentes on obtient (le vecteur des sources étant réduit à la source E de la branche 1:  $V=[E \ 0 \ 0 \ 0]$ ):

$$\begin{cases} (R1 + R2 + R4) v_{n_1} - (R1 + R2) v_{n_2} = -R1E \\ -(R1 + R2) v_{n_1} + (R1 + R2 + s.m3) v_{n_2} = R1E \end{cases}$$

équation 37

A partir de ce système d'équations on peut établir leurs expressions en différences finies. On pose  $R11=R1+R2+R4$ ,  $R22=R1+R2$ . On obtient:

$$\begin{cases} R11 vn_1 - R22 vn_2 = -R1E \\ -R22 vn_1 + \left( R22 + \frac{m3}{\delta} \right) vn_2 = R1E + \frac{m3}{\delta} kn_2 \end{cases}$$

équation 38

On définit immédiatement:

$$W = \begin{bmatrix} R11 & -R22 \\ -R22 & R22 + \frac{m3}{\delta} \end{bmatrix} \quad T = \begin{bmatrix} -R1E \\ R1E + \frac{m3}{\delta} kn_2 \end{bmatrix}$$

équation 39

Les solutions s'obtiennent en inversant la matrice W à chaque pas de temps:

$$W^{ab} vn_b = T^a \Rightarrow vn_b = M_{ba} T^a \quad \text{avec:} \quad M_{ba} = [W^{ab}]^{-1}$$

équation 40

On aborde ici un premier cas d'inversion de matrice. Étant donné sa petite dimension, on peut utiliser l'inversion directe. Elle s'effectue en trois étapes: transposition de la matrice. Calcul du déterminant et des cofacteurs. Nous donnons la procédure pour une matrice générique « abcd »:

$$M = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \rightarrow M^t = \begin{bmatrix} a & c \\ b & d \end{bmatrix} \rightarrow D = ad - cb \rightarrow M^{-1} = \begin{bmatrix} \frac{d}{D} & \frac{-b}{D} \\ \frac{-c}{D} & \frac{a}{D} \end{bmatrix}$$

équation 41

Un problème de CEM peut être de calculer la tension développée dans la résistance R4 qui peut être vue comme une impédance de masse. Il suffit pour cela, vu notre choix de référence, de sortir la tension au nœud 1. On obtient la courbe suivante:

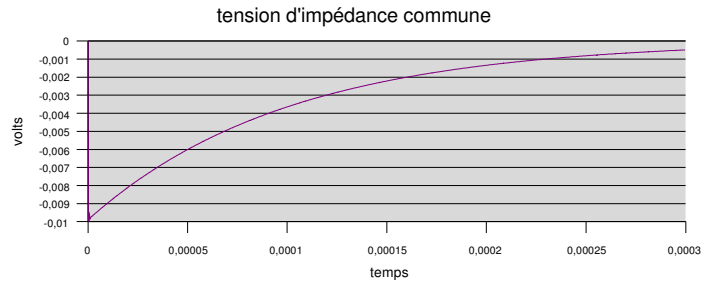


Figure 9

Le programme de calcul est donné ci-après. Notez l'organisation du programme qui suit les préconisations d'usage<sup>2</sup>.

```
#####
#
# Programme de calcul du circuit RC: m #
#
# Auteur: Olivier MAURICE, date: 9-6-08 #
#
#####
#

#definitions locales de fonctions
def det22(ww):
    #det22 calcule le determinant d'une matrice 2x2
    det=0.0
    det=ww[0][0]*ww[1][1]-ww[0][1]*ww[1][0]
    return det

def transpose22(mu):
    #transpose calcule la transposee d'une matrice 2x2
    wu=[[0.0,0.0],[0.0,0.0]]
    for i in range(2):
        for j in range(2):
            wu[i][j]=mu[j][i]
    return wu

def mp(A,B):
    C=[[0.0],[0.0]]
    #mp calcule un produit matriciel entre matrice 2x2 et une matrice 1
    for i in range(2):
        for j in range(2):
            C[i][0]=A[i][j]*B[j][0]+C[i][0]

    return C

def transposecl(X):
    Y=[0.0,0.0]
    for i in range(2):
```

<sup>2</sup> Programmer avec python. Gérard Swinnen, chez O'Reilly

```

        Y[i]=X[i][0]
    return Y

#valeurs des composants
R1=10.0
R2=1000.0
R4=0.001
m3=1E-7
E=1.0

#precalculs
R11=R1+R2+R4
R22=R1+R2

#pas de temps
d=3E-7

#nombre de pas
nbre=1000

#initialisation
row=[0.0]*2
kn=[]
vn=[]
#-----
for i in range(nbre):
    kn.append(row)

W=[[R11,-R22],[-R22,R22+m3/d]]
print 'W: \n'+str(W)
#inversion de W
invw=[[0.0,0.0],[0.0,0.0]]

tinvw=transpose22(W)
print 'tinvw: \n'+str(tinvw)

detinvw=det22(tinvw)
print 'detinvw: \n'+str(detinvw)

invw=[[ (R22+m3/d)/detinvw,R22/detinvw],[R22/detinvw,R11/detinvw]]
print 'invw: \n'+str(invw)

#boucle de calcul principal
for s in range(1,nbre):
    T=[[-R1*E],[R1*E+m3/d*(kn[s-1][1])]] #kn[s][1] car en python les
tableaux sont indic partir de 0: k(t,2) est donc kn[t][1]
    vn=mp(invw,T)
    #print 'vn: \n'+str(vn)
    kn[s]=transposecl(vn)
    #print 'kn: \n'+str(kn)

ofil=open('res_cal.csv','w')
for i in range(nbre):
    ofil.write(str(i*d)+'\n'+str(kn[i][0])+'\n')

```

```

ofil.close()
print 'fin de calcul'

```

## 2.3 Filtre RL et thermique

Dans cet exercice on considère le circuit présenté figure 10. Ce circuit est constitué d'une source de tension, de deux résistances R et S et d'une inductance L. La résistance R dépend de la température. On considère pour la résistance R une loi de la forme:

$$R(T) = R_0 + aT$$

équation 42

La température est déterminée par l'effet joule lié au courant circulant dans la résistance, et à la résistance thermique liée aux échanges de chaleur entre la résistance et l'air ambiant. Il faut regarder le réseau sous un autre jour, dans le domaine de l'infrarouge pour lui associer un nouveau réseau thermique.

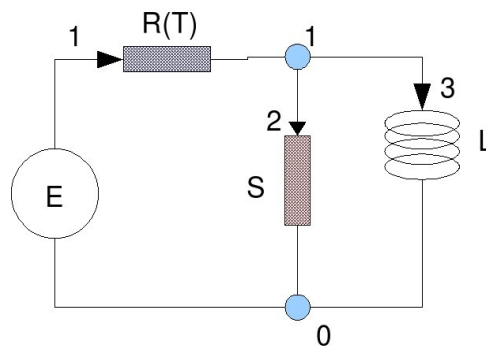


Figure 10

Le réseau thermique est présenté figure 11. Sa source est la puissance joule développée dans la résistance que nous considérons: R.

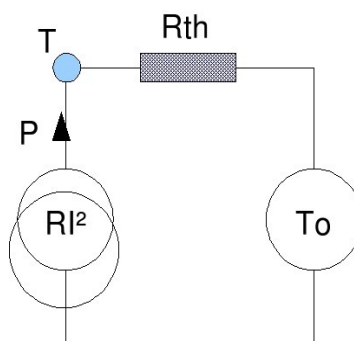


Figure 11

Equations du réseau électrique: la matrice d'incidence entre potentiels de noeuds et différences de potentiels peut être construite directement, sans passer par sa transposition

en donnant les relations de l'espace des noeuds vers l'espace des branches. Il en est ainsi que les matrices de connexions sont obtenues en positionnant en colonne l'espace de hiérarchie supérieure, et en ligne celui dont est issu la transformation. Ici l'espace de hiérarchie topologique supérieure est l'espace des branches. On en déduit la matrice d'incidence:

$$\mathbf{A} = \begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \end{bmatrix}$$

équation 43

La méthode nodale établit formellement l'égalité entre les courants de noeuds et la transformation des courants de branches. C'est comme ça qu'il faut comprendre cette équation. On peut s'aider de signes supplémentaires pour augmenter la compréhension du mécanisme. On écrit:

$$\overset{\circ}{\mathbf{Y}}^{uv} \left( \overset{\circ}{\mathbf{v}}_v \right) = \mathbf{A}_a^u \left[ \bar{\mathbf{Y}}^{ab} \left( \bar{\mathbf{E}}_b \right) \right] \quad \text{équation 44}$$

On a placé un petit cercle au-dessus des grandeurs de l'espace des noeuds et une petite barre au-dessus des grandeurs de l'espace des branches.

On exprimera chaque membre de l'équation 44 à titre d'exercice en partant de l'expression du tenseur des admittances dans l'espace des branches:

$$\bar{\mathbf{Y}}^{ab} = \begin{bmatrix} \frac{1}{\mathbf{R}(\mathbf{T})} & 0 & 0 \\ 0 & \frac{1}{\mathbf{S}} & 0 \\ 0 & 0 & \frac{1}{\mathbf{L}_t} \int dt (.) \end{bmatrix}$$

équation 45

On remarque que sous une description d'espace des noeuds, l'inductance devient un opérateur d'intégration et le condensateur un opérateur de dérivation. C'est l'inverse de ce que nous serons habitués à écrire dans un espace de description des mailles.

En forçant le potentiel du noeud 0 à 0 l'équation 44 se réduit à une seule équation. Le circuit thermique est décrit par une équation simple. Au final on obtient un système de deux équations à deux inconnues avec deux paramètres eux-mêmes décrits par un autre système:



```

RES=0.0
#print "Tab: "+str(Tab)
for i in range(I):
    RES=RES+Tab[i]
return RES

#variables générale
To=300.0; #temperature ambiante en kelvin
a=1.0E-3 #coefficient de température de R(T)
Ro=10.0
E=1.0
Rth=1.0E3
S=100.0
L=100.0E-9
V1=[0.0]
K1=[0.0]
Tmem=[To]
#pas de temps
d=0.1E-9

#initialisation
T=To
for t in range(1,1000):
    RT=Ro*(1+a*T)
    W=(1/RT+1/S+d/L)
    u=E/RT-d/L*sum(t,K1)
    V1.append(u/W)
    P=(V1[t]-E)**2/RT
    T=To+Rth*P
    Tmem.append(T)
    K1.append(V1[t])

result_L=open('result_L2.csv','w')
for t in range(1000):
    result_L.write(str(t*d)+' '+str(V1[t])+' '+str(Tmem[t])+'\n')
result_L.close()

```

Le résultat obtenu où l'on trace sur un même graphe les fonctions de potentiel de noeud et température est donné figure 12. On observe bien une décroissance de la tension de noeud à mesure que l'inductance se charge et une stabilisation de la température quand le courant devient constant.

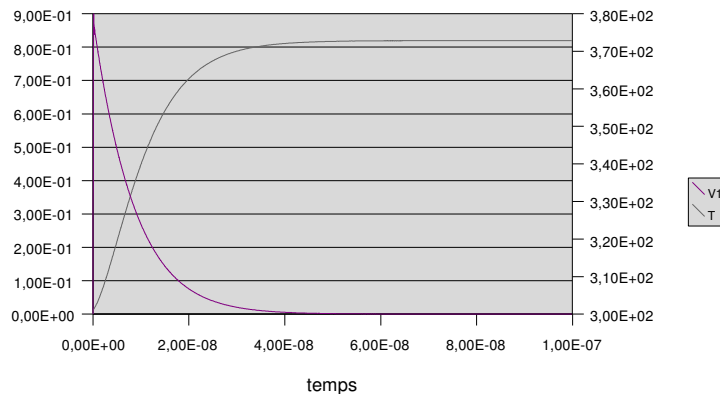


Figure 12

La constante de temps est en  $L/R$  soit 30ns typiquement.

### 3. Un problème de perturbation par détection hyperfréquence dans un boîtier électronique

C'est la deuxième version de proposition et résolution de ce problème finalement assez complexe, entre autre ramener à un cas simple le problème d'origine tout en gardant un comportement similaire ne s'est pas avéré si aisé. Mais finalement cet exercice est riche d'enseignement et met en évidence les difficultés de raccord des pas de temps, échantillonnages en fréquence et durée de calcul et taille mémoire. Une version précédente trop brouillonne n'a pas abouti, et j'ai préféré reprendre entièrement le problème pour le présenter d'une façon beaucoup plus claire. Par ailleurs vu la complexité et le nombre d'opérations effectuées sur des matrices de dimensions notables dans ce problème, nous le résoudrons en utilisant SCILAB. La transposition à python restant faisable avec un peu de rigueur et l'usage du module numpy.

On considère un boîtier contenant un circuit électronique analogique amplifiant le signal d'un capteur. L'amplificateur a un gain qui dépend de la température. Ce boîtier comporte un capot vissé mais subit des vibrations. Ces vibrations modifient l'ouverture résiduelle entre deux visse du capot et la transmission d'un champ externe vers l'intérieur du boîtier. Le couplage du champ vers la liaison capteur engendre une perturbation dès que la température est assez élevée.

La première étape consiste à construire les trois graphes correspondant aux trois physiques étudiés.

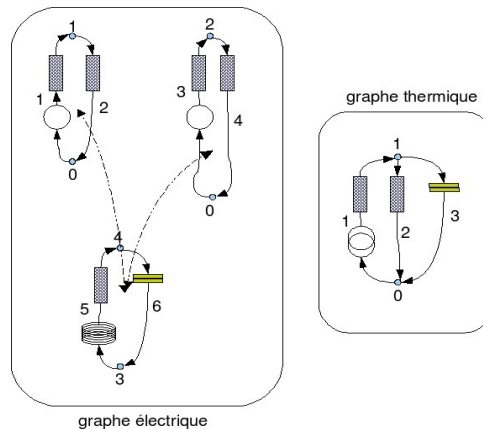


Figure 13

La figure 13 présente ces trois graphes. Le graphe lié à la vibration n'est pas utile ici, portant sur uniquement 2 noeuds. Les trois graphes sont liés, mais pas toujours par des relations bijectives. Ainsi les vibrations modifient la taille de la fente du boîtier, mais rien de l'électronique ou du thermique ne modifie les vibrations. Dans un premier temps il faut écrire les équations de chaque physique, puis se poser la question des éléments liés.

### ***Circuit électrique***

On doit définir la connectivité entre noeuds et branches. Tant qu'à faire on peut écrire directement cette connectivité dans l'éditeur de SCILAB (figure 14).

```
A=[1 -1 0 0 0 0
    0 0 1 -1 0 0
    0 0 0 0 -1 1
    0 0 0 0 1 -1];
```

figure 14

La construction de la métrique se fait en deux temps: c'est un inconvénient des descriptions dans l'espace des noeuds. On écrit d'abord la métrique ne contenant que les énergies électriques et de pertes. SCILAB n'étant pas un outil orienté calcul symbolique, on va donner des valeurs aux composants pour générer facilement la matrice Y. On définit tout d'abord les valeurs des composants, puis on définit un vecteur ligne que l'on diagonalise pour écrire la métrique limitée aux éléments R et C. La syntaxe sous SCILAB est très simple. La figure 14 présente un exemple simple de diagonalisation d'un vecteur de 3 composantes.

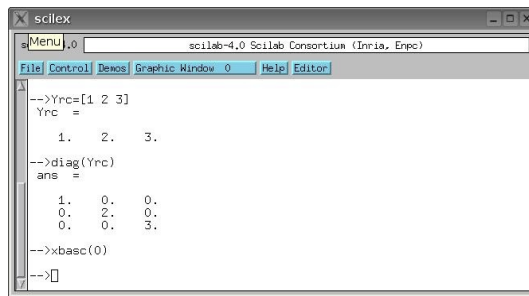


Figure 14

On résout cet exercice dans le domaine des fréquences. Le calcul de la métrique va donc se faire pour chaque valeur de fréquence.

Pour la métrique inverse des énergies magnétiques  $Y_m$  c'est moins évident. Il y a l'inductance de la fente<sup>3</sup> mais aussi les couplages entre la fente et la liaison du capteur. Pour construire la métrique inverse  $Y_m$ , il faut d'abord construire la métrique directe et l'inverser. Or les couplages de types magnétiques se définissent bien au niveau des branches. On peut en déduire la métrique inverse cherchée dans l'espace des branches par:

$$Y_m^{ba} = \left( Z_{ab} \right)^{-1}$$

équation 50

SCILAB se charge alors d'inverser cette matrice puis de l'ajouter à la partie RC:

```

bYv=[G1 G2 G3 G4 G5 C6*s]; //Y des RC dans l'espace des branches
bYrc=diag(bYv); //diagonalisation pour avoir le Yrc final, toujours dans l'espace des branches
bZm=[0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 L55*s 0
0 0 0 0 L66*s]; //impedance des energies magnetiques dans l'espace des branches
bYm=pinv(bZm);
bY=bYrc+bYm; //ici on calcule l'inverse de la métrique dans l'espace des branches
nY=A*bY*A';

```

Cette matrice est ici exprimée dans la forme de Laplace,  $s$  étant l'opérateur de Laplace. Pour l'instant les composants intervenant dans cette métrique ont des valeurs constantes. On désire tout d'abord mettre au point l'analyse purement électrique de ce réseau. Par ailleurs, ce type de matrice est souvent singulier. On utilise de préférence la pseudo-inversion que l'inversion directe (on aurait pu bien sûr inverser directement les éléments, mais cette procédure est plus générale et automatique). La connaissance de la métrique dans l'espace des branches implique sa connaissance dans l'espace des noeuds. On réalise l'opération (savoir la retrouver est un bon exercice de base):

$$\overset{\circ}{Y}^{ab} = A_{\mu}^b \bar{Y}^{\mu\nu} A_{\nu}^a \quad \text{équation 51}$$

On dispose de tous les éléments dans notre programme pour faire ce calcul. Ce calcul, si  $nY$

<sup>3</sup> voir « microondes, tome1 », de Badoual.

est le tableau résultat dans l'espace des noeuds et  $\mathbf{bY}$  la métrique inverse dans l'espace des branches, s'écrit simplement:

$$\mathbf{nY} = \mathbf{A} * \mathbf{bY} * \mathbf{A}' \quad \text{équation 52}$$

$\mathbf{A}'$  étant la matrice transposée de  $\mathbf{A}$ .

Il reste à définir le vecteur des sources. La source fonctionnelle est  $v_1$ , signal transmis ensuite avec amplification en  $v_3$ . Le champ externe se couple avec la fente en  $v_6$ , source non représentée sur la branche 6. On étudie la réponse du système pour une source constante en fréquence (c'est le cas dans le cadre des tests standards). Entre le champ externe et la tension développée dans la fente du boîtier on peut considérer (en supposant que la polarisation du champ est adéquat) une relation d'intégration simple:

$$\mathbf{V} = \int_{h_{\text{fente}}} \vec{\mathbf{E}} \cdot d\vec{\mathbf{h}} \quad \text{équation 53}$$

Le vecteur des sources s'écrit au départ:

$$\mathbf{bV} = \begin{bmatrix} 0.1 & 0 & 1 & 0 & 0 & \mathbf{E}_0 * \mathbf{h} \end{bmatrix} \quad \text{équation 54}$$

Ensuite, il faut lui ajouter les termes provenant du couplage avec la fente, qui s'exprime sous la forme de  $jM\omega i(\text{fente})$ .  $h$  est la hauteur de fente,  $E_0$  le champ incident au niveau de la fente. Le gain en transmission est de 10.

Les sources doivent être définies en dehors de la boucle temporelle au départ pour être initialisée au premier calcul. Leur mise à jour s'effectue ensuite en fin de boucle en fréquence, nous reviendrons sur ce point ultérieurement. Nous n'avons pas de sources en courant. On peut résoudre à ce niveau l'équation du réseau électrique donnée par:

$$\overset{\circ}{\mathbf{V}}_{\omega} = - \left( \overset{\circ}{\mathbf{Y}}^{u\omega} \right)^{-1} \cdot \left\{ \mathbf{A}_a^u \bar{\mathbf{Y}}^{ab} \bar{\mathbf{V}}_b \right\} \quad \text{équation 55}$$

On réarrange un peu les notations et valeurs dans notre programme, on spécifie une seule valeur de fréquence pour obtenir comme listing pour la boucle en fréquence:

```

//for f=[1:1000]
f=1;
s=%i*2*%pi*f*fo;
//metriques
bYv=[G1 G2 G3 G4 G5 C6*s]; //Y des RC dans l'espace des branches
bYrc=diag(bYv); //diagonalisation pour avoir le Yrc final, toujours dans l'espace des branches
bZm=[0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 L55*s 0
0 0 0 0 L66*s]; //impedance des energies magnetiques dans l'espace des branches
bYm=pinv(bZm);
bY=bYrc+bYm; //ici on calcule l'inverse de la métrique dans l'espace des branches
nY=A*bY*A';

nV=-pinv(nY)*(A*bY*bV');
If=[If; (nV(4)-nV(3))/(C6*s)];

bV=[0.01 -M26*s*If(f) 1-M36*s*If(f) 0 0 -E0*h]; //mise à jour des sources

//end,

```

Dès lors que le programme fonctionne pour une fréquence, on peut espérer qu'il fonctionne pour plusieurs fréquences. Il faut créer un tableau qui mémorise toutes les valeurs de tension de noeuds pour toutes les fréquences. On peut ensuite tracer l'évolution de la tension de noeud  $v_2$ , qui est la tension de sortie puisque l'on se réfère à une masse forçant un zéro volt sur le noeud à l'autre extrémité de la branche 4. Pour sortir le résultat avec la précision voulue  $e$ , on utilise la fonction `round` de SCILAB qui arrondi à l'entier le plus proche en faisant: `round(e*x)/e`. On se limite ici à  $10^{-1}$ . Le listing finalement retenu est le suivant:

```
// initialisations
clear;
xbasc();

//valeurs des composants
G1=1E-3;
G2=100;
G3=100;
G4=100;
G5=1E-3;
C6=100E-12;
M26=1E-11;
M36=1E-11;
L55=5E-9;
L66=5E-9;
//autres paramètres
h=1E-3;
E0=1E3;

//connectivites
A=[1 -1 0 0 0 0
0 0 1 -1 0 0
0 0 0 0 -1 1
0 0 0 0 1 -1];

//initialisation des sources
bV=[0.01 0 1 0 0 -E0*h];

//tableaux et constantes
K=[]; //tableau de memorisation des valeurs de tensions de noeuds
If=[];

fo=1E6;

//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for f=[1:1000]
//f=1;
s=%i*2*%pi*f*fo;
//metriques
bYv=[G1 G2 G3 G4 G5 C6*s]; //Y des RC dans l'espace des branches
bYrc=diag(bYv); //diagonalisation pour avoir le Yrc final, toujours dans l'espace des branches
bZm=[0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 L55*s 0
0 0 0 0 L66*s]; //impedance des energies magnetiques dans l'espace des branches
bYm=pinv(bZm);
bY=bYrc+bYm; //ici on calcule l'inverse de la métrique dans l'espace des branches
nY=A*bY*A';

nV=-pinv(nY)*(A*bY*bV');
K=[K;nV'];
If=[If;(nV(4)-nV(3))/(C6*s)];

bV=[0.01 -M26*s*If(f) 1-M36*s*If(f) 0 0 -E0*h];

end,
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
//traces
subplot(2,1,1); plot2d([1:length(K(:,2))]*fo,round(10*abs(K(:,2)))/10,style=9,logflag="nn"); xtitle("Tension de sortie");
subplot(2,1,2); plot2d([1:length(K(:,2))]*fo,round(10*abs(If))/10,style=9,logflag="nn"); xtitle("courant de fente");
```

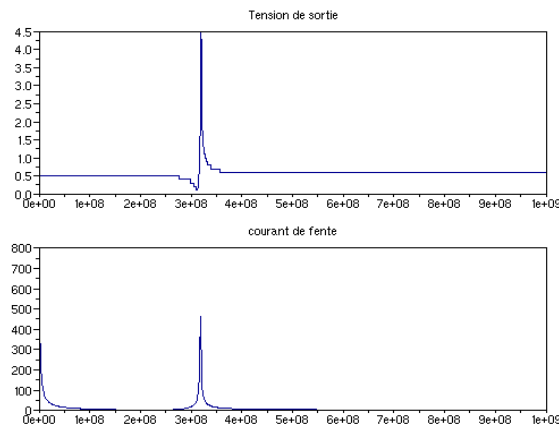


Figure 15

Testons ce premier programme, où le condensateur C6 est fixe et le gain est indépendant de la température. On obtient les traces suivantes présentées figure 15.

Le courant de fente atteint une valeur très élevée à une résonance vers 320 MHz. A cette même fréquence il induit sur le signal capteur un parasite important, reporté sur la sortie de l'amplificateur avec ici un gain de 10.

Concernant la mise à jour des sources en fin de boucle. La valeur de courant prise en compte pour la fréquence  $f+1$  est celle calculée à la fréquence  $f$ . Il y a donc une erreur commise en contrepartie de disposer de matrices mieux conditionnées. On peut admettre que l'échantillonnage est suffisant pour que la variation des courants reste faible entre deux points d'échantillonnage. Cette condition réalisée, l'erreur dépend de la pente locale de la fonction. On pourra étudier ces questions plus précisément ensuite, ce n'est pas l'objet de ces exercices. Les personnes impatientes de comprendre les notions de conditionnement de matrices, etc., peuvent lire le « Nougier<sup>4</sup> », très concis et précis sur ce sujet.

Abordons maintenant l'influence de la température.

## ***Circuit thermique***

Le circuit thermique se résout lui dans le domaine du temps. Bien qu'extrêmement simple, on résout ce circuit par la méthode des noeuds, pour l'entraînement!

On force le deuxième noeud comme référence. La connectivité se réduit alors à une seule ligne:

$$\mathbf{A} = \begin{bmatrix} 1 & -1 & -1 \end{bmatrix} \text{ équation56}$$

L'inverse de la métrique se détermine facilement:

---

4 Méthodes de calcul numérique de J.P.Nougier, chez Masson.

$$\bar{Y}^{ab} = \begin{bmatrix} G1 & 0 & 0 \\ 0 & G2 & 0 \\ 0 & 0 & C3.s \end{bmatrix} \quad \text{équation 57}$$

Le calcul de l'inverse de la métrique dans l'espace des noeuds se détermine tout aussi simplement:

$$A_a^\mu \bar{Y}^{ab} A_b^\nu = \dot{Y}^{\mu\nu} = [G1 + G2 + C3.s]$$

équation 58

L'équation temporelle à résoudre s'écrit dans un premier temps dans la forme de Laplace:

$$(G1 + G2 + C3.s) \dot{Q}_1 = \dot{P}^1 \quad \text{avec:} \quad A_b^\mu \bar{P}^b = \dot{P}^1$$

équation 59

Q est l'accroissement de température, P la puissance issue d'une branche du réseau électrique. On résout ce système en différences finies avec un pas de temps d. On écrit:

$$\left[ G1 + G2 + \frac{C3}{d} \right] \dot{Q}_1 = P^1 + \frac{C3}{d} K(\dot{Q}_1, t)$$

équation 60

Comme précédemment, on programme ce circuit d'abord indépendamment. On force la source à 1 Watt (P1=1). Le listing du programme est le suivant:

```
clear;
xbase;

//composants
R1=100;G1=1/R1;
R2=1000;G2=1/R2;
C3=1;
To=293;

d=1;

Q1=0;

qK=zeros(1);
P1=1;
sorti=[];

for t=[2:1000]
W=G1+G2+C3/d;
Q1=1/W*(P1+C3/d*qK(t-1));
qK=[qK;Q1];

end,
plot2d([1:1000]*d,qK+To,style=9,logflag="nn");xtitle("Evolution de la temperature [K]);
```

Comme Q représente l'incrément de température, on lui ajoute la température ambiante. La courbe d'évolution de la température obtenue est donnée figure 16.

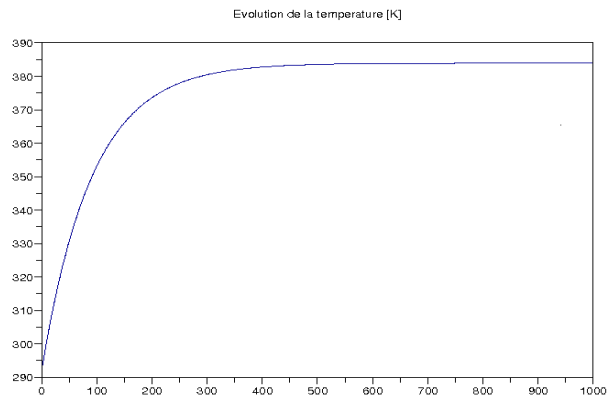


Figure 16

Nous aborderons ultérieurement les connexions entre les réseaux des différentes physiques.

## Vibrations

Le noeud 3 est supposé fixe. La vibration est traduite sur les coordonnées du point 4. Pour une élasticité définie, le déplacement relatif du noeud 4 suit la vibration. Pour un noeud unique, il n'y a pas lieu de décrire un réseau. On donne simplement la loi de déplacement du noeud 4 en hauteur (direction z) relativement à la position du noeud 3:

$$n_4(z) = 2 \cdot 10^{-3} + 10^{-3} \sin\left(2\pi \frac{t}{10^{-3}}\right)$$

équation 61

La figure 17 montre la vibration obtenue pour certains réglages des paramètres.

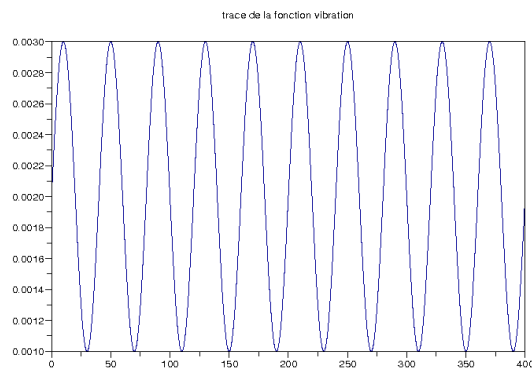


Figure 17

Le script programmé est le suivant:

```

clear;
d=0.1;

n=[];
for t=[1:4000]
s=t*d;
n=[n;2E-3+1E-3*sin(2*%pi*s/40)];
end,
plot2d([1:4000]*d,n,style=9,logflag="nn");xtitle("trace de la fonction vibration");

```

## Connexions des différents réseaux

Les trois physiques sont reliées par le biais de paramètres dont les valeurs évoluent au cours du temps. Le temps est l'espace fondamental qui va encadrer les calculs effectués dans le domaine des fréquences. Le programme principal va donc être une boucle dans le domaine du temps, dans laquelle un calcul est effectué en fonction de la fréquence. Le résultat doit donc être un spectre qui évolue avec le temps. Avant de compiler le programme complet, il faut retravailler les paramètres de façon à vérifier que les domaines en temps et fréquences sont compatibles et aboutissent à un temps de calcul raisonnable. La boucle en temps s'étale sur 4000 points de calcul. Pour chacun de ces points on va calculer un spectre qui est actuellement calculé sur 1000 points.

L'instruction timer() de SCILAB permet de connaître le temps cpu consommé pour une succession de calculs. Lorsque l'on ressort le temps passé pour les 1000 points en fréquence on obtient 2,21 secondes. Pour 4000 points, on obtient un temps de calcul de 147 minutes, ce qui est beaucoup trop long. Si l'on réduit le nombre d'échantillons à 100 points, on obtient un temps de calcul de 0,17s. Soit 11' pour 4000 points. Pour 1000 points on obtient 3' de temps de calcul environ, ce qui commence à être acceptable. Partons sur ce nombre.

Les paramètres gain et condensateur sont déduits des calculs qui précèdent dans le domaine du temps. Le schéma global (l'algorithme) du programme est présenté figure 18.

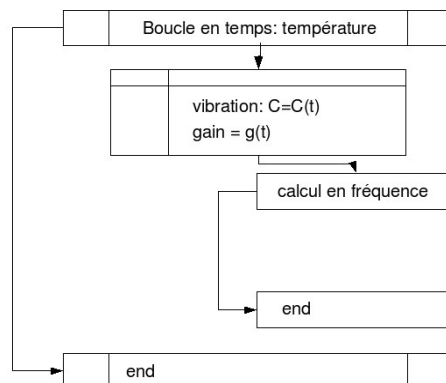


Figure 18

Dans un premier temps, il faut insérer le programme de calcul des vibrations dans le programme thermique. On ajuste le pas de temps du thermique, dix fois trop grand, aux vibrations en divisant ce premier par 10. On ajoute le tableau (n) au programme thermique. On obtient le listing suivant:



$$C6(t) = \frac{\alpha}{n(t)}$$

équation 62

Ayant 100pF pour 2mm de hauteur de fente, alpha vaut  $200 \cdot 10^{-15}$ . Le gain lui, vaut 100 à l'origine et chute avec la température. On admet ici une loi du type:

$$\text{gain} = \text{gain}_0 \left( \frac{Q(t)}{Q_0} \right)^4$$

équation 63

Ce réglage s'effectue dans le même bloc que celui où l'on règle le condensateur de fente.

Le dernier point est la création d'un tableau qui mémorise la variation des résultats de spectres électriques en fonction du temps. La gestion de tableaux de dimension supérieure à 2x2 est difficile. Un objet disponible sous SCILAB peut faciliter cette gestion, c'est la liste. En ajoutant à chaque pas de temps la matrice qui donne le spectre de chaque courant en fréquence à la liste, on mémorise toutes les données.

Malheureusement ce type de liste peut prendre des dimensions très importantes, en l'occurrence ici 100000 données. Souvent SCILAB va afficher un message de stacksize exceed, très pénible, indiquant que la mémoire disponible pour les données est insuffisante pour contenir le tableau. On peut essayer d'agrandir ce stack mais c'est souvent peine perdue, car SCILAB indique le besoin en cours, et non le besoin final. Le plus simple est de réduire les données de sorties!

Dans notre problème, ce qui nous intéresse est la valeur de tension de sortie en fréquence. Ne mémorisons donc que cette seule donnée. On peut aussi réduire l'excursion en temps à 200 points, au moins dans une première étape et tenter d'agrandir le stacksize. Le résultat peut être ramené à un simple tableau 2x2, mais on en oubliera pas pour autant la possibilité des listes pour d'autres usages éventuels ultérieurs.

Le tracé du résultat en trois dimension pose quelques difficultés aussi. En extrayant les premiers points temporels qui empêchent de profiter d'une échelle linéaire mettant en évidence les résonances. La figure 20 montre une vue aérienne de la sortie, indiquant le changement de fréquence de la résonance attaché à la vibration qui modifie la valeur du condensateur et la figure 21 une vue latérale qui présente l'atténuation de la résonance avec le temps.

On constate une augmentation du niveau de tension à la résonance avec l'augmentation de température au cours du temps. Cette application n'exploite pas des modèles physiques réels, mais avait pour but d'approcher la résolution de problèmes multiphysiques et montrer les difficultés provenant de l'implémentation numérique des calculs.

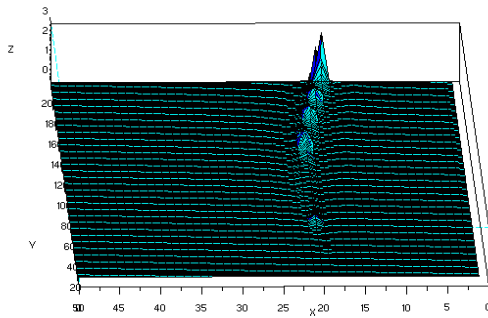


Figure 20: abscisses x = fréquence (x20MHz), y = temps (s)

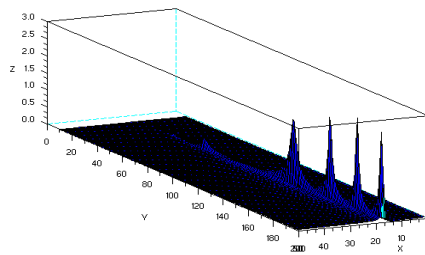


Figure 21: abscisses x = fréquence (x20MHz), y = temps (s); ordonnée: V4 en volt

Le listing obtenu finalement est donné ci-après.

```

clear;
xbase;
stacksize(8000000);

//composants
tR1=100;tG1=1/tR1;
tR2=1000;tG2=1/tR2;
tC3=1;
To=293;
//valeurs des composants
G1=1E-3;
G2=100;
G3=100;
G4=100;
G5=1E-3;
C6=100E-12;
M26=1E-11;
M36=1E-11;
L55=5E-9;
L66=5E-9;
//autres paramètres
h=1E-3;
E0=1E3;
d=1;

//connectivites
A=[1 -1 0 0 0 0
0 0 1 -1 0 0
0 0 0 0 -1 1
0 0 0 0 1 -1];

//initialisation des sources
bv=[0.01 0 0 0 0 -E0*h];

Q1=0;

qK=zeros(1);
P1=1;
sorti=[];
n=[0];
//tableaux et constantes

```

```

Rft=[]; //Rft est la liste qui compile les spectres en fonction du temps
fo=20E6;
gain=100; //valeur de gain a l'origine

disp "debut de calcul";
//-----debut boucle temporelle
for t=[2:200]
W=tG1+tG2+tC3/d;
Q1=1/W*(P1+tC3/d*qK(t-1));
qK=[qK;Q1];

//-----programme des vibrations
ts=t*d/10;
n=[n;2E-3+1E-3*sin(2*pi*ts/40)];
//-----
//ici les paramètres à fixer dans le temps sont définis. On règle ceux utilisés en fréquence
C6=200E-15/n(t);
gain=(qK(t)/290)^4*100;
//-----
K=[]; //tableau de memorisation des valeurs de tensions de noeuds
If=[];
for f=[1:50]
//f=1;
s=i*2*pi*f*fo;
//metriques
bYv=[G1 G2 G3 G4 G5 C6*s]; //Y des RC dans l'espace des branches
bYrc=diag(bYv); //diagonalisation pour avoir le Yrc final, toujours dans l'espace des branches
bZm=[0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 L55*s 0
0 0 0 0 L66*s]; //impedance des energies magnetiques dans l'espace des branches
bYm=pinv(bZm);
bY=bYrc+bYm; //ici on calcule l'inverse de la métrique dans l'espace des branches
nY=A*bY*A';

nV=-pinv(nY)*(A*bY*bV');
K=[K;nV'];
If=[If;(nV(4)-nV(3))/(C6*s)];

bV=[0.01 -M26*s*If(f) gain*(0.01-M36*s*If(f)) 0 0 -E0*h];

end, //fin boucle en frquence
//abs(K(:,2))
//-----
Rft=[Rft,abs(K([1:50],2))];

end,
//-----fin boucle temporelle
disp "fin de calcul";
plot3d([1:50],[10:199],Rft(:,[10:199]));

```

### 4. Méthode nodale différentielle

Comme précédemment, nous allons nous entraîner sur des circuits simples avant d'aborder des problèmes plus complexes. Un premier exemple peut être constitué d'un simple transistor en régime de petits signaux. La figure 22 montre le graphe du problème où les branches sont numérotées et deux paires de noeuds sont repérées par des lignes pointillées.

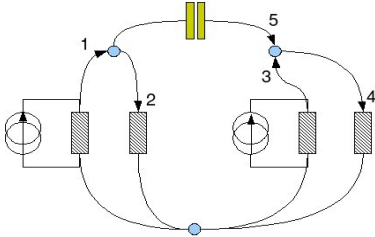


Figure 22

L'équation s'exprime par une relation entre les tensions des paires de noeuds et les courants de paires de noeuds. Les sources de courants de branches sont connues, ainsi que la métrique inverse. En repérant l'espace des paires de noeuds par deux points, on écrit:

$$\ddot{\mathbf{I}}^k = \ddot{\mathbf{Y}}^{kp} \ddot{\mathbf{V}}_p.$$

équation 64

On établit la connectivité entre tensions de branches et de paires de noeuds comme celle entre courants de mailles et courants de branches:

$$\mathbf{C}_m^k = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \\ -1 & 1 \end{bmatrix}$$

équation 65

La définition de Y dans l'espace des branches est ici très simple puisqu'il n'y a pas de couplages. Comme pour les approches nodales, la difficulté dans l'usage de la métrique inverse est dans le calcul des inductances, qui oblige à inverser la matrice Y des inductances avant utilisation.

On doit ensuite transformer les deux objets Y et les sources I de l'espace des branches vers l'espace des paires de noeuds. On calcule:

$$\ddot{\mathbf{Y}}^{ab} = \mathbf{C}_\mu^a \bar{\mathbf{Y}}^{\mu\nu} \mathbf{C}_\nu^b \quad \ddot{\mathbf{I}}^k = \mathbf{C}_m^k \bar{\mathbf{I}}^m$$

équation 66

En inversant l'inverse de la métrique dans l'espace des paires de noeuds on calcule les valeurs de tensions de paires de noeuds. On peut ensuite en déduire les tensions de branches. Programmons ce problème sous SCILAB.

Si l'on désire le programmer en temporel, il nous faut détailler l'expression en différences finies de l'équation 64. La détailler signifie savoir exprimer formellement le produit de Y par les termes inconnus, exprimé dans l'espace des paires de noeuds. Mieux que le petit logiciel que nous avons développé en python, on peut faire appel à Maxima. Maxima est un logiciel gratuit de calcul formel, en ligne de commande. On peut rentrer la définition théorique d'une matrice, ici Y dans l'espace des branches:

$$\bar{\mathbf{Y}}^{ab} = \begin{bmatrix} \mathbf{G1} & 0 & 0 & 0 & 0 \\ 0 & \mathbf{G2} & 0 & 0 & 0 \\ 0 & 0 & \mathbf{G3} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{G4} & 0 \\ 0 & 0 & 0 & 0 & 1/(\mathbf{s.C}) \end{bmatrix}$$

### équation 67

La figure 23 montre la fenêtre de Maxima (en fait une fenêtre de console, ici sous Linux, dans laquelle on a appelé Maxima) et où l'on a rentré notre matrice Y.

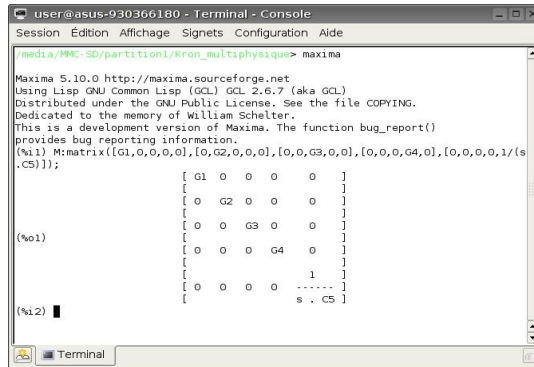


Figure 23

On rentre ensuite la connectivité. Maxima peut alors résoudre l'équation 66 pour obtenir l'expression de Y dans l'espace des paires de noeuds. La figure 24 montre la programmation dans Maxima. De ce résultat on déduit le système d'équation à résoudre, de dimension 2 puisque on a défini deux tensions de paires de noeuds. On doit aussi ressortir les sources. En procédant de la même façon que précédemment, on obtient le vecteur des sources de courant exprimé dans l'espace des paires de noeuds. Pour faciliter l'obtention des matrices sous Maxima, on exprime les impédances, ou admittances en fonctions de Laplace.

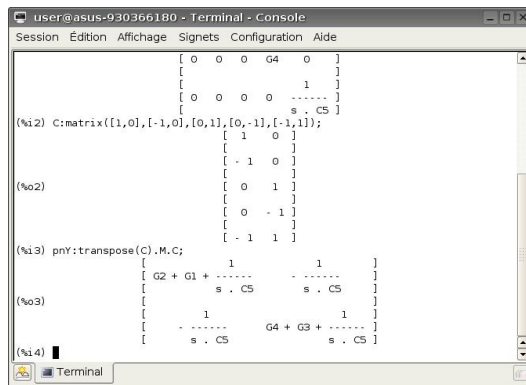


Figure 24

Le système d'équation à résoudre en différences finies est:

$$\begin{cases} (G2 + G1 + sC5) \ddot{v}_1 - sC5 \ddot{v}_2 = \ddot{I}^1 \\ -sC5 \ddot{v}_1 + (G4 + G3 + sC5) \ddot{v}_2 = \ddot{I}^2 \end{cases}$$

### équation 68

Vous avez (vous auriez) dû noter que je me suis trompé: travaillant en admittance, il faut remplacer  $1/(sC)$  par  $sC$ ! On établit alors facilement les équations en différences finies:

$$\begin{cases} \left( G2+G1+\frac{C5}{\delta} \right) \ddot{v}_1(t) - \frac{C5}{\delta} \ddot{v}_2(t) = \ddot{I}^1(t) + \frac{C5}{\delta} K(\ddot{v}_1, t-1) - \frac{C5}{\delta} K(\ddot{v}_2, t-1) \\ -\frac{C5}{\delta} \ddot{v}_1(t) + \left( G4+G3+\frac{C5}{\delta} \right) \ddot{v}_2(t) = \ddot{I}^2(t) - \frac{C5}{\delta} K(\ddot{v}_1, t-1) + \frac{C5}{\delta} K(\ddot{v}_2, t-1) \end{cases}$$

équation 69

On en déduit le Wronksien et les sources:

$$W = \begin{bmatrix} G2+G1+\frac{C5}{\delta} & -\frac{C5}{\delta} \\ -\frac{C5}{\delta} & G4+G3+\frac{C5}{\delta} \end{bmatrix} \quad S = \begin{bmatrix} \ddot{I}^1 \\ \ddot{I}^2 \end{bmatrix}$$

$$T = \begin{bmatrix} \frac{C5}{\delta} K(\ddot{v}_1, t-1) - \frac{C5}{\delta} K(\ddot{v}_2, t-1) \\ -\frac{C5}{\delta} K(\ddot{v}_1, t-1) + \frac{C5}{\delta} K(\ddot{v}_2, t-1) \end{bmatrix}$$

équation 70

On résout le système à chaque pas de temps en résolvant:

$$W^{\mu\nu} \ddot{v}_{\nu} = S^{\mu} + T^{\mu}$$

équation 71

Le listing suivant est le programme écrit sous SCILAB.

```
clear;

//entree des donnees
//composants de branches
G1=0.01;
G2=0.01;
G3=0.01;
G4=0.01;
C5=1E-9;

//connectivite
C=[1 0
-1 0
0 1
0 -1
-1 1];

//pas de temps & echnatillonnage
nbre=1000;
d=1E-9;

v=[];
K=zeros(2,2);
res=[];

//debut de boucle temporelle
for t=[2:nbre]
I1=10*(1-exp(-t/100));
I2=1;

W=[G2+G1+C5/d -C5/d
-C5/d G4+G3+C5/d];
```

```
S=[I1;I2];
T=[C5/d*K(t-1,1)-C5/d*K(t-1,2)
-C5/d*K(t-1,1)+C5/d*K(t-1,2)];

v=pinv(W)*(T+S);

K=[K;v'];

res=[res;C(4,:)*v];

end,

plot2d([1:length(res)]*d,res,style=9,logflag='nn');xgrid(9);
```

On trace sous SCILAB l'évolution dans le temps de la tension aux bornes de la résistance R4 (figure 25).

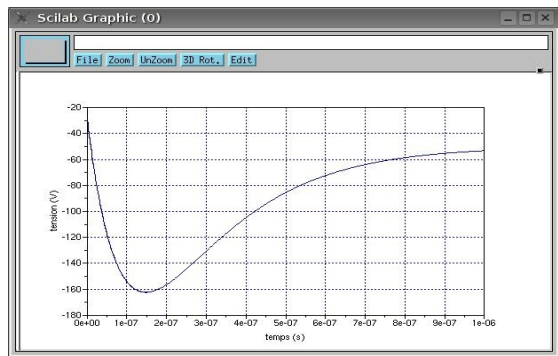


Figure 25

