IUT du Limousin
Département GEII BRIVE

Travaux Pratiques de Mathématiques Appliquées

Deuxième Année

Raymond Quéré

Septembre 2009

Résumé

Cette série de 6 TP de 2h est destinée à vous familiariser avec quelques notions de mathématiques appliquées et avec l'utilisation d'un logiciel dédié à ces applications. Ce logiciel est le logiciel Scilab développé en France par l'INRIA (Institut National de Recherche en Informatique et en Automatique). Il est disponible gratuitement sur le site de l'INRIA et nous travaillerons avec la version Scilab 4.1.2. Ce logiciel a les mêmes fonctionnalités (en beaucoup moins cher) que logiciel Matlab très utilisé dans l'industrie et qui de fait est un standard. Le logiciel Scilab est, en fait un langage de programmation que l'on appelle 'Script' c'est-à-dire que les instructions sont interprétées et exécutées pas à pas à la différence du langage C par exemple où l'ensemble des instructions doit être compilé avant de lancer l'exécution. De ce fait celle-ci est plus lente mais l'apprentissage est beaucoup plus simple. Pour chaque TP il vous faudra rédiger un compte rendu en utilisant le logiciel OpenOffice. La liaison entre Scilab et ce logiciel vous permettra d'intégrer les courbes que vous aurez obtenues avec Scilab dans votre compte rendu.

Cette série de TP est organisée en quatre chapitres qui sont :

1. Introduction à Scilab
2. Matrices
3. Equations différentielles
4. Applications aux systèmes électriques

Chapitre 1

Introduction à Scilab

1.1 Vecteurs

En général les calculatrices effectuent des opérations sur des grandeurs **scalaires** : (a=2, b=5.68, $\cdots$ sont des **scalaires**) et on réalise les opérations +, -, *, / ainsi que l'ensemble des opérations mathématiques qui sont des applications de $\mathbb{R} \mapsto \mathbb{R}$.

Un logiciel tel que Scilab permet d'effectuer ces mêmes opérations sur un espace vectoriel de dimension quelconque. Ainsi la commande suivante initialise la vrariable **v** comme un **vecteur colonne** de composantes $(1,2)$. On dit que $\mathbf{v} \in \mathbb{R}^2$. Plus généralement un vecteur comportant n composantes sera tel que $\mathbf{v} \in \mathbb{R}^n$

```
-->v=[1;2]
 v  =

    1.
    2.
```

De même la commande

```
-->u=[1,2]
 u  =

    1.    2.
```

définit un vecteur **ligne** qui est le transposé de **v** noté $\mathbf{u} = \mathbf{v}'$ en langage Scilab. L'opération de **transposition** consiste donc à échanger les lignes et les colonnes. On peut effectuer les opérations suivantes sur les vecteurs :

addition	$\mathbf{u},\mathbf{v} \in \mathbb{R}^n \Rightarrow \mathbf{w} = \mathbf{u} + \mathbf{v}$	–> u=[1 ;2 ;3] ;v=[0.5 ;1 ;2] ; –>w=u+v -> w = 1.5 3. 5.
multiplication par un scalaire	$\mathbf{v} \in \mathbb{R}^n; \lambda \in \mathbb{R} \Rightarrow \mathbf{w} = \lambda \cdot \mathbf{v} \in \mathbb{R}^n$	lambda=2 ; w=lambda*u -> w = 2. 4. 6.
produit scalaire	$\mathbf{u},\mathbf{v} \in \mathbb{R}^n \Rightarrow \mu = \mathbf{u}' \cdot \mathbf{v} \in \mathbb{R}$	mu=u'*v mu = 8.5
produit simple	$\mathbf{u},\mathbf{v} \in \mathbb{R}^n \Rightarrow \mathbf{w} = \mathbf{u} \times \mathbf{v} \in \mathbb{R}^n$	w=u.*v w = 0.5 2. 6. (notez le point devant *)
division simple	$\mathbf{u},\mathbf{v} \in \mathbb{R}^n \Rightarrow \mathbf{w} = \mathbf{u}/\mathbf{v} \in \mathbb{R}^n$	–>w=u./v w = 2. 2. 1.5
Produit	$\mathbf{u},\mathbf{v} \in \mathbb{R}^n \Rightarrow \mathbf{w} = \mathbf{u} \cdot \mathbf{v}$	interdit

TABLE 1.1 – Opérations élémentaires sur les vecteurs dans Scilab

De plus Scilab autorise de calculer des fonctions mathématiques portant sur des vecteurs : $\mathbf{v} \in \mathbb{R}^n \Rightarrow \mathbf{w} = f(\mathbf{v}) \in \mathbb{R}^n$.

Exemple :

```
v=[0,%pi/2,%pi,3*%pi/2]'
w=cos(v)
```

l'opération $\cos(v)$ sera effectuée sur toutes les composantes de v de façon transparente. On dit que Scilab est un calculateur vectoriel.

1.2 Nombres complexes

Les opération dans Scilab portent sur des scalaires réels ou des vecteurs réels mais il existe d'autres types de variables.

- variables de type logique qui peuvent prendre les valeurs %T (true) et %F (false)
- variables de type chaîne de caractères
- variables de type matrice
- ... (voir les différents types dans l'aide de scilab)

Un type de variable très intéressant est le type complexe. Si $x,\ y \in \mathbb{R}$ alors la variable complexe $z = x + j \cdot y$ où j est tel que $j^2 = -1$ peut être traitée directement dans Scilab. Le nombre complexe j est noté `%i`, dans Scilab et les différentes opérations sur les complexes sont connues du logiciel

module `abs()`

argument `atan()`

partie réelle `real()`

partie imaginaire `imag()`

conjugué `conj()`

```
-->x=1;y=2; z=x+%i*y
 z  =
    1. + 2.i
-->abs(z)
 ans  =
    2.236068

-->atan(y,x)
 ans  =
    1.1071487

-->real(z)
 ans  =
    1.

-->imag(z)
 ans  =
    2.

--> conj(z)
 ans  =
    1. - 2.i
```

En outre on peut utiliser toutes les fonctions mathématiques qui admettent un argument de type complexe et en particulier la fonction exponentielle `exp()` . Scilab est donc un outil qui permet l'étude des fonctions complexes de la variable réelle.

1.3 Graphiques

Les possibilités graphiques de Scilab sont très impressionnantes (en 2D, en 3D, ...) ; nous n'en utiliserons que les fonctionnalités les plus simples.

1.3.1 Un exemple simple

On veut tracer le graphe de la fonction $y = \sin(x)$ pour $x \in [0, 2\pi]$. Pour cela on peut construire un vecteur x comprenant $N+1$ points équidistants de $\frac{2\pi}{N}$ et calculer la fonction de façon vectorielle en utilisant le code suivant ($N = 100$).

`x=linspace(0,2*%pi,101)'`	construit un vecteur colonne de 101 points espacés de 2 %pi/100
`y=sin(x)`	calcule un vecteur colonne de 101 points en prenant les sinus de chaque élément
`plot2d(x,y)`	trace la courbe où x est le vecteur des abscisses et y les ordonnées

1.3.2 Un exemple un peu plus compliqué

L'exemple suivant utilise l'ensemble des paramètres de la fonction `plot2d` et permet de tracer plusieurs courbes sur le même graphique.

`x=linspace(0,2*%pi,101)'`	construit un vecteur colonne de 101 points espacés de 2 %pi/100
`y1=sin(x)`	calcule la première courbe
`y2=cos(x)`	calcule la seconde courbe
`y3=sin(x)+cos(x)`	calcule la troisième courbe
`plot2d(x,[y1,y2,y3])`	trace les 3 courbes

On utilisera utilement la documentation de `plot2d` pour identifier les différentes options de la fonction. On peut aussi utiliser la fonction `plot2d` pour tracer des fonctions paramétriques de la façon très simple suivante :

`t=linspace(0,2*%pi,101)'`	construit un vecteur colonne de 101 points espacés de 2 %pi/100
`y=sin(t)`	calcule la première fonction
`x=cos(t)`	calcule la seconde fonction
`plot2d(x,y)`	trace la courbe paramétrée

1.3.3 exercices

1. A l'aide des exemples précédents représenter sur un même graphique les fonctions :

$$v_1 = \sin(x) \qquad v_2 = \sin(x + \frac{4\pi}{3}) \qquad v_3 = \sin(x + \frac{2\pi}{3})$$

puis

$$u_{12} = v_1 - v_2 \quad u_{31} = v_3 - v_1 \quad u_{23} = v_2 - v_3 \quad u_{21} = v2 - v1 \quad u_{13} = v1 - v3 \quad u_{32} = v3 - v2$$

2. Représenter la fonction
$$y = e^{-m \cdot x} \cdot \sin(x + \frac{\pi}{4})$$
pour les valeurs suivantes de m = [1, 0.5, 0.1, 0.05]. On commencera par tracer la courbe pour chaque valeur de m puis on écrira un programme permettant de tracer les quate courbes à l'aide d'une boucle `for ... end`

3. Représenter les parties réelles et imaginaires en fonction de $\omega \in [-100, 100]$ des fonctions suivantes. Représenter aussi les parties imaginaires en fonction des parties réelles (courbes paramétrées).
$$Z(\omega) = \frac{1}{1+2j\omega}$$
$$Z(\omega) = \frac{1+j\omega}{1+2j\omega}$$
$$Z(\omega) = \frac{e^{-j\omega/4}}{1+j\omega}$$
$$Z(\omega) = \frac{1}{(1+j\omega)(1-\frac{\omega^2}{100}+2j\omega)}$$

4. Utiliser les possibilités de "'copier-coller"' de Scilab pour rédiger votre compte rendu dans Open Office.

Chapitre 2

Matrices

2.1 Matrices

2.1.1 Définition

Les matrices sont des ensembles de nombres réels indicés et disposés sous forme de tableau :

$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ est une matrice 2,2	`A=[1,2;3,4]`
$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{bmatrix}$ est une matrice 2,3	`B=[1,2,3;4,5,6]`
$\mathbf{C} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \\ c_{31} & c_{32} \end{bmatrix}$ est une matrice 3,2	`C=[1,2;3,4;5,6]`

TABLE 2.1 – Définition des matrices dans Scilab

Plus généralement on dit que la matrice $\mathbf{A} \in \mathbb{R}^{m \times n}$ possède m lignes et n colonnes.

2.1.2 Opérations élémentaires

Les opération élémentaires possibles sur les matrices sont l'addition, la multiplication et l'inversion. Elles obéissent à des règles bien précises.

Addition Elle ne peut s'effectuer que sur des matrices de même dimension

Multiplication Si **A** et **B** sont deux matrices, le produit **A.B** n'est possible que si le nombre de colonnes de **A** est égal au nombre de lignes de **B**. Cela signifie que si **A** et **B** ne sont pas carrées le produit **B.A** n'est pas possible.

Transposition Cette opération consiste à inverser les lignes et les colonnes d'une matrice. Ainsi si

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1p} \\ a_{21} & a_{22} & \cdots & a_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mp} \end{bmatrix} \in \mathbb{R}^{m \times p}$$

est une matrice avec m lignes et p colonnes alors la transposée de **A** est

$$\mathbf{A}^t = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1p} & a_{2p} & \cdots & a_{mp} \end{bmatrix} \in \mathbb{R}^{p \times m}$$

. L'opération de transposition est notée par le signe ' dans Scilab. `A'` est la transposée de `A` .

Inversion Cette opération remplace la division. Elle n'est pas toujours possible.

Ces opérations sont résumées dans la Table-2.2

2.1.3 Déterminant d'une matrice

Dans le plan ou dans l'espace la notion de déterminant permet de calculer une aire ou un volume. Cette notion a été généralisée à des espaces à n dimensions.

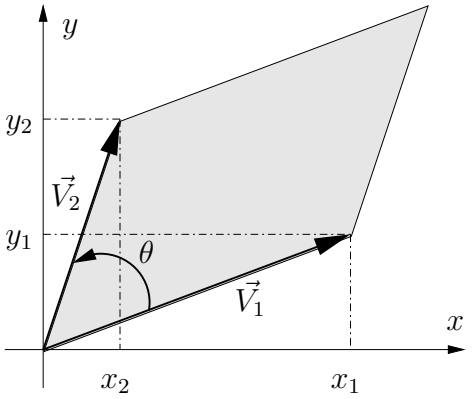
On considère les deux vecteurs $\vec{V_1} = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$ et $\vec{V_2} = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$, la matrice composée des coordonnées des deux vecteurs s'écrit : $\mathbf{V} = \begin{bmatrix} x_1 & x_2 \\ y_1 & y_2 \end{bmatrix}$.

Le déterminant de la matrice **V** est donné par

$$det(\mathbf{V}) = \begin{Vmatrix} x_1 & x_2 \\ y_1 & y_2 \end{Vmatrix} = x_1 \cdot y_2 - x_2 \cdot y_1$$

$$det(\mathbf{V}) = \begin{Vmatrix} x_1 & x_2 \\ y_1 & y_2 \end{Vmatrix} = \left|\vec{V_1}\right| \cdot \left|\vec{V_2}\right| \cdot \sin\theta$$

et représente l'aire grisée de la figure ci-contre



Cette notion de déterminant peut être étendue au calcul du volume défini par 3 vecteurs dans l'espace ou encore n vecteurs dans l'espace à n dimensions. Le déterminant d'une matrice $n \times n$ peut se calculer en le développant par rapport à une ligne ou à une colonne de la façon suivante :

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{vmatrix} = a_{11} \cdot \begin{vmatrix} a_{22} & a_{23} & \cdots & a_{2n} \\ a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n2} & a_{n3} & \cdots & a_{nn} \end{vmatrix} - a_{21} \cdot \begin{vmatrix} a_{12} & a_{13} & \cdots & a_{1n} \\ a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n2} & a_{n3} & \cdots & a_{nn} \end{vmatrix}$$

$$+ a_{31} \cdot \begin{vmatrix} a_{12} & a_{13} & \cdots & a_{1n} \\ a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n2} & a_{n3} & \cdots & a_{nn} \end{vmatrix} - \cdots$$

Le calcul des déterminants est réalisé dans Scilab par l'instruction `det(.)`.

2.1.4 exercices

1. On donne trois matrices :

$$\mathbf{A} = \begin{bmatrix} 0 & 2 & -1 \\ -2 & -1 & 2 \end{bmatrix} ; \mathbf{B} = \begin{bmatrix} 1 & 0 & 1 \\ -1 & 1 & 2 \\ 0 & 2 & 1 \end{bmatrix} ; \mathbf{C} = \begin{bmatrix} -2 & 1 \\ 1 & 0 \\ 0 & 2 \end{bmatrix}$$

Calculer les produits : $\mathbf{A} \cdot \mathbf{B}$, $\mathbf{B} \cdot \mathbf{C}$, $(\mathbf{A} \cdot \mathbf{B}) \cdot \mathbf{C}$ *et* $\mathbf{A} \cdot (\mathbf{B} \cdot \mathbf{C})$. On effectuera les calculs à la main puis à l'aide de Scilab.

Opération en notation mathématique	Test à réaliser dans Scilab
Addition : $\mathbf{A},\mathbf{B} \in \mathbb{R}^{m\times n} \Rightarrow \mathbf{C} = \mathbf{A}+\mathbf{B} \in \mathbb{R}^{m\times n}$	`A=[1,2;3,4] ; B=[0,2;0.5,1] ; C=A+B`
Multiplication : $\mathbf{A} \in \mathbb{R}^{p\times n}\ \mathbf{B} \in \mathbb{R}^{n\times q} \Rightarrow \mathbf{C} = \mathbf{A}\cdot\mathbf{B} \in \mathbb{R}^{p\times q}$ attention le produit $\mathbf{C} = \mathbf{B}\cdot\mathbf{A}$ n'est pas possible	`B=[1,2,3;4,5,6] ; C=[1,2;3,4;5,6] ; D=B*C`
Multiplication par un scalaire : $\mathbf{A} \in \mathbb{R}^{m\times n}\ \lambda \in \mathbb{R} \Rightarrow \mathbf{B} = \lambda\cdot\mathbf{A} \in \mathbb{R}^{m\times n}$	`Lambda=3.0 ; A=[1,2;3,4] ; B=Lambda*A`
Matrice Identité : $\mathbf{I} \in \mathbb{R}^{n\times n}\qquad \mathbf{I} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots \\ \vdots & 0 & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \end{bmatrix}$	`I=eye(4,4)`
Inversion : $\mathbf{A} \in \mathbb{R}^{n\times n}\ det(\mathbf{A}) \neq 0 \Rightarrow \mathbf{B} = \mathbf{A}^{-1} \in \mathbb{R}^{n\times n} \Rightarrow \mathbf{B}\cdot\mathbf{A} = \mathbf{I}$	`A=[1,2;3,4] ; B=inv(A) ; B*A`

TABLE 2.2 – Définition des opérations matricielles dans Scilab

2. Soit la matrice $\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$ Calculer le déterminant de la matrice **A**, vérifiez votre calcul à l'aide de l'instruction `det(A)`.

   On donne maintenant la matrice $\mathbf{B} = \begin{bmatrix} -\frac{3}{4} & \frac{1}{4} & 1 \\ \frac{5}{4} & \frac{1}{4} & -2 \\ -\frac{1}{4} & -\frac{1}{4} & 1 \end{bmatrix}$ Calculer les produits $\mathbf{A}\cdot\mathbf{B}$ et $\mathbf{B}\cdot\mathbf{A}$. Que représente la matrice **B** ?

2.2 Diagonalisation

2.2.1 Valeurs propres et vecteurs propres

Soit $\mathbf{A} \in \mathbb{R}^{n\times n}$ une matrice carrée quelconque de dimension $n \times n$. On appelle **valeur propre** de **A** tout nombre réel ou complexe λ qui satisfait la relation :

$$\mathbf{A}\cdot\mathbf{u} = \lambda\cdot\mathbf{u} \quad \mathbf{u} \neq 0$$

$\mathbf{u} \in \mathbb{R}^n$ est appelé **vecteur propre** de **A** associé à la valeur propre λ.

2.2.2 Polynôme caractéristique

Le polynôme caractéristique de la matrice **A** est défini par

$$P_A(\lambda) = \det(\mathbf{A} - \lambda\mathbf{I})$$

où $\mathbf{I} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & 1 \end{bmatrix}$ est la matrice identité de dimension $n \times n$

- Le polynôme caractéristique est de degré n et possède donc n racines réelles ou complexes.
  - Si les valeurs propres sont calculées sur $\mathbb{C}$, il existe n valeurs propres distinctes ou confondues.

  - Ce n'est pas le cas si les valeurs propres sont calculées sur $\mathbb{R}$ car le polynôme peut ne pas avoir toutes ses racines réelles
- On appelle p l'ordre de multiplicité d'une valeur propre λ_i si $P_A(\lambda) = (\lambda - \lambda_i)^p \cdot Q_A(\lambda)$ où $Q_A(\lambda)$ est un polynôme de degré $n - p$.

2.2.3 Diagonalisation d'une matrice

Une matrice $\mathbf{A} \in \mathbb{R}^{n \times n}$ est diagonalisable si elle peut se mettre sous la forme :

$$\mathbf{A} = \mathbf{P} \cdot \mathbf{D} \cdot \mathbf{P}^{-1}$$

où $\mathbf{D} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}$ est la matrice diagonale constituée des valeurs propres de $\mathbf{A}$ et $\mathbf{P}$ est la matrice de passage constituée des vecteurs propres associés.

La matrice $\mathbf{A}$ est diagonalisable sur $\mathbb{R}$ si les vecteurs propres associés aux valeurs propres forment une base.

2.2.4 Exercices

1. Calculer les valeurs propres de la matrice $\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$. Montrer que les vecteurs $u_1 = \begin{bmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix}$ et $u_2 = \begin{bmatrix} -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix}$ sont des vecteurs propres. Vérifiez vos résultats à l'aide de l'instruction `[D,P,bs]=bdiag(A)`
2. Effectuer le même calcul pour la matrice $\mathbf{A} = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$. Que peut-on dire des valeurs propres ? . Calculez les dans $\mathbb{C}$ ainsi que les vecteurs propres associés. Vérifiez vos résultats en utilisant l'instruction : `[D,P,bs]=bdiag(A+%i*0)`

Chapitre 3

Equations différentielles

3.1 Equations différentielles linéaires

Une équation qui relie les dérivées d'une fonction à cette fonction et à la variable est dite différentielle. Si cette relation est linéaire l'équation est une équation différentielle linéaire.

3.1.1 Equation du premier ordre à coefficients constants

Considérons une fonction $x(t)$ et sa dérivée[1] $x'(t) = \frac{dx(t}{dt} = \dot{x}(t)$. $x(t)$ est solution d'une équation différentielle à coefficients constants si elle satisfait :

$$\begin{cases} \frac{dx(t)}{dt} + \frac{x(t)}{\tau} = f(t) \\ x(0) = X_0 \end{cases}$$

où $f(t)$ est une fonction quelconque et τ est une constante. La résolution de cette équation s'effectue en deux étapes :

Equation sans second membre Cette étape consiste à résoudre l'équation homogène encore appelée équation sans second membre : $\frac{dx(t)}{dt} + \frac{x(t)}{\tau} = 0$. Cette équation est à variables séparables et peut encore s'écrire : $\frac{dx(t)}{x(t)} = -\frac{dt}{\tau}$ dont la solution est : $\ln x(t) = -\frac{t}{\tau} + C$ où C est une constante. En prenant l'exponentielle des membres de gauche et de droite on obtient :

$$x(t) = K \cdot e^{-\frac{t}{\tau}} \quad K = e^C$$

Solution générale Cette étape dépend de la complexité de la fonction $f(t)$ et deux approches sont possibles.

- Si $f(t)$ est une fonction telle qu'il est simple de trouver une solution particulière $x_p(t)$ de l'équation différentielle, alors la solution générale s'écrit : $x(t) = K \cdot e^{-\frac{t}{\tau}} + x_p(t)$ et la constante K est déterminée à partir de la condition initiale $x(0) = K + x_p(0)$.
  Par exemple si $f(t) = A = cte$, la solution $x_p(t)$ est telle que $\dot{x}_p(t) = 0$ et la solution particulière est $x_p(t) = \tau \cdot A$ ce qui, en tenant compte de la condition initiale $x(0) = X_0 = K + \tau \cdot A$ fournit

1. Les notations suivantes de la dérivée sont équivalentes

la solution générale :

$$x(t) = X_0 \cdot e^{-\frac{t}{\tau}} + \tau \cdot A \cdot \left(1 - e^{-\frac{t}{\tau}}\right)$$

- Si $f(t)$ ne permet pas de trouver simplement une solution particulière on utilise la méthode de variation de la constante en posant la solution générale sous la forme $x(t) = K(t) \cdot e^{-\frac{t}{\tau}}$. En dérivant cette expression et en la reportant dans l'équation différentielle on obtient : $\frac{dK(t)}{dt} \cdot e^{-\frac{t}{\tau}} - \frac{K(t) \cdot e^{-\frac{t}{\tau}}}{\tau} + \frac{K(t) \cdot e^{-\frac{t}{\tau}}}{\tau} = f(t)$.

  Après simplification on obtient : $\frac{dK(t)}{dt} \cdot e^{-\frac{t}{\tau}} = f(t)$ et la solution générale est donnée par :

$$\begin{cases} x(t) = K(t) \cdot e^{-\frac{t}{\tau}} \\ K(t) = \int f(t) \cdot \exp(\frac{t}{\tau}) dt \\ x(0 = X_0 \end{cases}$$

3.1.2 Résolution numérique

La méthode du paragraphe précédent n'est pas la seule. On peut aussi utiliser la Transformée de Laplace ou la résolution numérique en utilisant un logiciel tel que Scilab. Dans le cas de la résolution numérique on est amené à **discrétiser** l'équation sur l'ensemble des points

$t = \{0, h, 2h, \cdots nh, \cdots\}$, où h est le pas de discrétisation. La résolution numérique consiste alors à réaliser une approximation de la dérivée au point $t_n = nh$. Plusieurs approximations sont possibles comme indiqué à la figure-3.1.

Euler arrière ou méthode explicite Dans ce cas on approxime la dérivée par la formule : $\left.\frac{dx(t)}{dt}\right|_{t=nh} = \frac{x(n+1) - x(n)}{h}$ et l'équation différentielle s'écrit en $t = nh$

$$x(n+1) = (1 - \frac{h}{\tau}) \cdot x(n) + h \cdot f(nh)$$

qui se calcule aisément lorsque $x(0)$ est connue et h fixé.

Euler avant ou méthode implicite Dans ce cas on approxime la dérivée par la formule : $\left.\frac{dx(t)}{dt}\right|_{t=nh} = \frac{x(n) - x(n-1)}{h}$ et l'équation différentielle s'écrit en $t = nh$

$$x(n) = \frac{x(n-1)}{1 + \frac{h}{\tau}} + \frac{h}{1 + \frac{h}{\tau}} \cdot f(nh)$$

Si $\frac{h}{\tau} \ll 1$ on a $\frac{1}{1 + \frac{h}{\tau}} \simeq 1 - \frac{h}{\tau}$ et les deux méthodes son équivalentes. On verra par la suite que la méthode implicite est plus complexe à mettre en oeuvre que la méthode explicite mais qu'elle est stable à la différence de la méthode explicite.

Méthode du trapèze Dans cette approximation on calcule l'aire située sous la courbe donnant la dérivée $\dot{x}(t)$ de la fonction à obtenir. En utilisant un trapèze pour approximer cette aire on obtient :

$$Aire = x(n+1) - x(n) = \int_{nh}^{(n+1)h} \dot{x}(t) dt = \frac{h}{2} \cdot (\dot{x}(n) + \dot{x}(n+1))$$

et l'équation différentielle devient :

$$x(n+1) = \frac{1-\frac{h}{2\tau}}{1+\frac{h}{2\tau}} \cdot x(n) + \frac{h}{2(1+\frac{h}{2\tau})}(f(n)+f(n+1))$$

C'est la méthode utilisée par défaut dans la majorité des programmes informatiques.

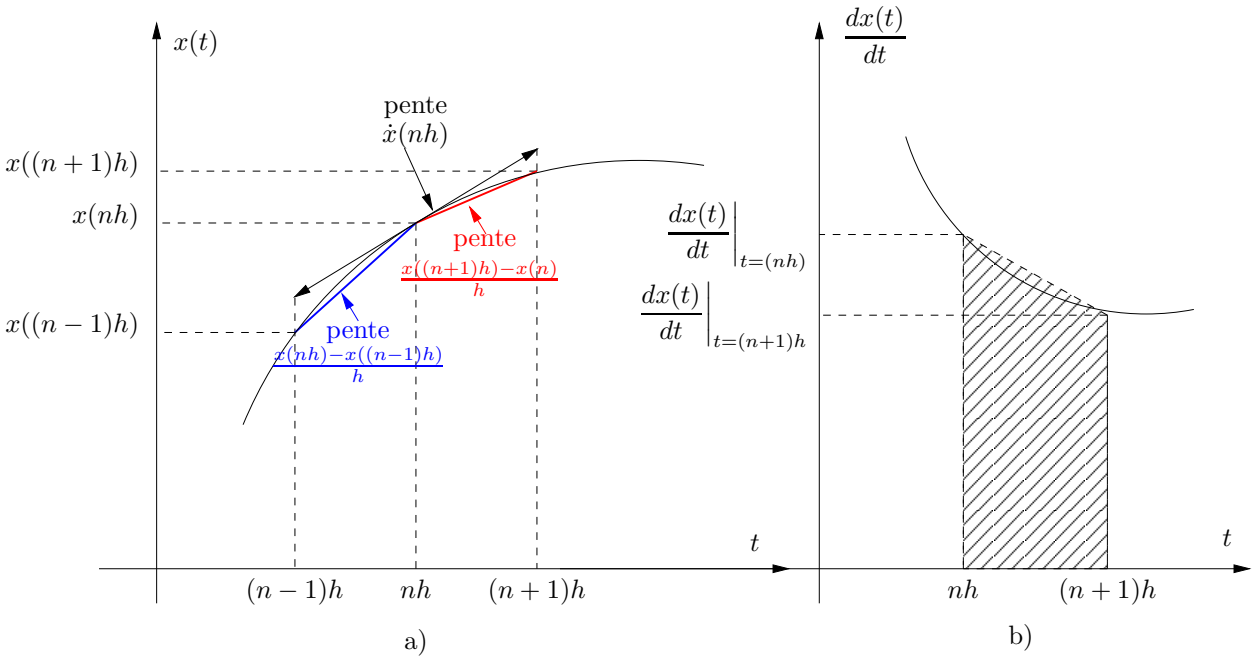


FIGURE 3.1 – Approximation des dérivées a) ; Approximation du trapèze b)

3.1.3 Application

On considère l'équation différentielle suivante :

$$\begin{cases} \dfrac{dx(t)}{dt} + x(t) = 1 \\ x(0) = 0 \end{cases}$$

1. Résoudre théoriquement l'équation précédente. Utiliser Scilab pour tracer la solution pour $t \in [0,5]$
2. Résolution numérique
   (a) Programmer la méthode explicite à l'aide d'une boucle for du type de celle décrite ci-dessous

```
for i=2:N
   x(n)=x(n-1)*(1-h)+h*f(n);
end
```

   Calculer la suite $x(n)$ pour les valeurs suivantes de $h = 0.05$ $h = 0.1$ $et h = 0.25$. Représenter cette suite et comparer les valeurs obtenues aux valeurs théoriques.

   (b) Effectuer le même exercice pour la méthode implicite et pour la méthode des trapèzes. Comparer les trois méthodes.

3.2 Equations différentielles du second ordre à coefficients constants

3.2.1 Méthodes de résolution matricielles

Dans ce cas nous n'examinerons que la résolution numérique. On considère l'équation différentielle suivante

$$\begin{cases} \dfrac{1}{\omega_0^2}\cdot\dfrac{d^2(y(t)}{dt^2}+\dfrac{2\cdot m}{\omega_0}\dfrac{dy(t)}{dt}+y(t)=f(t) \\ y(0)=X_{10} \\ \left.\dfrac{y(t)}{dt}\right|_{t=0}=X_{20} \end{cases}$$

Pour la résoudre numériquement on pose : $y(t)=x_1(t)$ et $\dfrac{dy(t)}{dt}=x_2(t)=\dot{x}_1(t)$. En reportant ces changements de variable dans l'équation différentielle on obtient le système de 2 équations :

$$\begin{cases} \dot{x}_1(t)=x_2(t) \\ \dot{x}_2(t)=-\omega_0^2 x_1(t)-2m\cdot\omega_0 x_2(t)+\omega_0^2 f(t) \\ x_1(0)=X_{10} \\ x_2(0)=X_{20} \end{cases}$$

En définissant le vecteur $\mathbf{X}(t)=\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$ le sytème d'équations précédent peut être mis sous une forme matricielle

$$\begin{cases} \dot{\mathbf{X}}(t)=\mathbf{A}\cdot\mathbf{X}(t)+\mathbf{B}\cdot f \\ \mathbf{X}(0)=\mathbf{X_0} \end{cases}$$

où $\mathbf{A}\in\mathbb{R}^{2\times 2}$ est une matrice donnée par $\mathbf{A}=\begin{bmatrix} 0 & 1 \\ -\omega_0^2 & -2m\omega_0 \end{bmatrix}$ et $\mathbf{B}\in\mathbb{R}^{2\times 1}$ est donnée par $\mathbf{B}=\begin{bmatrix} 0 \\ \omega_0^2 \end{bmatrix}$ Les méthodes de résolution exposées précédemment sont toujours valables à condition de travailler sur le vecteur $\mathbf{X}(t)$. On a ainsi les formules suivantes :

Méthode explicite Dans ce cas on a :

$$\mathbf{X}(n+1)=(\mathbf{I}+h\mathbf{A})\cdot\mathbf{X}(n)+h\cdot\mathbf{B}\cdot f(n)$$

où $\mathbf{I}=\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ est la matrice identité.

Méthode implicite

$$(\mathbf{I}-h\mathbf{A})\cdot\mathbf{X}(n)=\mathbf{X}(n-1)+h\cdot\mathbf{B}\cdot f(n)$$

Dans ce cas la solution n'est pas obtenue immédiatement. Elle nécessite le calcul de l'inverse de la matrice $\mathbf{I}-h\mathbf{A}$

Méthode du trapèze

$$(\mathbf{I}-\frac{h}{2}\mathbf{A})\cdot\mathbf{X}(n+1)=(\mathbf{I}+\frac{h}{2}\mathbf{A})\cdot\mathbf{X}(n)+\frac{h}{2}\cdot\mathbf{B}\cdot(f(n)+f(n+1))$$

3.2.2 Applications

On considère l'équation différentielle du second ordre précédente pour laquelle on choisit $\omega_0=1$, $f(t)=1$ et $\mathbf{X}(0)=\mathbf{0}$. On a donc $\mathbf{A}=\begin{bmatrix} 0 & 1 \\ -1 & -2m \end{bmatrix}$ et $\mathbf{B}=\begin{bmatrix} 0 \\ 1 \end{bmatrix}$. Le programme suivant permet de résoudre l'équation par les trois méthodes : explicite, implicite et trapèze.

```
[ok,h,m]=getvalue("choix du pas et du coefficient d amortissement",...
                  ["h";"m"],list("vec",1,"vec",1),["0.05";"0.5"]);
 t=[0:h:50]';
 N=prod(size(t));
 A=[0,1;-1,-2*m];
 B=[0;1];
 I=eye(2,2);
 f=ones(N,1);
 Xexact=zeros(2,N);Xexpl=zeros(2,N);Ximpl=zeros(2,N);Xtrap=zeros(2,N);
 for n=2:N
  Xexpl(:,n)=(I+h*A)*Xexpl(:,n-1)+h*B*f(n);
  Ximpl(:,n)= inv(I-h*A)*(Ximpl(:,n-1)+h*B*f(n));
  Xtrap(:,n)=inv(I-h*A/2)*((I+h*A/2)*Xtrap(:,n-1)+h*B/2*(f(n)+f(n-1)));
 end
 plot2d(t,[Xexpl',Ximpl',Xtrap'])
```

1. Déterminer l'expression de la solution exacte. (On pourra utiliser les transformées de Laplace)
2. Etudier le fonctionnement du programme précédent pour les valeurs suivantes de $m = 0.25$; $m = 0.5$; $m = 1$ Pour chacune de ces valeurs on choisira différents pas de calcul $h = 0.1$; $h = 0.5$; $h = 1$.
3. Comparer les trois méthodes.

3.3 Equations différentielles non linéaires

Une équation différentielle est dite non linéaire si les coefficients de l'équation dépendent de la solution. C'est une équation de la forme : $a(y)\cdot\dfrac{d^2y(t)}{dt^2}+b(y)\cdot\dfrac{dy(t)}{dt}+c(y)=f(t)$. Cette équation peut être traitée suivant les méthodes exposées précédemment en la décomposant en un système d'équations du premier ordre.

3.3.1 Equation de Vanderpol

L'équation de Vanderpol est l'une des plus célèbres de la physique. Elle s'écrit :

$$\frac{d^2y(t)}{dt^2}=c\cdot(1-y^2(t))\cdot\frac{dy(t)}{dt}-y(t)$$

Pour la résoudre on la transforme en un système de deux équations du premier ordre en posant : $y(t)=u_1(t)$ et $\dfrac{dy(t)}{dt}=u_2(t)=\dot{u}_1(t)$. Le système obtenu s'écrit alors :

$$\frac{d}{dt}\begin{bmatrix}u_1(t)\\u_2(t)\end{bmatrix}=\begin{bmatrix}u_2(t)\\c\cdot(1-u_1^2(t))u_2(t)-u_1(t)\end{bmatrix}=\begin{bmatrix}F_1(u_1,u_2)\\F_2(u_1,u_2)\end{bmatrix}$$

La fonction Scilab `ode}`permet de résoudre ce système. La syntaxe simplifiée de cette fonction est la suivante : `[u]=ode(u0,t0,t,F)` où

$$[\mathbf{u}]=\begin{bmatrix}u_1(t_0) & u_1(t_1) & \cdots & u_1(t_M)\\ u_2(t_0) & u_2(t_1) & \cdots & u_2(t_M)\end{bmatrix}$$

est la matrice des solutions calculées aux points $t=[t_0,\,t_1,\cdots,\,t_M]$ fournis dans le vecteur `t`. `t0` est l'instant initial qui sera généralement pris à 0 et `u0` est le vecteur des conditions initiales.

On a $[\mathbf{u_0}]=\begin{bmatrix}u_1(0)\\u_2(0)\end{bmatrix}$. `F` est une chaîne de caractères donnant le nom de la fonction non linéaire.

Exemple : **Equation de Vanderpol** [2]

```
deff('[f]=Vanderpol(t,x)',['f(1)=x(2)';'f(2)=0.05*(1-x(1)^2)*x(2)-x(1)'])
t=linspace(0,50,500);
u0=[-5;5];
[u]=ode(u0,0,t,Vanderpol);
plot2d(u(1,:)', u(2,:)')
```

Le tracé est effectué dans le plan $(u_1(t),\ u_2 = \frac{du_1(t)}{dt})$ que l'on appelle le plan de phase. La courbe paramétrique $\Phi(t) = \begin{bmatrix}\Phi_1(t,u_1(0))\\ \Phi_2(t,u_1(0))\end{bmatrix}$ est appelée trajectoire de phase. En chaque point du plan de phase on peut définir le vecteur tangent à la trajectoire dont les composantes sont $\begin{bmatrix}du_1 = F_1(u_1,u_2)dt\\ du_2 = F_2(u_1,u_2)dt\end{bmatrix}$. L'ensemble de ces vecteurs représente le champ de vecteurs caractéristique du système d'équations différentielles. On peut le tracer avec l'instruction Scilab `fchamp`. Le script suivant réalise le tracé du champ de vecteur pour l'équation de VanderPol.

```
deff('[f]=Vanderpol(t,x)',['f(1)=x(2)';'f(2)=0.05*(1-
x(1)^2)*x(2)-x(1)'])
t=linspace(0,50,500);
x=linspace(-5,5,30);
xbasc();
fchamp(Vanderpol,0,x,x)
xselect()
u0=[-5;5];
[u]=ode(u0,0,t,Vanderpol);
plot2d(u(1,:)', u(2,:)',2,"000")
```

3.3.2 Exercices

1. Réaliser plusieurs tracés pour les valeurs suivantes de $c = 0.05,\ c = 0.1,\ c = 0.5, \tilde{=} 1.0$
2. Etudier l'influence des conditions initiales
3. Résoudre l'équation différentielle suivante :

$$\frac{d^2\phi(t)}{dt^2} + 2m\cdot\cos\phi\frac{d\phi(t)}{dt} + \sin\phi(t) = 0$$

pour $m = [0.1,\ 0.25,\ 0.5,\ 0.707,\ 1]$. Tracer le champ de vecteurs du système ainsi que les trajectoires dans le plan de phase.

2. Le script se trouve dans le fichier Vanderpol.sce

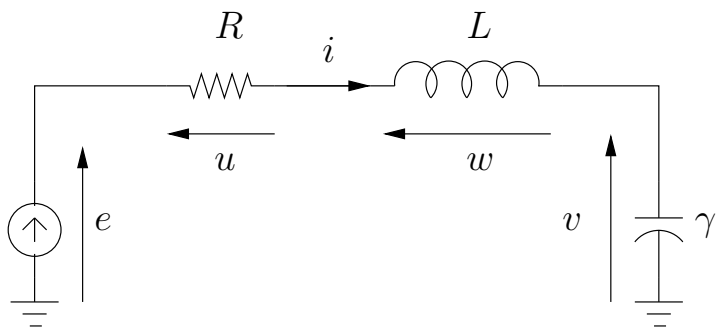
Chapitre 4

Applications aux systèmes électriques

4.1 Réponse d'un système linéaire : utilisation de `syslin`

4.1.1 Forme d'état d'un système linéaire

La méthode vue au TD précédent est en fait la méthode des équations d'état d'un système linéaire. Elle est très employée en électricité et en automatique et Scilab dispose de fonctions intégrées permettant de calculer de tels systèmes. On considère le circuit RLC suivant :



Les équations du circuit sont les suivantes $\begin{cases} \gamma \dfrac{dv}{dt} & = i \\ L\dfrac{di}{dt} & = -v - Ri + e \end{cases}$ On pose $\mathbf{X(t)} = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} \gamma v \\ Li \end{bmatrix}$ et $\dot{\mathbf{X}}(t) = \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} \gamma \dot{v} \\ L\dot{i} \end{bmatrix}$

1. Montrer que l'on a $\dot{\mathbf{X}}(t) = \mathbf{A} \cdot \mathbf{X}(t) + \mathbf{B} \cdot e$
   avec $\mathbf{A} = \begin{bmatrix} 0 & \dfrac{1}{L} \\ -\dfrac{1}{\gamma} & -\dfrac{R}{L} \end{bmatrix}$ et $\mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$
2. On veut mesurer les tensions u , v et w . On pose alors $\mathbf{Y} = \begin{bmatrix} v \\ u \\ w \end{bmatrix}$, $\mathbf{C} = \begin{bmatrix} \dfrac{1}{\gamma} & 0 \\ 0 & \dfrac{R}{L} \\ -\dfrac{1}{\gamma} & -\dfrac{R}{L} \end{bmatrix}$ et $\mathbf{D} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$.
   Montrer que l'on a alors $\mathbf{Y}(t) = \mathbf{C} \cdot \mathbf{X}(t) + \mathbf{D} \cdot e$
   On dit que le système linéaire se met sous la **forme d'état canonique** :

$$\begin{cases} \dot{\mathbf{X}}(t) & = \mathbf{A} \cdot \mathbf{X}(t) + \mathbf{B} \cdot e \\ \mathbf{Y}(t) & = \mathbf{C} \cdot \mathbf{X}(t) + \mathbf{D} \cdot e \end{cases}$$

c'est à dire que le système est complètement caractérisé par la connaissance des matrices d'état : **A**, **B**, **C**, **D**, .

3. Calculer la Transformée de Laplace du système d'équations précédent. Montrer alors que le vecteur des transformées de Laplace $\mathbf{Y}(p)$ est donné par :

$$\mathbf{Y}(p) = \left[\mathbf{C}\cdot(p\mathbf{I}-\mathbf{A})^{-1}\cdot\mathbf{B}+\mathbf{D}\right]\cdot E(p)$$

où $\mathbf{I} = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$ est la matrice identité et $E(p)$ est la Transformée de Laplace de $e(t)$. Montrer que l'on a :

$$(p\cdot\mathbf{I}-\mathbf{A})^{-1} = \frac{1}{p^2+2m\omega_0+\omega_0^2}\cdot\begin{bmatrix}p+\dfrac{R}{L} & \dfrac{1}{L}\\ -\dfrac{1}{\gamma} & p\end{bmatrix}$$

Donner les valeurs de m et ω_0.

4. En déduire les expressions des fonctions de transfert suivantes : $H_1(p) = \dfrac{V}{E}(p)$, $H_2(p) = \dfrac{U}{E}(p)$ et $H_3(p) = \dfrac{W}{E}(p)$.

4.1.2 Utilisation de `Scilab`

En fait Scilab permet de traiter le système suivant à l'aide de l'instruction `syslin`. Ainsi le système linéaire précédent pourra être représenté par un objet S défini par : `S=syslin('c',A,B,C,D)`. Une fois S défini il sera possible de lui appliquer toutes les opérations utiles telles que :

- `csim` qui permet de calculer la réponse temporelle à une excitation donnée
- `bode` qui permet de tracer le diagramme de Bode de la fonction de transfert
- `nyquist` qui permet de tracer le diagramme de Nyquist de la fonction de transfert
- `plzr` qui permet de représenter les pôles et les zéros de la fonction de transfert.

On donne les valeurs numériques suivantes $L = 10^-3\,H$, $\gamma = 10^{-9}\,F$ et $R = 15.811\,\Omega$.

1. Calculer numériquement m et ω_0.
2. A partir de l'aide de `csim`, on utilisera le programme qui permet de tracer la réponse du système à un échelon de tension $e(t) = 1\ t \geq 0$

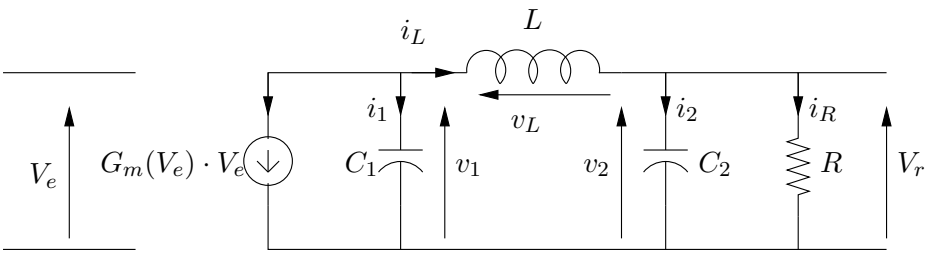
```
t=linspace(0,40/omega0,201);
[y]=csim(step,t,h);
xbasc()
plot2d1(onn,t,y);
xgrid();
```

3. Tracer les diagrammes de Bode de Nyquist des pôles et des zéros des fonctions de transfert . $H_1(p) = \dfrac{V}{E}(p)$, $H_2(p) = \dfrac{U}{E}(p)$ et $H_3(p) = \dfrac{W}{E}(p)$.
4. Renouveler les calculs précédents pour les valeurs de R suivantes [1] :
   $R = 50,\ 100;200,\ 400,\ 800,\ 1600,\ 3200,\ 6400\,\Omega$

1. On pourra consulter le programme `RLC.sce` donné en annexe

4.2 Oscillateur Colpitts

On considère le circuit électrique représenté à la figure suivante :



$G_{m0} = 0.05$ $R = 500\Omega$ $L = 1\mu H$ $C_1 = 0.6nF$ $C_2 = 0.2nF$

On définit les variables d'état : $x_1 = C_1 \cdot v_1$, $x_2 = L \cdot i_L$, $x_3 = C_2 \cdot v_2$ et le vecteur $\mathbf{X} = [x_1\ x_2\ x_3]^T$. Ecrire les équations électriques du circuit et mettre ces équations sous la forme d'état :

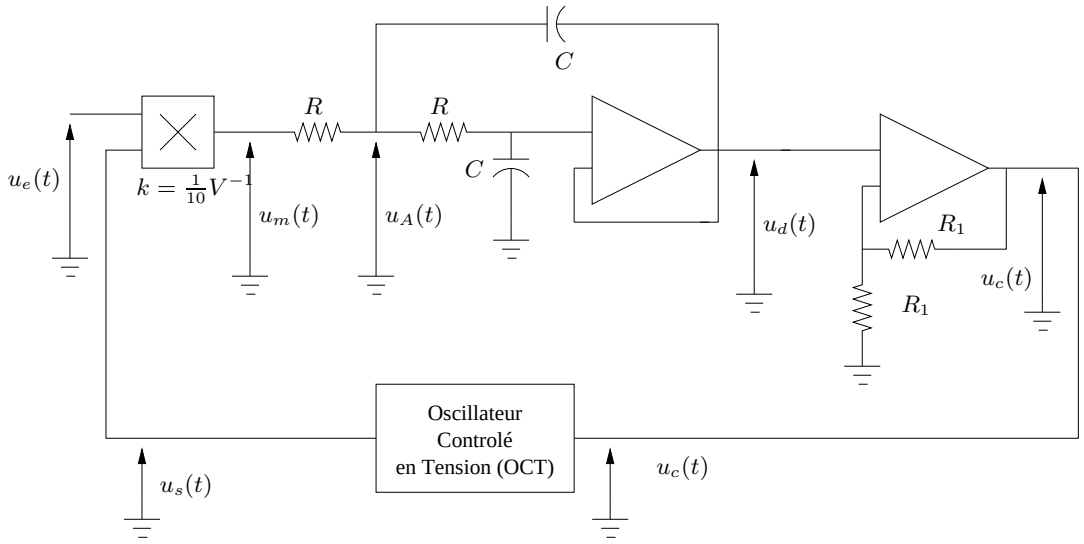
$$\begin{cases} \dot{\mathbf{X}} & = \mathbf{A}\cdot\mathbf{X} + \mathbf{B}\cdot V_e \\ V_r & = \mathbf{C}\mathbf{X} \end{cases}$$

On suppose qu'en "petit signal" on a : $G_m(V_e) = G_{m0}$. Donner l'expression des matrices **A**, **B** et **C**. A l'aide de la fonction `syslin` calculer et tracer :

- Les diagrammes de Bode et de Nyquist de la transmittance $T(p) = \dfrac{V_r(p)}{V_e(p)}$
- Déterminer la fréquence f_0 pour laquelle on a : $\phi(\omega_0) = \arg(T(\omega_0)) = 0$
- La transconductance $G_m(V_e)$ est donnée par
  $G_m(V_e) = G_{m0} \cdot \dfrac{2 \cdot B_1(x)}{x \cdot B_0(x)}$ où $x = \dfrac{V_e}{V_T}$ avec $V_T = 0.025$ et où $B_1(x)$ et $B_0(x)$ sont les fonctions de Bessel modifiées de première espèce qui sont calculées dans Scilab à l'aide de la fonction `besseli(alpha,x)` où alpha=0 pour B_0 et alpha=1 pour B_1. Utiliser ces fonctions pour tracer $|T(\omega_0)|$ en fonction de V_e. En déduire la valeur de V_e pour laquelle $|T(\omega_0)| = 1$

4.3 Boucle de phase du premier ordre

On considère une boucle de phase du premier ordre représentée à la figure suivante :



$R = 10K\Omega$ $C = 1nF$ $R_1 = 2.2K\Omega$

Le signal d'entrée est sinusoidal et s'écrit $u_e(t) = U_e \cdot \sin(\omega_e t + \varphi_e)$. Le signal en sortie de l'oscillateur contrôlé en tension est : $u_s(t) = U_s \cdot \cos(\omega_0 t + \varphi_s(t))$ avec $\dfrac{d\varphi_s(t)}{dt} = K_0 \cdot u_c(t)$. Le multiplieur réalise la multiplication des deux signaux et sa sortie est donnée par $u_m(t) = k \cdot u_e(t) \cdot u_s(t)$ où $k = \frac{1}{10} \cdot V^{-1}$. On pose $RC = \tau$.

- Montrer que les relations entre les tensions $u_m(t)$, $u_A(t)$ et $u_d(t)$ s'écrivent sous la forme du système d'équations différentielles :

$$\begin{cases} \dfrac{du_d(t)}{dt} = & -\dfrac{1}{\tau} \cdot u_d + \dfrac{1}{\tau} \cdot u_A \\ \dfrac{du_A(t)}{dt} = & -\dfrac{1}{\tau} \cdot u_A + \dfrac{1}{\tau} \cdot u_m(t) \end{cases}$$

- En tenant compte des relations données par le multiplieur et l'OCT et en posant $x_1(t) = u_d(t)$, $x_2(t) = u_A(t)$ et $x_3(t) = \varphi_s(t)$, montrer que le fonctionnement de la boucle de phase est régie par le système d'équations du premier ordre.

$$\begin{cases} \dfrac{dx_1(t)}{dt} = & -\dfrac{1}{\tau} \cdot x_1 + \dfrac{1}{\tau} \cdot x_2 \\ \dfrac{dx_2(t)}{dt} = & -\dfrac{1}{\tau} \cdot x_2 + \dfrac{kU_eU_s}{\tau} \cdot \sin(\omega_e t) \cdot \cos(\omega_0 t + x_3) \\ \dfrac{dx_3(t)}{dt} = & 2K_0 \cdot x_1 \end{cases}$$

- On donne les valeurs numériques suivantes : $\tau = 10^{-5}$, $U_e = U_s = 3\ V$, $k = 0.1\ V^{-1}$, $f_0 = \frac{\omega_0}{2\pi} = 50KHz$. On suppose de plus que les valeurs initiales des variables d'état x_1 , x_2 , x_3 sont $x_1(0) = 0$, $x_2(0) = 0$, $x_3(0) = 0$. Par ailleurs on doit construire le vecteur des dates pour la résolution du système d'équations différentielles. On prend $t = [0, \Delta t, 2\Delta t, \cdots, t_{max}]$ avec $\Delta t = \dfrac{1}{50 f_0}$ et $t_{max} = 5\dfrac{1}{|f_e - f_0|}$
  A l'aide de la fonction `ode` résoudre le système d'équations différentielles précédent pour les valeurs de la fréquence d'entrée suivantes :
  $f_e = 41KHz, 43KHz, 45KHz, 47KHz, 49KHz, 51KHz, 53KHz, 55KHz, 57KHz, 59KHz$
  Tracer les graphes des tensions $u_d(t)$ et de la quantité $\phi_s(t) = x_3 - (\omega_e - \omega_0) \cdot t$
- On prend $f_e = 50\ KHz$ et $\tau = 10^{-4}$. Tracer $\phi_s(t)$ et $u_d(t)$. Conclusion ?

Annexe : Programme RLC.sce

```
// Programme RLC.sce
 deff('[A,B,C,D]=RLC_init(L,C,R)',['A=[0,1/L;-1/C,-R/L]','B=[0;1]','C=[1/C,0;0,R/L;-1/C,-R/L]
labels=["Resistance";"Inductance";"Capacite"];
[ok,R,L,C0]=getvalue("Entrez les valeurs des elements du circuit",labels,...
                     list("vec",1,"vec",1,"vec",1),["200";"1e-3";"1e-9"])
[A,B,C,D]=RLC_init(L,C0,R);
omega0=1/sqrt(L*C0);

t=linspace(0,40/omega0,1001)';

h=syslin('c',A,B,C,D);

[y]=csim('step',t,h);
xset("window",0);
xbasc()
plot2d(t,y');
xselect()
xgrid();

xset("window",1);
xbasc()
bode(h,0.01*omega0,10*omega0)

xset("window",2);
xbasc()
nyquist(h,0.01*omega0,10*omega0)

xset("window",3);
xbasc()
plzr(h)
```