ANALYSE NUMÉRIQUE

T.P. N° 2

Alexandre Peltier
Louis Pendu

N° Groupe : B6

Observations :

Notes :	ANALYSE	:	**Total :**
	RÉSULTATS	:	
	PROGRAMMATION	:	
	RAPPORT	:	

EISTI, 1er juin 2011

Rapport du TP2

1er juin 2011

Table des matières

1 Introduction

De façon générale, une réaction chimique consiste à transformer une quantité d'un produit X en un produit Y. Ceci est également valide pour la réaction de Van Der Vusse, qui entre autre est une expérience exothermique(qui dégage de la chaleur), et qui va nous intéresser par la suite.
Le but de ce TP est, à partir de données observées à savoir l'entrée et la sortie de produits chimiques dans un réacteur, d'élaborer un système de prévision.Ce système évaluera la quantité de sortie du produit à un instant t quelconque en se basant uniquement sur des expériences passées.

2 Un peu de théorie...

Dans cette réaction de Van Der Vusse nous avons plusieurs paramètres à prendre en compte :

-c_A : la concentration du produit A (entrée)

-T : la température dans le réacteur

-T_K : la température dans le réfrigérant

-F : le flot d'entrée du produit A

-Q_K : dégagement de chaleur de la réaction

-c_B : la concentration du produit B (sortie)

Les trois premiers paramètres sont mesurés, F et Q_K sont des variables manipulées et c_B est une variable contrôlée par ces deux variables.

Pour décrire la dynamique du système, on introduit un système d'état dont l'équation (sous forme différentielle) est la suivante :

$$\dot{x} = Ax(t) + bu(t) \tag{1}$$

x(t) représente les variations des flots d'entrée/sortie à un instant t
x_1 représente la variation du flot d'entrée du produit A
x_2 représente la variation du flot d'entrée du produit B
u(t) représente la commande du système à un instant t
bu(t) permet de gérer la stabilisation de la réaction.

Nous choisissons un temps initial $t_0 = 0$, un temps final $t_1 = 12$ et un temps d'échantillonnage $dt = 0.05$.

Dans notre cas, nous avons A et b qui sont constants et la commande u(t) qui vaudra 1 ou 5. Ainsi nous avons les conditions suivantes :

$$\dot{x} = \begin{pmatrix} \dot{x_1} \\ \dot{x_2} \end{pmatrix} \tag{2}$$

$$A = \begin{pmatrix} -1 & 10 \\ 1 & 0 \end{pmatrix} \tag{3}$$

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \tag{4}$$

$$b = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{5}$$

$$u(t) = 1 \tag{6}$$

ou

$$u(t) = 5 \tag{7}$$

Ainsi, nous allons essayer de mettre en dépendance les variables x_1 et x_2. Pour cela nous allons utiliser l'identification par la méthode des moindres carrés. On déclare le fait que $x_2(t+1)$ est une combinaison linéaire des $x_1(0)$, $x_1(1)$, ..., $x_1(t)$, $x_2(0)$, $x_2(1)$, ..., $x_2(t)$

$$x_2(t+1) = a_1x_1(t) + ... + a_{t+1}x_1(0) + b_1x_2(t) + ... + b_{t+1}x_2(0) \tag{8}$$

a_1,...,a_{t+1},b_1,...b_{t+1} expriment ici la dynamique du système.
Le problème que l'on cherche à résoudre étant le suivant :

Peut-on calculer la dépendance entre ce qui sortira à un instant t et ce qui est déjà entré et sorti auparavant ?

3 ... Avant la pratique

Comme indiqué dans l'énoncé nous avons créé quatre programmes en Scilab : un étant le menu principal et les autres effectuant l'identification de la sortie du système par moindres carrés selon les modèles du premier, deuxième et troisième ordre.

Pour lancer le programme veuillez ouvrir Scilab, vous mettre dans le bon répertoire courant et taper la commande TP2()

3.1 Premier ordre

Pour le modèle du premier ordre on a l'équation suivante :

$$x_2(t+1) = a_1 x_1(t) + b_1 x_2(t) \tag{1}$$

Notons :

$$a = \begin{pmatrix} a_1 \\ b_1 \end{pmatrix} \tag{2}$$

Pour trouver la matrice "a" nous sommes amenés à résoudre le système matriciel suivant :

$$y = X * a \tag{3}$$

$$\begin{pmatrix} x_2(1) \\ \dots \\ \dots \\ x_2(241) \end{pmatrix} = \begin{pmatrix} x_1(0) & x_2(0) \\ \dots & \dots \\ \dots & \dots \\ x_1(240) & x_2(240) \end{pmatrix} \begin{pmatrix} a_1 \\ b_1 \end{pmatrix} \tag{4}$$

L'inconnue dans l'équation précédente est a. Le but étant de le trouver nous allons réécrire l'équation précédente afin de trouver a.
Le problème est que la matrice X n'est pas une matrice carrée, par conséquent son inverse n'existe pas. Nous passons alors par son pseudoïnverse (X^+). Cela revient à écrire l'équation suivante :

$$a = X^+ * y \tag{5}$$

Le passage par le pseudoïnverse est le "coeur" de notre problème et le but de notre programme. En effet, on essaye de voir à quel(s) moment(s) nous pouvons l'utiliser et de quelle manière. Tout d'abord nous utilisons la fonction svd (singular value decomposition) de Scilab qui prend en paramètre la matrice X. Elle renvoie alors 3 matrices : U, Delta et V que nous allons utiliser pour le calcul de X^+.

Il faut, au préalable, retoucher Delta et U en calculant respectivement leur pseudoïnverse $Delta^+$ et transposée U^T. On a ainsi l'équation suivante :

$$X^+ = V * Delta^+ * U^T \tag{6}$$

A présent, on peut aisément résoudre l'équation (5) en faisant le calcul matriciel de X^+ et de y.

3.2 Deuxième ordre

Pour le modèle du deuxième ordre on a l'équation suivante :

$$x_2(t+1) = a_1 x_1(t) + a_2 x_1(t-1) + b_1 x_2(t) + b_2 x_2(t-1) \tag{7}$$

Notons :

$$a = \begin{pmatrix} a_1 \\ b_1 \\ a_2 \\ b_2 \end{pmatrix} \tag{8}$$

Pour trouver la matrice "a" nous sommes amenés à résoudre le système matriciel suivant :

$$y = X * a \tag{9}$$

$$\begin{pmatrix} x_2(2) \\ ... \\ ... \\ x_2(241) \end{pmatrix} = \begin{pmatrix} x_1(0) & x_2(0) & x_1(1) & x_2(1) \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ x_1(239) & x_2(239) & x_1(240) & x_2(240) \end{pmatrix} \begin{pmatrix} a_1 \\ b_1 \\ a_2 \\ b_2 \end{pmatrix} \tag{10}$$

L'inconnue dans l'équation précédente est a. Le but étant de le trouver nous allons réécrire l'équation précédente afin de trouver a.
Le problème est que la matrice X n'est pas une matrice carrée, par conséquent son inverse n'existe pas. Nous passons alors par son pseudoïnverse (X^+). Cela revient à écrire l'équation suivante :

$$a = X^+ * y \tag{11}$$

Le passage par le pseudoïnverse est le "coeur" de notre problème et le but de notre programme. En effet, on essaye de voir à quel(s) moment(s) nous pouvons l'utiliser et de quelle manière. Tout d'abord nous utilisons la fonction svd (singular value decomposition) de Scilab qui prend en paramètre la matrice X. Elle renvoie alors 3 matrices : U, Delta et V que nous allons utiliser pour le calcul de X^+.

Il faut, au préalable, retoucher Delta et U en calculant respectivement leur pseudoïnverse $Delta^+$ et transposée U^T. On a ainsi l'équation suivante :

$$X^+ = V * Delta^+ * U^T \tag{12}$$

A présent, on peut aisément résoudre l'équation (11) en faisant le calcul matriciel de X^+ et de y.

3.3 Troisième ordre

Pour le modèle du troisième ordre on a l'équation suivante :

$$x_2(t+1) = a_1x_1(t)+a_2x_1(t-1)+a_2x_1(t-2)+b_1x_2(t)+b_2x_2(t-1)+b_3x_2(t-2) \tag{13}$$

Notons :

$$a = \begin{pmatrix} a_1 \\ b_1 \\ a_2 \\ b_2 \\ a_3 \\ b_3 \end{pmatrix} \tag{14}$$

Pour trouver la matrice "a" nous sommes amenés à résoudre le système matriciel suivant :

$$y = X * a \tag{15}$$

$$\begin{pmatrix} x_2(3) \\ ... \\ ... \\ x_2(241) \end{pmatrix} = \begin{pmatrix} x_1(0) & x_2(0) & x_1(1) & x_2(1) & x_1(2) & x_2(2) \\ ... & ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... & ... \\ x_1(238) & x_2(238) & x_1(239) & x_2(239) & x_1(240) & x_2(240) \end{pmatrix} \begin{pmatrix} a_1 \\ b_1 \\ a_2 \\ b_2 \\ a_3 \\ b_3 \end{pmatrix} \tag{16}$$

L'inconnue dans l'équation précédente est a. Le but étant de le trouver nous allons réécrire l'équation précédente afin de trouver a.
Le problème est que la matrice X n'est pas une matrice carrée, par conséquent son inverse n'existe pas. Nous passons alors par son pseudoïnverse (X^+). Cela revient à écrire l'équation suivante :

$$a = X^+ * y \tag{17}$$

Le passage par le pseudoïnverse est le "coeur" de notre problème et le but de notre programme. En effet, on essaye de voir à quel(s) moment(s) nous pouvons l'utiliser et de quelle manière. Tout d'abord nous utilisons la fonction svd (singular value decomposition) de Scilab qui prend en paramètre la matrice X. Elle renvoie alors 3 matrices : U, Delta et V que nous allons utiliser pour le calcul de X^+.

Il faut, au préalable, retoucher Delta et U en calculant respectivement leur pseudoïnverse $Delta^+$ et transposée U^T. On a ainsi l'équation suivante :

$$X^+ = V * Delta^+ * U^T \tag{18}$$

A présent, on peut aisément résoudre l'équation (17) en faisant le calcul matriciel de X^+ et de y.

4 Résultats

Dans cette partie, nous allons nous intéresser aux différents résultats que nous avons obtenus à travers nos fichiers Scilab dans le but d'en tirer des conclusions dans la partie suivante.

Nous avons souhaité faire un calcul de l'erreur commise à chaque ordre et pour chaque valeur de commande et voir ainsi plus tard leur influence sur l'efficacité et la précision des calculs. Ainsi nous obtenons six erreurs. Nous avons calculer l'erreur de la manière suivante :

$$Erreur = \frac{|ValeurPratique - ValeurThéorique|}{ValeurPratique} \tag{1}$$

Nous cherchons ici l'incertitude entre la valeur exacte et la valeur approchée. On souhaite déterminer et quantifier notre erreur de calcul

4.1 Premier ordre

Pour ce qui est du premier ordre nous obtenons les deux erreurs suivantes :

-u=1 : **Erreur** $= 8,6529 * 10^{-1}$

-u=5 : **Erreur** $= 2,1780 * 10^{-1}$

4.2 Deuxième ordre

Pour ce qui est du deuxième ordre nous obtenons les deux erreurs suivantes :

-u=1 : **Erreur** $= 7,0600 * 10^{-7}$

-u=5 : **Erreur** $= 1,4350 * 10^{-6}$

4.3 Troisième ordre

Pour ce qui est du troisième ordre nous obtenons les deux erreurs suivantes :

-u=1 : **Erreur** $= 1,5990 * 10^{-6}$

-u=5 : **Erreur** $= 4,0550 * 10^{-7}$

N.B. : Etrangement les différents fichiers ne nous renvoient pas la même choses s'ils sont lancés individuellement ou s'ils sont lancés à travers le fichier représentant le menu. Cette différence varie d'une puissance de 10 allant de 0 à 1. Nous avons essayé par tous les moyens de la résoudre mais nous n'y sommes pas arrivés. Nous arrivons cependant aux mêmes conclusions avec les deux jeux de valeurs.

5 Analyse

Dans cette partie, nous allons donc interpréter les différents résultats obtenus ci-dessus.

Intéressons-nous tout d'abord à l'impact de l'ordre sur l'identification. On remarque que pour le premier ordre on a une erreur beaucoup plus importante qu'aux autres ordres (qui est de l'ordre de 10^{-6}).

On peut tout simplement dire que nous ne pouvons prévoir avec exactitude la sortie à l'instant t+1 en se basant simplement aux entrée/sortie à l'instant t.

Par contre, on constate que l'erreur est sensiblement la même au deuxième et au troisième ordre. On peut ainsi en conclure qu'à partir de l'ordre 2 la qualité de prédiction est assez bonne. Ainsi il suffit de savoir toutes les actions des instants t et t+1 pour savoir la sortie à l'instant t+2.

Passons ensuite à l'effet de la commande sur la qualité de l'identification. On remarque qu'au premier ordre, la commande a un impact non négligeable sur l'erreur.

De plus, pour les deuxième et troisième ordre, la commande a un effet que l'on peut qualifier de "nul" sur l'erreur et donc sur la prévision de la sortie. Par conséquent on peut en conclure que la commande n'influe en rien sur la prédiction de l'évolution du système.

6 Conclusion

Pour conclure par rapport à ce TP, on en tire comme conclusion que l'identification par la méthode des moindres carrés est assez fiable.

En effet nous ne dépassons à aucun moment les 1% d'erreur. Cependant nous pouvons affirmer que cette méthode a un meilleur rapport précision / coût pour le modèle du deuxième ordre. En effet, on constate que l'ordre de l'erreur entre le premier et le deuxième ordre est de 10^{-6} alors qu'entre le deuxième et le troisième l'ordre est équivalent.

De plus les calculs obtenus par le deuxième ordre sont moins longs et demande ainsi moins de ressources à la machine.