Partie 01_TP de simulation sur PC

TP 1: Résolution des équations différentielles représentant les dynamiques des systèmes

I. *Objectif de TP*

Dans ce TP, on cherche à résoudre un système d'équation différentielles ordinaires (en Anglais: *Ordinary Differential Equations* ***ODE***) du premier ordre en utilisant le logiciel MatLab, dont la forme est la suivante:

$$\frac{dy_1}{dt} = f_1(y_1, y_2, y_3, \dots\dots\dots\dots y_n, \dots\dots\dots t)$$

$$\frac{dy_2}{dt} = f_2(y_1, y_2, y_3, \dots\dots\dots\dots y_n, \dots\dots\dots t)$$

$$\vdots$$

$$\vdots$$

$$\frac{dy_n}{dt} = f_n(y_1, y_2, y_3, \dots\dots\dots\dots y_n, \dots\dots\dots t)$$

$$y_1(0), y_2(0), \dots\dots\dots\dots y_n(0)$$

II. *L'utilisation de la commande "ode45" dans la résolution des équations différentielles*

La définition d'un tel système repose sur la définition de n fonctions, ces fonctions doivent être programmer dans une fonction "MatLab" sous la forme suivante:

$$Function\ dy = nom(t, y)$$

$$dy(1) = expression\ qui\ contient\ y(1), y(2) \dots\dots\dots y(n);$$

$$dy(2) = expression\ qui\ contient\ y(1), y(2) \dots\dots\dots y(n);$$

$$\vdots$$

$$\vdots$$

$$dy(n) = expression\ qui\ contient\ y(1), y(2) \dots\dots\dots y(n);$$

$$dy = d\acute{y};$$

où,

$\acute{y}$ est la matrice de transposition de la matrice y, elle sert à transformer le vecteur ligne y à un vecteur colonne.

On remarque que les $y(i)$ et les $dy(i)$ sont regroupées dans des vecteurs.

La dernière ligne est nécessaire car la fonction doit ressortir un vecteur colonne et même pas un vecteur ligne.

En suite pour résoudre cette équation différentielle, il faut appelé un solveur (Solver) et lui transmettre au minimum:

- Le nom de la fonction;
- Les pôles d'intégration (l'intervalle de calcule);
- Les conditions initiales.

Remarque

Le solveur fournit un vecteur colonne représentant les instants d'intégration t et une matrice dont:

- La ***1ère*** colonne représente les $\boldsymbol{y_1}$ calculées à ces instants;
- La ***2ème*** colonne représente les $\boldsymbol{y_2}$;
- La ***nième*** colonne représente les $\boldsymbol{y_n}$.

Sur le logiciel MatLab, l'appelle du solveur prends la forme générale suivante:

$$\gg [\boldsymbol{t}, \boldsymbol{y}] = \boldsymbol{ode45}\left('\boldsymbol{nom\ de\ la\ fonction}', [\boldsymbol{t_{initiale}}\ \ \boldsymbol{t_{finale}}], [\boldsymbol{y10}\ \ \boldsymbol{y20} \ldots\ldots\ \ \boldsymbol{yn0}]\right)$$

$$\gg \boldsymbol{y1} = \boldsymbol{y}(:, \boldsymbol{1});$$

$$\gg \boldsymbol{y2} = \boldsymbol{y}(:, \boldsymbol{2});$$
$\vdots$
$\vdots$
$$\gg \boldsymbol{yn} = \boldsymbol{y}(:, \boldsymbol{n});$$

ode45 est l'algorithme Rang-Kutta d'ordre 4 et 5.

III. Application

Soit qu'on a l'équation différentielle suivante:

$$\frac{d^2y}{dt} + 3.5\frac{dy}{dt} + 2.5y(t) = x(t)$$

Calculer théoriquement (sur papier) et par MatLab (en mode script) pour un intervalle de [0 5], $y(t)$ pour $x(t) = \sigma(t)$ et sachant que $y(0) = -1$ et $\frac{dy(0)}{dt} = -1.5$

$\sigma(t)$: échelon unitaire