TP N° 1 : Résolution des équations différentielles ordinaires dans Matlab

Partie 1 : Résolution des équations différentielles ordinaires par ODE45

On cherche à résoudre un système d'équations différentielles ordinaires de la forme (modèle d'état d'un système) :

$$\begin{cases} \dot{x}_1 = f_1(t, x_1, \ldots, x_n) \\ \vdots \\ \dot{x}_n = f_n(t, x_1, \ldots, x_n) \end{cases}, \text{ avec les conditions initiales : } x_1(0) = x_{10}, \ldots, x_n(0) = x_{n0}.$$

Rappelons que toute équation différentielle d'ordre supérieur ou FT peut s'écrire sous la forme d'état. Comment utiliser la fonction ODE45 pour résoudre des équations différentielles ordinaires ?

On crée une fonction Matlab (fichier fun.m dans notre cas) qui définit le système, la fonction a en entrée un scalaire t et un vecteur x (vecteur d'état) et en sortie un vecteur xp qui représente la dérivée.	La syntaxe du solveur Matlab ODE45 est donnée ci-dessous avec : - Intervalle d'intégration : $t_s = [t_0 \quad t_f]$ - Conditions initiales : $x_0 = [x_{10} \quad \ldots \quad x_{n0}]$
function $\quad xp = \text{fun}(t, x)$ $xp = zeros(n,1)$ $xp(1) =$ une expréssion de $x_1, \ldots, x_n$ et t $\vdots$ $xp(n) =$ une expréssion de $x_1, \ldots, x_n$ et t *end*	$[T, X] = \text{ode}45(@\,fun, t_s, x_0)$ - T : Vecteur contenant les instants auxquels la solution a été calculée. - X : Matrice de n colonnes où $x_1 = X(:,1), \ldots, x_n = X(:,n)$.

Remarque :

- Pour tracer $x_1(t)$ par exemple, utiliser : $plot(T, X(:,1))$
- Pour tracer une trajectoire dans le plan de phase $x_2 = g(x_1)$, utiliser : $plot(X(:,1), X(:,2))$
- Les noms des programmes et des fonctions MATLAB peuvent être changés.
- Spécification des instants d'évaluation de la solution : Si t_s contient plus de deux éléments, alors T est identique à t_s. Par exemple, si $t_s = t_0 : h : t_f$ avec h est le pas, alors T sera identique à t_s.
- Pour faire passer des paramètres supplémentaires à la fonction ODE45, on déclare une fonction Matlab de la forme : function $\quad xp = \text{fun}(t, x, m)$ (où m est un paramètre) et appeler ODE45 comme suit : $[T, X] = \text{ode}45(@(t,x)\, fun(t,x,m), t_s, x_0)$

Exemple : Intégrer les deux systèmes non linéaires suivants en utilisant ode45 :

1). $\begin{cases} \dot{x}_1 = 4x_1 - 2x_1 x_2 \\ \dot{x}_2 = x_1 x_2 - 3x_2 \end{cases}$, $x_1(0) = 2$, $x_2(0) = 3$, Tracer $x_1(t)$ et $x_2(t)$ en fonction du temps $(t \in [0, 25])$ et tracer $x_2(t)$ en fonction de $x_1(t)$.

2). $\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -x_1 + x_2(1 - x_1^2 - x_2^2) + (1 + 0.5\cos(x_1))u \end{cases}$, $x_1(0) = 0$, $x_2(0) = 1$, Tracer $x_1(t)$ et $x_2(t)$ en fonction du temps $(t \in [0, 25])$. Refaire la simulation pour $x_1(0) = 1$, $x_2(0) = 0$. On donne :

$$u = \frac{1}{1 + 0.5\cos(x_1)}\left(x_1 - x_2(1 - x_1^2 - x_2^2) - \sin(t) + 10(\cos(t) - x_2) + 25(\sin(t) - x_1)\right)$$

Partie 2 : Résolution des équations différentielles dans SIMULIK

Utiliser SIMULIK pour simuler les deux équations différentielles précédentes.