TP N° 1 : Résolution des équations différentielles dans Matlab

Partie 1 : Résolution des équations différentielles par ODE45

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

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

Rappelons que tout système ou équation différentielle d'ordre supérieur peut s'écrire sous cette forme (forme d'état).

Ces fonctions devront être programmées dans une fonction MATLAB (fichier fun.m dans notre cas) de la forme suivante :

function $xp = \text{fun}(t, x)$
$xp = zeros(n,1)$
$xp(1) =$ une expréssion de $x_1, \ldots, x_n$ et t
$\vdots$
$xp(n) =$ une expréssion de $x_1, \ldots, x_n$ et t

Ensuite, pour résoudre cette équation différentielle, il faut appeler le solveur ODE45 et lui transmettre au minimum : le nom de la fonction, les bornes d'intégration (t_0 et t_f), et les conditions initiales $x_0 = [x_{10} \quad \ldots \quad x_{n0}]$. La syntaxe de cette fonction Matlab ODE45 est donnée par :

$[T, X] = \text{ode}45\left(@fun, [t_0, t_f], x_0\right)$

T : Vecteur contenant les instants auxquels la solution a été calculée

X : Matrice de n colonnes où $x_1 = X(:,1), \ldots, x_n = X(:,n)$.

- 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 : $plot(X(:,1), X(:,2))$

Remarque : Les noms des programmes et des fonctions MATLAB peuvent être changés.

Intégrer les équations différentielles suivantes 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 ($t \in [0, 25]$). 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.

Partie 1 : L'oscillateur de Van der Pol et intégration temporelle

L'oscillateur de Van der Pol est un système non-linéaire du 2ème ordre décrit par l'équation :

$$\ddot{x} + m\left(x^2 - 1\right)\dot{x} + x = 0\,,\; m \in R$$

Donner la forme d'état de cet oscillateur en utilisant : $x_1 = x$ et $x_2 = \dot{x}$.

- Trouver les points d'équilibres et déterminer le type (la nature) de chaque point d'équilibre.
- Intégrer l'équation différentielle obtenue à l'aide de la fonction *ode45()* de Matlab ($t \in [0, 25]$).
- Pour chaque valeur de $m \in \{-0.5, 0, 0.5\}$, tracer l'évolution du système dans le plan de phase pour les conditions initiales $x(0) = (0.5, 0)$ et $x(0) = (3, 3)$.

Partie 2 : Méthode des isoclines

L'obtention du plan de phase d'un système peut aussi se faire à partir de la méthode des isoclines. Les isoclines sont l'ensemble des courbes où la pente des trajectoires est constante. On choisit alors une pente constante $a = \dfrac{dx_2}{dx_1}$ pour former avec l'équation de la dynamique $\begin{cases} \dot{x}_1 = f_1(x_1, x_2, m) \\ \dot{x}_2 = f_2(x_1, x_2, m) \end{cases}$, un système d'équations que l'on résout pour x_2. L'expression obtenue : $x_2 = h(x_1, a, m)$, donne l'isocline de pente a .

- Calculez l'expression analytique $x_2 = h(x_1, a, m)$, prendre $m = 0.5$.
- On choisit pour a les valeurs $a = [-4, -2, -1, -0.5, 0, 0.5, 1, 2, 4]$.
- Utiliser $x_1 = \text{linspace}(-2.5, 2.5, 30)$ pour obtenir un vecteur de 30 valeurs entre -2.5 et 2.5.
- Pour chaque valeur a et pour chaque valeur de x_1, calculer $x_2 = h(x_1, a, m)$, $c = \cos(\text{atan}(a))$, $s = \sin(\text{atan}(a))$. Puis tracer le segment de droite de pente a entre les points $(x_1 - d \times c, x_2 - d \times s)$ et $(x_1 + d \times c, x_2 + d \times s)$ en utilisant l'instruction : $plot([x_1 - d*c \quad x_1 + d*c], [x_2 - d*s \quad x_2 + d*s])$, $d = 0.1$ (segment de droite de centre (x_1, x_2), de pente a et de longueur $2d$). N'oublier pas d'utiliser l'instruction *hold on*, *axis([-2.5 2.5 -2.5 2.5])*.
- Pour chaque valeur de a , tracer sur la même figure l'isocline correspondante. Pour cela utiliser la commande de Matlab *fplot* (limiter l'axe x entre -2.5 et 2.5).
- Tracer sur la même figure la trajectoire de phase du système pour $m = 0.5$ et $x(0) = (0.5, 0)$.

Partie 3 : Méthode du graphe des pentes (champs des directions)

- Choisir $m = 0.5$.
- Utiliser $x_1 = \text{linspace}(-2.5, 2.5, 20)$ pour obtenir un vecteur de 20 valeurs entre -2.5 et 2.5.
- Utiliser $x_2 = \text{linspace}(-2.5, 2.5, 20)$ pour obtenir un vecteur de 20 valeurs entre -2.5 et 2.5.
- Pour chaque valeur de x_1 et x_2 calculer : f_1, f_2 et le facteur $b = \dfrac{d}{\sqrt{f_1^2 + f_2^2}}$ (le facteur b est introduit pour avoir des segments de droite (des pentes) de même longueur d). Puis, tracer le vecteur définit par les points (x_1, x_2) et $(x_1 + f_1 \times b, x_2 + f_2 \times b)$ avec $d = 0.2$.
  Utilizer: quiver(x1, x2, f1*b, f2*b,'MaxHeadSize',1)
- Tracer sur la même figure la trajectoire de phase du système pour $m = 0.5$ et $x(0) = (0.5, 0)$.