TP O.N.L. : Les moindres carrés

On s'intéresse à la réponse indicielle (*i.e.* à un échelon) d'un système dynamique du second ordre (circuit RLC ou système masse/ressort avec frottements visqueux). Un tel système peut être représenté par l'équation différentielle suivante :

$$\ddot{s} + a_1\dot{s} + a_2 s = ke(t),$$

où $ke(t) = H(t)$, H étant la fonction échelon ou fonction de Heaviside.

La sortie en fonction du temps du système $s(t)$ correspond alors à un mouvement oscillatoire amorti dont le fichier **donnees.mat** contient un échantillon de valeurs obtenues expérimentalement.

Note : *Le fichier donnees.mat est un tableaux à deux lignes : tps et sortie. Le vecteur tps contient les différents temps de mesure et sortie les valeurs obtenues à chaque mesure. Pour récupérer chacun d'eux, on écrira les commandes suivantes :*

— *data=load('donnees');*
— *tps=data.tps;*
— *sortie=data.sortie;*

Exercice 1. *Ecrire l'algorithme en pseudo langage d'une fonction permettant de calculer le polynôme de degrés* $\boldsymbol{n}$ *d'approximation au sens des moindres carrés du nuage de point* $\mathbf{xp} = (x_i)_{1\leqslant i\leqslant m}$ *et* $\mathbf{yp} = (f(x_i))_{1\leqslant i\leqslant m}$ *avec* $n < m$.

Exercice 2. *Ecrire une fonction Matlab d'entête*

function [coeff]=MoindresCarres(xp,yp,n)

qui retourne le vecteur ***coeff*** $= (a_n, \cdots, a_0)$ *contenant les coefficients du polynôme* $P(X) = a_n X^n + \cdots + a_0$ *de meilleure approximation (au sens des moindres carrés) du nuage de points donné par les vecteurs* $\mathbf{xp} = (x_i)_{1\leqslant i\leqslant m}$ *et* $\mathbf{yp} = (f(x_i))_{1\leqslant i\leqslant m}$ *avec* $\boldsymbol{n} < m$.

Exercice 3. *Ecrire un script Matlab qui :*

1. *charge le fichier des données expérimentales* ***donnees.mat****,*
2. *définit le degré du polynôme de meilleure approximation,*
3. *calcule les coefficients du polynôme de meilleure approximation par moindres carrés,*

4. évalue le polynôme en le vecteur $\boldsymbol{t}$ *représentant une subdivision régulière à* $N+1$ *points de l'intervalle de temps* $[0, 0.5]$ *(on utilisera pour cela la fonction matlab polyval),*

5. représente sur une même figure le nuage de points et le polynôme de meilleure approximation.

Faire des tests numériques en faisant varier le degrés du polynôme calculé et commenter vos résultats.