TD 1 : Interpolation et approximation au sens des moindres carrés

17/03/2020

L'objectif de ce TD est d'étudier les phénomènes d'interpolation et d'approximation à travers des exemples traités sur **MATLAB**®. Pour ce faire, vous devez ouvrir les fichiers *"TD_1_*.m"* et vous assurez que le workspace dans lequel vous travaillez est bien situé dans le même dossier que les fichiers.

N'hésitez pas à vous familiariser avec le langage en utilisant la commande *help* $[FUNC]$ (en remplaçant $[FUNC]$ par la fonction dont vous voulez vous servir).

1 Interpolation

1.1 Interpolation par polynômes de Lagrange

On mesure un signal périodique, sig, à des instants précis entre 0 et 1.2 s, x, séparer par des périodes $T_e = 0.06$ s. On aimerait connaître les valeurs de ce signal entre ces échantillons et l'on choisit une interpolation par polynômes de Lagrange sur une grille de 50 points.

Pour rappel, l'interpolation par polynômes de Lagrange permet de trouver une fonction polynomiale P passant par les points (x_i, y_i), $\forall i = 1 \dots I$ d'un signal échantillonné :

$$P(x) = \sum_{i=1}^{I} y_i \, L_i(x)$$

avec $L_i(x) = \prod\limits_{j=1 j\neq i}^{I} \frac{x-x_i}{x_j-x_i}$.

La fonction $LagrangeInterp$ contenu dans le fichier du même nom permet cette interpolation. Elle prend en argument la grille d'interpolation, les instants de mesures et les échantillons du signal.

1) Chargez les données du signal contenus dans le fichier *"sig.txt"*, à l'aide de la fonction *load*.

2) Grâce à la fonction *linspace*, créez la grille d'interpolation.
3) Remplissez la fonction $LagrangeInterp$. Compilez et lancez le script.

1.2 Interpolation pour la transformation d'image

On veut appliquer une rotation d'un angle ang à l'image *"saturn.png"*. On va donc vouloir changer les coordonnées "cartésiennes" de chaque pixels de l'image en coordonnées polaires. En formulation matricielle, soit un pixel p de coordonnées cartésiennes (i, j) :

$$p = \begin{bmatrix} i \\ j \end{bmatrix}$$

Les nouvelles coordonnées du pixels sont données par :

$$p_{rot} = R \cdot p$$

avec

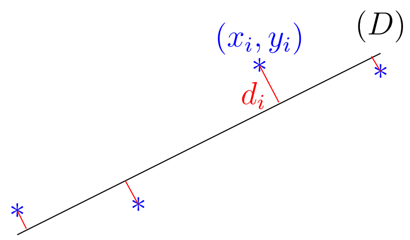
$$R = \begin{bmatrix} cos(ang) & -sin(ang) \\ sin(ang) & cos(ang) \end{bmatrix}$$

l'opérateur de rotation.

4) Remplissez les parties changement de base et rotation du code.
5) Qu'observez vous ?
6) Utilisez la fonction *imrotate* afin d'appliquer la rotation et l'interpolation bi-linéaire sur l'image.

2 Approximation au sens des moindres carrés

Les moindres carrés permettent de trouver la droite (D) réduisant les distances d_i entre chaque points de mesure (x_i, y_i) et celle-ci :



2.1 Régression linéaire

Le site de l'Institut National de la Statistique et des Études Économiques (Insee) propose de nombreuses données sur l'économie et la vie sociale française. Ainsi, on peut trouver le nombre de décès chaque année (voir figure 1). On cherche à trouver la meilleure droite représentant ces points.

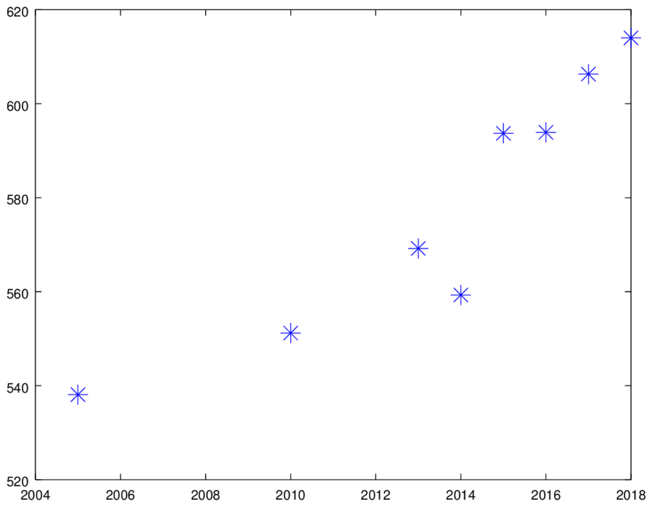


FIGURE 1 – Nombre de décès (en milliers) par années en France.

Dans un premier temps, on va exprimer la droite (d_1) passant par le premier et le dernier point des échantillons.

7) Remplissez la première partie du code "calcul de droite".

8) Calculez l'erreur d'approximation définie par :

$$e = \sum_{i=1}^{I} y_i - d_1(x_i)$$

La solution n'étant pas satisfaisante, on décide d'utiliser une approximation linéaire au sens des moindres carrés.

9) Remplissez la deuxième partie du code.

10) Calculez l'erreur d'approximation. Qu'observez-vous ?

2.2 Reconstruction d'un signal 1D

La modulation dans un signal périodique est une manière efficace de transporter de l'information. On retrouve cette fonction dans beaucoup d'applications en télécommunication, ainsi que dans d'autres domaines où elle sert à modéliser des phénomènes physique (ondes gravitationnelles).

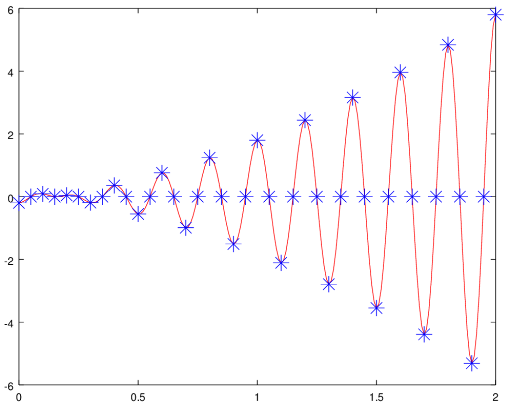


FIGURE 2 – Signal étudié ici. En rouge : y ; Points bleu : y_n ;

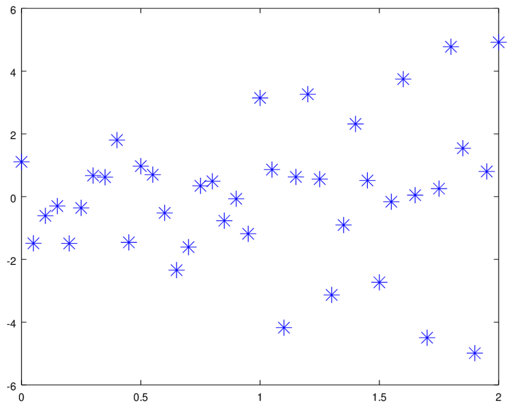


FIGURE 3 – Signal mesuré en sortie du capteur y_{bn}.

La figure 2 montre une partie d'un signal sinusoïdale, modulé en amplitude. On peut le modéliser par :

$$y(t) = a(t)\, cos(2\,\pi\, 5\, t)$$

$a(t)$ étant ici une fonction de modulation que l'on cherche à estimer et $t \in [0; 2]\ \ s$.

Ce signal est mesurer par un capteur (antenne) et est donc "échantillonné" (voir figure 2. En d'autres mots, on a accès au signal $y(t)$ seulement à certains instants $t = n\, T_e$:

$$y_n = a_n\, cos(2\,\pi\, 5\, n\, T_e)$$

avec $Te = 0.05\ s$

De plus, le capteur rajoute du bruit de lecture en mesurant le signal. Ainsi, un signal parasite est rajouté au signal échantillonné :

$$y_{bn} = y_n + b_n$$

avec b_n le bruit de lecture.

Au final, on obtient le signal représenté figure 3.

Comme on peut le voir, le bruit à ici dégradé suffisamment le signal pour qu'une méthode d'interpolation ne parvienne pas à trouver un modèle fiable du signal et donc de la modulation $a(t)$...

On décide donc d'utiliser une méthode d'approximation au sens des moindres carrés afin de trouver $a(t)$. On utilise une fonction polynomiale dans ce but :

$$p(t) = \sum_{i=0}^{I} a_i\, t^i$$

11) Écrivez la fonction "score" des moindres carrés.
12) Écrivez la version matricielle de cette fonction.
13) Déduisez-en l'expression vectorielle de a, avec $a^T = [a_1\ a_2\ \ldots a_I]$.
14) Complétez le code du fichier *"TD_1_4.m"*. Choisissez le degré du polynôme permettant la meilleure approximation.

Au final, vous devriez obtenir une bonne approximation du signal y, considérant le fait que le signal mesuré est bruité.