

# Compte rendu du projet Interface Homme-Machine sur MATLAB

Equipe vert

Réaliser par : Adam EL GHAZI, Zakaria AJLI

Encadrée Par : Regis FOURNIER

- I. Remerciements
- II. Introduction à notre projet
  - A. Présentation du groupe.
  - B. Présentation du projet
- III. Rappels théoriques
  - A. Rappel sur les matrices.
    - 1. Notations de matrices
    - 2. Multiplication de matrices
    - 3. Transposée d'une matrice.
    - 4. Déterminant d'une matrice.
    - 5. Inverse d'une matrice
  - B. Application aux systèmes d'équations linéaires
    - 1. Méthode de Gauss-Jordan
    - 2. Décomposition LU
    - 3. Décomposition QR
    - 4. Décomposition en valeurs singulières
    - 5. Méthode de Cholesky
  - C. Rappels statistiques.
    - 1. Loi normale
    - 2. Variance et écart-type
- IV. La théorie derrière notre projet.
  - A. Régression par la méthode des moindres carrés.
    - 1. Introduction des termes
    - 2. Minimisation de la fonction phi
  - B. Qualité de l'ajustement.
    - 1. Démonstration de l'équation de régression.
    - 2. Coefficient de détermination et coefficient de corrélation
    - 3. Variance expliquée et variance résiduelle
  - C. Formulation matricielle de la régression
    - 1. Passage du linéaire aux matrices.
    - 2. Calcul matriciel des paramètres de régression

- 3. Matrice de variance-covariance
- D. Formulation matricielle de la régression linéaire
- E. Régression linéaire multiple
- F. Interprétation probabilistique de la régression
  - 1. Introduction
  - 2. Test des résidus
  - 3. Test des paramètres
- V. Présentation de l'interface graphique homme- machine
  - A. Objectifs et fonctionnalités de l'IHM
  - B. Conception de l'interface
    - 1. Outil de développement
    - 2. Éléments de l'interface
    - 3. Architecture de l'IHM
- VI. Conclusion
  - A. Evaluation de la matière
  - B. Conclusion projet
  - C. Perspective d'amélioration
  - D. Glossaire

## I. Remerciements

Nous aimerions exprimer notre gratitude à notre professeur M. Fournier, pour son soutien, ses conseils et son expertise tout au long de ce projet. Nous avons énormément appris grâce à ses enseignements et sa présence encourageante. Nous ne pourrions pas avoir réussi sans son aide précieuse.

Nous sommes également reconnaissants de l'occasion que ce projet nous a offerte de travailler en équipe pendant deux mois. Nous avons appris à collaborer et à communiquer efficacement, ce qui nous sera très utile dans nos futures carrières. Nous sommes fiers du travail que nous avons accompli ensemble et nous espérons que cette expérience nous servira dans nos futures entreprises.

Nous avons travaillé ensemble en tant qu'équipe pour atteindre les objectifs de ce projet et chaque membre a apporté sa propre expertise et ses compétences uniques. C'est grâce à leur dévouement et leur travail acharné que nous avons été en mesure de réussir ce projet.

Enfin, nous souhaitons exprimer notre gratitude envers tous ceux qui ont contribué à la réalisation de ce projet. Nous sommes conscients que notre réussite est le fruit de la coopération de nombreuses personnes, et nous sommes très reconnaissants de tout le soutien que nous avons reçu.

## II. Introduction à notre projet

Notre projet de recherche s'intéresse à l'acquisition, au traitement et à la modélisation statique des données. Ce projet a été mené par un groupe composé de deux étudiants venant de formations différentes : un étudiant venant d'un DUT Réseaux et Télécoms et un autre étudiant venant d'une licence Informatique. Notre objectif était de concevoir une interface homme-machine (IHM) sur Matlab pour l'analyse des données.

Ce projet nous a permis d'approfondir nos connaissances en matière de statistiques et d'algorithmes de traitement de données. Nous avons également acquis de nouvelles compétences en programmation et en conception d'interfaces graphiques. Ce projet nous a ainsi offert une expérience pratique enrichissante dans notre parcours académique.

Dans ce rapport, nous présenterons les différentes étapes de notre projet, de l'analyse des données à la conception de l'IHM sur Matlab. Nous exposerons également les résultats de nos analyses et les conclusions auxquelles nous sommes parvenus.

## A. Présentation du groupe

Notre groupe de travail est composé de deux étudiants motivés et passionnés par les technologies de l'information. Venant DUT d'horizons différents, ils ont déjà acquis des connaissances en matière de réseaux informatiques, de traitement de signal et de programmation, mais aussi en algorithmique, en développement web et en base de données.

Le choix de former un groupe de travail composé d'étudiants de formations différentes a été fait en vue d'optimiser nos compétences et nos connaissances en matière de traitement de données. Nous avons travaillé en étroite collaboration, en mettant en commun nos compétences et en les adaptant aux différents aspects de notre projet. Notre collaboration a été fructueuse et nous sommes satisfaits des résultats obtenus.

## B. Présentation du projet

Le projet que nous avons entrepris porte sur l'interface homme-machine (IHM) pour le traitement et la modélisation des données sur MATLAB. Ce projet est très intéressant pour plusieurs raisons, notamment parce qu'il est en phase avec notre cursus académique, qui porte sous certains aspects sur l'acquisition, le traitement et la modélisation de données. De plus, l'utilisation de MATLAB est un aspect important de notre formation, et nous voulions approfondir notre connaissance de ce logiciel en explorant ses fonctionnalités relatives à l'IHM.

Notre projet consiste à concevoir une IHM intuitive et conviviale pour le traitement et la modélisation des données à l'aide de MATLAB. Nous voulons créer une interface qui permette à l'utilisateur de charger ses données, de les visualiser et de les analyser en utilisant différents outils et algorithmes proposés par MATLAB. Notre IHM devrait offrir une expérience utilisateur agréable et efficace, en présentant les résultats de manière claire et compréhensible.

Notre projet a également pour objectif de mettre en pratique les connaissances théoriques que nous avons acquises dans le cadre de notre cursus, en mettant l'accent sur les méthodes de régression par la méthode des moindres carrés.

Nous voulons appliquer ces méthodes à des jeux de données réels et examiner les résultats obtenus à l'aide de notre IHM.

### III. Rappels théoriques

#### A. Rappel sur les matrices

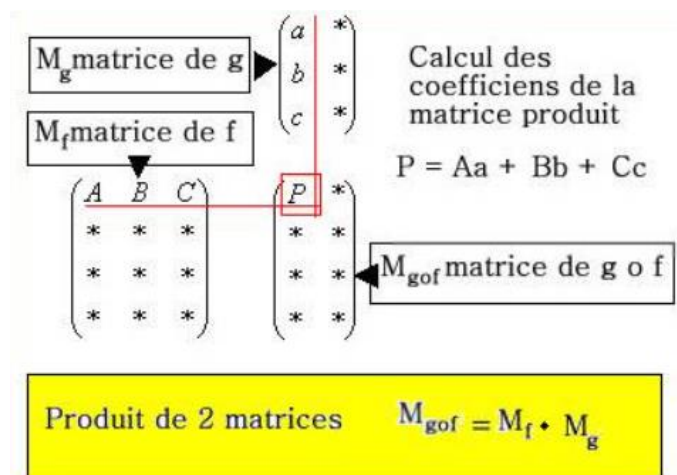
Les matrices sont des tableaux rectangulaires d'éléments. Les éléments d'une matrice sont appelés entrées ou coefficients. Les matrices sont souvent utilisées pour représenter des systèmes d'équations linéaires et des transformations linéaires. Les matrices peuvent être notées de différentes manières. La notation la plus courante est la notation matricielle, où une matrice  $A$  est représentée par des parenthèses autour des coefficients  $a_{ij}$ , comme suit :

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,j} & \dots & \dots & a_{1,p} \\ \vdots & \dots & \dots & \dots & \dots & \dots & \vdots \\ a_{i,1} & \dots & \dots & a_{i,j} & \dots & \dots & a_{i,p} \\ \vdots & \dots & \dots & \dots & \dots & \dots & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,j} & \dots & \dots & a_{n,p} \end{pmatrix}$$

i-ième ligne

j-ième colonne

Le produit de deux matrices  $A$  et  $B$  est défini si et seulement si le nombre de colonnes de  $A$  est égal au nombre de lignes de  $B$ . Le produit  $AB$  est alors défini comme suit :



Cela signifie que l'entrée  $ij$  de  $AB$  est obtenue en multipliant la  $i$ ème ligne de  $A$  par la  $j$ ème colonne de  $B$  et en sommant les résultats.

La transposée d'une matrice  $A$  est obtenue en échangeant les lignes et les colonnes. Elle est notée  $A^T$ .

Le déterminant d'une matrice carrée  $A$  est une quantité qui mesure la façon dont  $A$  modifie le volume de l'espace.

L'inverse d'une matrice carrée  $A$  est une matrice  $A^{-1}$  telle que  $AA^{-1} = Id$ , où  $I$  est la matrice identité.

## B. Application aux systèmes d'équations linéaires

### 1. Méthode de Gauss-Jordan

Il s'agit d'une méthode algorithmique pour résoudre les systèmes d'équations linéaires en utilisant des opérations élémentaires sur les lignes de la matrice augmentée correspondant au système. Ces opérations élémentaires incluent l'échange de deux lignes, la multiplication d'une ligne par une constante et l'addition d'un multiple d'une ligne à une autre ligne. La méthode de Gauss-Jordan permet d'obtenir une matrice échelonnée réduite qui peut être facilement utilisée pour déterminer les solutions du système.

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m \end{array} \right. \quad \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

### 2. Décomposition LU

Il s'agit d'une méthode de factorisation de matrices carrées en deux matrices triangulaires inférieure et supérieure ( $A = LU$ ). Une fois qu'une matrice est décomposée en matrices  $L$  et  $U$ , il est possible de résoudre des systèmes d'équations linéaires en utilisant la substitution avant et arrière pour résoudre les systèmes  $Lx = b$  et  $Ux = y$ . Cette méthode est souvent utilisée pour résoudre des systèmes d'équations linéaires avec des matrices de grande taille.



### 3. Décomposition QR

Il s'agit d'une méthode de factorisation de matrices carrées en une matrice orthogonale  $Q$  et une matrice triangulaire supérieure  $R$  ( $A = QR$ ). Cette méthode est souvent utilisée pour résoudre des problèmes d'optimisation et de régression linéaire.

$$(Q^t)^{-1} = Q$$

$$Ax = b$$

$$QRx=b$$

$$(Q^t)^{-1}Q^tRx=(Q^t)^{-1}b$$

### 4. Décomposition en valeurs singulières

Il s'agit d'une méthode de factorisation de matrices rectangulaires en trois matrices : une matrice orthogonale, une matrice diagonale et une matrice orthogonale. Cette méthode est souvent utilisée pour résoudre des problèmes d'approximation matricielle et de réduction de dimension.

### 5. Méthode de Cholesky

Il s'agit d'une méthode de factorisation de matrices symétriques définies positives en une matrice triangulaire inférieure  $L$  et sa transposée  $L^t$ . Cette méthode est souvent utilisée pour résoudre des systèmes d'équations linéaires avec des matrices symétriques définies positives.

$$Ax = b$$

$$A=L^t(L)$$

$A$  définie positive

$$\rightarrow \text{Pour tout } x \quad x^t Ax > 0$$

$$\det(L^t(L)) = \det(L) \cdot \det(L^t)$$

$$\det(L)^2 = \sum_{i=1}^n (L_{ii})^2$$

## C. Rappels statistiques

### 1. Loi normale

La loi normale, également appelée loi de Gauss, est une loi de probabilité continue qui est utilisée pour modéliser de nombreuses variables aléatoires dans divers domaines. Elle est caractérisée par deux paramètres, la moyenne  $\mu$  et l'écart-type  $\sigma$ , et est souvent notée  $N(\mu, \sigma)$ .

La densité de probabilité de la loi normale est donnée par la formule suivante :

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

### 2. Variance et écart-type

La variance d'une variable aléatoire  $X$  est une mesure de la dispersion de  $X$  par rapport à sa moyenne. Elle est définie comme suit :

$$\text{Var}(X) = E[(X - E[X])^2]$$

- $E[X]$  est la moyenne de  $X$ .

L'écart-type est simplement la racine carrée de la variance. Il est noté  $\sigma$ .

La transposée d'une matrice est une opération qui consiste à inverser les lignes et les colonnes de la matrice. Si  $A$  est une matrice de dimensions  $m \times n$ , alors la transposée de  $A$ , notée  $A^T$ , est une matrice de dimensions  $n \times m$ .

Le déterminant d'une matrice carrée est une valeur qui peut être calculée à partir des éléments de la matrice. Le déterminant est nul si et seulement si la matrice n'est pas inversible. Le calcul du déterminant dépend de la dimension de la matrice.

L'inverse d'une matrice carrée est une autre matrice carrée qui, lorsqu'elle est multipliée par la matrice originale, donne une matrice identité. L'inverse d'une matrice carrée ne peut être calculé que si son déterminant est non nul.

## IV. La théorie derrière notre projet

Notre projet consiste à utiliser la méthode des moindres carrés pour effectuer une régression linéaire multiple sur un jeu de données. Cette méthode permet de trouver une fonction linéaire qui modélise la relation entre une variable dépendante et plusieurs variables indépendantes. Cette fonction linéaire est déterminée en minimisant la somme des carrés des différences entre les valeurs observées et les valeurs prédites.

### A. Régression par la méthode des moindres carrés

La régression linéaire multiple par la méthode des moindres carrés consiste à trouver la meilleure droite qui minimise la distance entre les données et la droite. Cette méthode est couramment utilisée en statistique et en sciences de l'ingénieur pour prédire des valeurs de sortie à partir de plusieurs entrées. Nous allons expliquer les termes clés de la méthode des moindres carrés.

#### 1. Introduction des termes

Dans notre cas, nous avons un ensemble de données composé d'une variable dépendante  $Y$  et de plusieurs variables indépendantes  $X$ . Nous pouvons écrire notre modèle sous la forme :

Somme de  $i$  allant de 0 à  $n$  des  $(b_i * x^i)$

$$Y = b_0 + b_1x_1 + b_2x_2 + \dots + b_nx_n + \text{erreur}$$

- .  $Y$  est la variable dépendante
- .  $X_i$  sont les variables indépendantes
- .  $B_i$  sont les coefficients de régression

## 2. Minimisation de la fonction phi

L'objectif de la méthode des moindres carrés est de trouver les coefficients de régression  $b_0, b_1, b_2, \dots, b_n$  qui minimisent la somme des carrés des différences entre les valeurs observées et les valeurs prédites. Cette somme est appelée fonction phi et s'écrit comme suit :

Somme de  $k$  allant de 1 à  $n$  des  $(y_k - \hat{y}_k)^2$

$$\phi(b_0, b_1, \dots, b_n) = \sum_{i=1}^n (Y_i - (b_0 + b_1 X_{1i} + \dots + b_n X_{ni}))^2$$

$Y_i$  sont les valeurs observées de  $Y$

$X_i$  sont les valeurs observées des variables indépendantes

$n$  est le nombre de variables indépendantes

## B. Qualité de l'ajustement

Une fois que nous avons déterminé les coefficients de régression, nous pouvons évaluer la qualité de l'ajustement de notre modèle. Pour cela, nous utiliserons le coefficient de détermination et le coefficient de corrélation.

### 1. Démonstration de l'équation de régression

L'équation de régression nous permet de prédire la valeur de  $Y$  pour une combinaison donnée de variables indépendantes. L'équation s'écrit comme suit

Somme de  $i$  allant de 0 à  $n$  des  $(b_i * x^i)$

$$Y = b_0 + b_1 X_1 + b_2 X_2 + \dots + b_n X_n$$

### 2. Coefficient de détermination et coefficient de corrélation

Le coefficient de détermination  $R^2$  est une mesure de la proportion de la variance totale de la variable dépendante expliquée par les variables indépendantes. Il varie entre 0 et 1 et plus il est proche de 1, plus le modèle est précis. Le coefficient de corrélation  $r$  est une mesure de la force et de la direction de la relation linéaire entre la variable dépendante et les variables indépendantes.

### 3. Variance expliquée et variance résiduelle

La variance expliquée est la variance totale des valeurs de la variable dépendante qui peut être expliquée par la régression linéaire. En d'autres termes, c'est la variance de la moyenne prédite des variables dépendantes autour de la vraie moyenne de la variable dépendante. La variance expliquée est souvent exprimée sous forme de pourcentage, ce qui donne une idée de la quantité de variation dans la variable dépendante qui peut être expliquée par le modèle de régression.

$$V_e = (SS_e)/(p-1)$$

$$SS_e = \text{Somme de } k \text{ allant de } 1 \text{ à } n \text{ des } (\hat{y}_k - \bar{y})^2$$

P = Nombre de paramètre du modèle

La variance résiduelle, quant à elle, est la variance des résidus de la régression. Les résidus sont la différence entre les valeurs observées de la variable dépendante et les valeurs prédites par la régression linéaire. La variance résiduelle représente donc la quantité de variation dans la variable dépendante qui ne peut pas être expliquée par le modèle de régression.

$$V_e = (SS_r)/(n-p)$$

$$SS_e = \text{Somme de } k \text{ allant de } 1 \text{ à } n \text{ des } (y_k - \hat{y}_k)^2$$

P = Nombre de paramètre du modèle

N = nombre de mesure

## C. Formulation matricielle de la régression

### 1. Passage du linéaire aux matrices

: La formulation matricielle de la régression permet de traiter les données de manière plus efficace et plus rapide. Elle consiste à exprimer les variables dépendantes et indépendantes sous forme de matrices. Les variables indépendantes sont regroupées dans une matrice  $X$ , et la variable dépendante est regroupée dans un vecteur  $Y$ .

### 2. Calcul matriciel des paramètres de régression

Les paramètres de régression, à savoir les coefficients de la droite de régression, peuvent être calculés à l'aide de la méthode des moindres carrés. Cette méthode consiste à minimiser la somme des carrés des écarts entre les valeurs observées et les valeurs prédites.

### 3. Matrice de variance-covariance

La matrice de variance-covariance permet de mesurer la précision des estimations des paramètres de régression. Elle permet de calculer les intervalles de confiance et les tests d'hypothèses sur les paramètres. La matrice est donnée par la formule  $V = V_r * (A^{-1})$ , où  $V_r$  est la variance résiduelle.

## D. Formulation matricielle de la régression linéaire

Estimation par les moindres carrées :

La formulation matricielle de la régression linéaire est une façon de représenter la régression linéaire à l'aide de matrices. Le modèle de régression linéaire est représenté sous la forme  $Y = X\beta + \epsilon$ ,

$Y$  est le vecteur de variables dépendantes,  $X$  est la matrice de variables indépendantes,  $\beta$  est le vecteur des coefficients de régression à estimer, et  $\epsilon$  est le vecteur des erreurs aléatoires.

## E. Régression linéaire multiple

La régression linéaire multiple est une extension de la régression linéaire simple qui permet de modéliser la relation entre plusieurs variables indépendantes et une variable dépendante. Le modèle de régression linéaire multiple est représenté sous la forme  $Y = \text{Beta}(0) + \dots + \text{Beta}(p)x_p + \text{Erreur}$

Y est la variable dépendante,  $x_1, x_2, \dots, x_p$  sont les variables indépendantes,  $\text{Beta}(0), \dots, \text{Beta}(p)$  sont les coefficients de régression à estimer et  $\epsilon$  est l'erreur aléatoire au cours de la régression.

## F. Interprétation probabilistique de la régression

L'interprétation probabilistique de la régression linéaire est une approche qui utilise des concepts statistiques pour interpréter les résultats de la régression linéaire. Elle se base sur la distribution normale des erreurs aléatoires  $\epsilon$ . Les principales méthodes d'interprétation probabilistique sont les tests de résidus et les tests des paramètres.

Introduction :

L'interprétation probabilistique de la régression linéaire permet de vérifier si le modèle de régression linéaire est une bonne représentation des données et si les coefficients de régression sont significatifs.

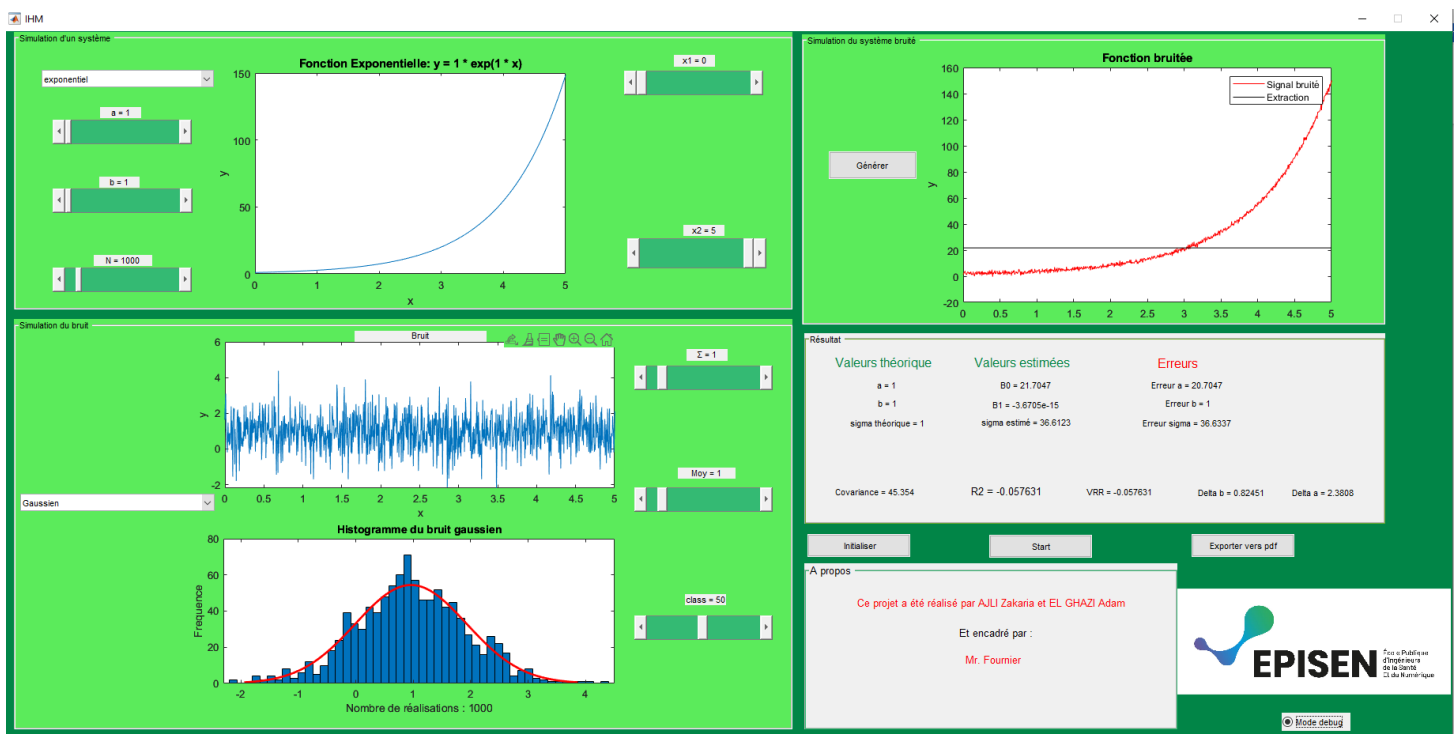
**Tests des résidus** : Les tests des résidus sont utilisés pour vérifier si les erreurs aléatoires  $\epsilon$  suivent une distribution normale et si elles sont indépendantes. Les tests les plus couramment utilisés sont les tests de normalité des résidus et les tests d'autocorrélation des résidus.

**Test des paramètres** : Les tests des paramètres permettent de vérifier si les coefficients de régression  $\beta$  sont significatifs. Les tests les plus couramment utilisés sont le test visant à déterminer une moyenne ou pour en comparer deux.

En utilisant les tests des résidus et des paramètres, nous pouvons vérifier si le modèle de régression linéaire est une bonne représentation des données et si les coefficients de régression sont significatifs.

# Partie technique projet Matlab

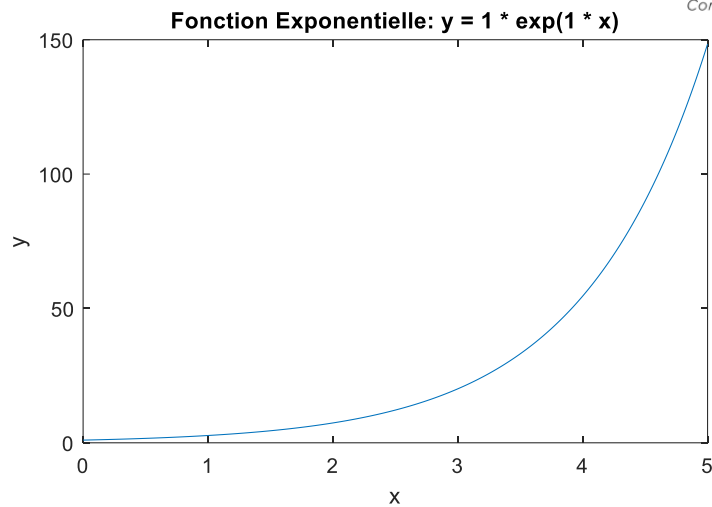
## V. Présentation de l'interface graphique homme-machine



Notre interface Comporte dans la première partie qui représente la simulation d'un système :

- Un premier slider qui nous permet de choisir le type de système.
- Un slider qui nous permet de choisir la valeur a qui représente le paramètre multiplicatif qui détermine l'amplitude de la courbe.
- Un slider qui nous permet de choisir b qui est un paramètre qui détermine le taux de croissance ou décroissance de la courbe.
- Un slider N qui nous permet de choisir le nombre de points.
- Deux slider x1 et x2 qui représentent les valeurs de xmin et xmax.
- Un axe où l'on pourra afficher notre résultat sous forme de courbe représentant le système choisi. Exemple :

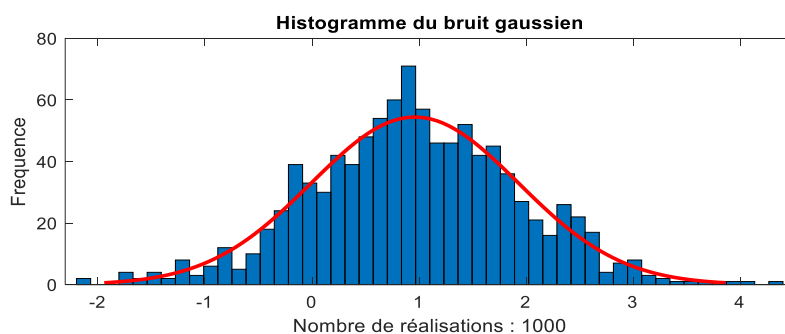


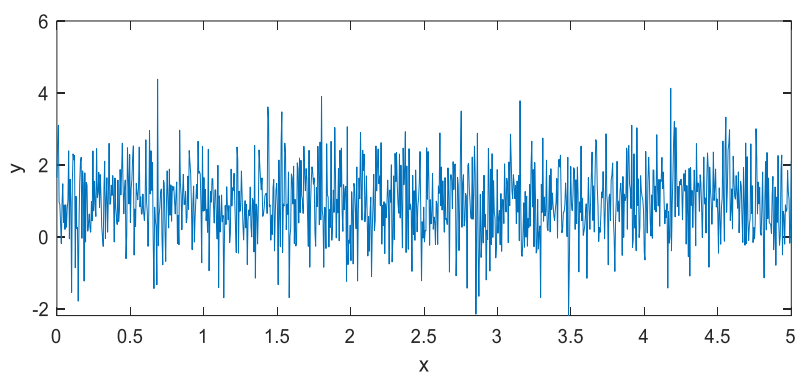


Dans la deuxième partie on a la simulation du bruit :

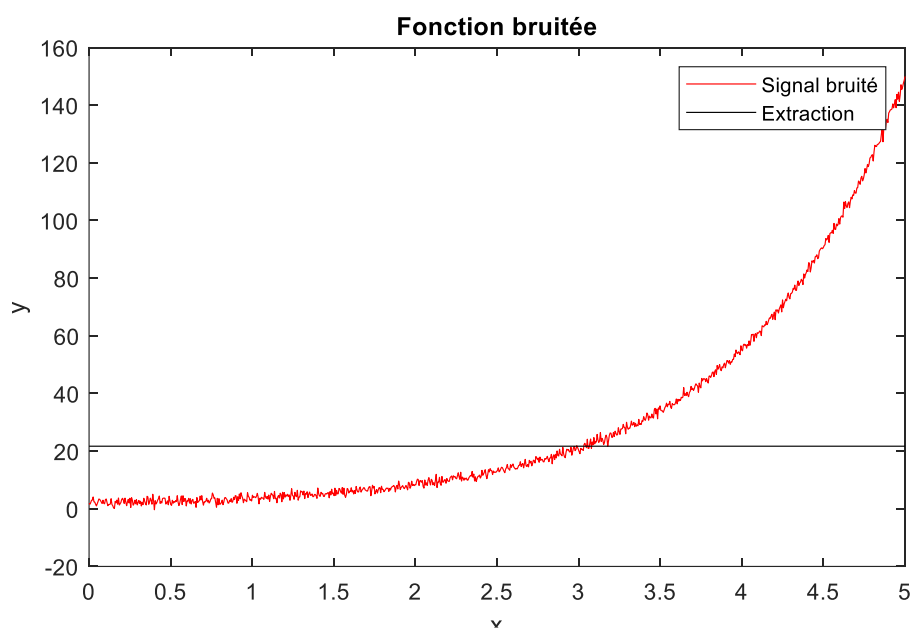
- Un popUp menu qui nous permet de choisir le type de bruit à afficher soit gaussien ou uniforme.
- Un slider sigma qui nous permet de choisir une valeur de l'écart-type du bruit ajouté à la fonction d'origine.
- Un slider pour choisir la moyenne.
- Un slider pour choisir le nombre de classe de l'histogramme.

Voici un exemple de la fonction exponentielle précédente représentant un bruit gaussien :





La 3<sup>ème</sup> partie contient la simulation du système bruité ainsi que la représentation de la fonction d'extraction qu'on peut afficher sur un axe en déclenchant le bouton générer. Exemple :



On a ensuite la partie résultat dans laquelle affichera les différents résultats tels que les valeurs théoriques, les valeurs estimées et les erreurs ainsi que le débogage en se basant sur la fonction d'extraction.



The screenshot shows a software interface with a green header and footer. The main content area is divided into sections. At the top, there's a 'Résultat' section with three columns: 'Valeurs théoriques', 'Valeurs estimées', and 'Erreurs'. Below these are buttons for 'Initialiser', 'Start', and 'Exporter vers pdf'. A 'A propos' section contains text about the project and the supervisor. A 'Mode debug' button is located at the bottom right.

Valeurs théoriques	Valeurs estimées	Erreurs
a = 1	B0 = 21.7047	Erreur a = 20.7047
b = 1	B1 = -3.6705e-15	Erreur b = 1
sigma théorique = 1	sigma estimé = 36.6123	Erreur sigma = 36.6337

Covariance = 45.354      R2 = -0.057631      VRR = -0.057631      Delta b = 0.82451      Delta a = 2.3808

Initialiser      Start      Exporter vers pdf

A propos

Ce projet a été réalisé par AJLI Zakaria et EL GHAZI Adam

Et encadré par :

Mr. Fournier

Mode debug

- Les valeurs b0 et b1 représentent les coefficients d'un modèle de régression linéaire.
- R2 représente le coefficient de détermination qui évolue la qualité de l'ajustement d'un modèle de régression linéaire.
- VRR représente la mesure de la qualité de l'ajustement du modèle.
- Le bouton radio debug nous permet d'afficher la cov, R2, VRR, Delta a, Delta b.

On a aussi ajouté 3 autres boutons :

- Initialiser qui donne met à jour la valeur des slider et des popup menu à des valeurs spécifiques données.
- Start qui permet de lancer la simulation.
- Exporter vers PDF qui permet de créer un fichier PDF qui vas nous permettre de stocker les valeurs des résultats obtenu par la fonction d'extraction.

## A. Objectifs et fonctionnalités de l'IHM

### Objectifs :

- L'objectif principal de cette IHM est d'analyser et d'extraire des données à partir de différentes fonctions mathématiques en ajoutant du bruit.
- Les utilisateurs peuvent visualiser les fonctions, le bruit et les résultats d'extraction dans des graphiques.
- Les utilisateurs peuvent également obtenir des valeurs théoriques, estimées et les erreurs associées, ainsi que d'autres paramètres tels que la covariance, VRR, Delta a et Delta b (en mode Debug).

### Fonctionnalités :

- Sélection d'une fonction mathématique (exponentielle, logarithmique, linéaire ou puissance) à partir d'un menu déroulant.
- Réglage des paramètres a et b à l'aide de curseurs.
- Sélection du type de bruit (gaussien ou uniforme) et réglage de la valeur de sigma à l'aide d'un curseur.
- Bouton pour démarrer le processus d'extraction des données.
- Affichage des graphiques des fonctions originales, des fonctions bruitées et des extractions.
- Affichage des valeurs théoriques et estimées des paramètres a et b, ainsi que de sigma.
- Affichage des erreurs pour a, b et sigma.
- En mode Debug, affichage de la covariance, VRR, Delta a et Delta b.
- Exportation des résultats dans un fichier PDF.
- Bouton d'initialisation pour réinitialiser l'IHM aux valeurs par défaut.
- Bouton radio pour activer/désactiver le mode Debug.

## B. Conception de l'interface

Pour la conception de l'interface de votre IHM, on a utilisé MATLAB et son environnement de développement intégré (IDE) avec l'outil GUIDE (Graphical User Interface Development Environment). GUIDE nous a permis de créer une interface utilisateur graphique en plaçant et en organisant des composants tels que des boutons, des sliders, des popupmenu, des axes pour les graphiques et des zones de texte.

**uicontrols** : Pour créer des éléments interactifs dans votre interface, tels que des boutons, des curseurs, des menus déroulants, des boutons radio et des zones de texte.

**uipanel**s Pour organiser et regrouper les éléments de l'interface utilisateur. Les uipanel facilitent la compréhension de l'interface en séparant visuellement les différentes fonctionnalités et options de l'application.

**Axes** Pour afficher les graphiques des fonctions originales, des fonctions bruitées et des extractions. Ces axes permettent à l'utilisateur de visualiser les résultats du processus d'extraction.

**Callback functions** : Pour définir les actions à effectuer lorsqu'un utilisateur interagit avec un élément de l'interface, par exemple en cliquant sur un bouton ou en modifiant la valeur d'un curseur. Les fonctions de rappel permettent de lier les actions de l'utilisateur aux calculs et aux mises à jour de l'interface.

## 1. Outil de développement

Pour développer une interface IHM GUIDE sur Matlab, nous avons utilisé l'outil GUIDE (Graphical User Interface Development Environment). GUIDE est un outil de développement graphique qui nous a permis de créer des interfaces utilisateur pour nos programmes Matlab. C'est un outil très intéressant car il fournit un environnement convivial pour la création d'interfaces avec des contrôles interactifs tels que des push button, des pop-ups menus, des sliders, etc.

## 2. Éléments de l'interface

- **Push button** pour lancer des actions spécifiques,
- **Radio button** pour permettre à l'utilisateur de faire des choix exclusifs,
- **sliders** pour permettre à l'utilisateur de régler une valeur numérique dans une plage donnée,
- **Popup menu** pour permettre à l'utilisateur de sélectionner une option à partir d'une liste prédéfinie,
- **Graphique** pour afficher des données et des informations visuelles à l'utilisateur.

## 3. Architecture de l'IHM

Nous avons pu personnalisés les différents éléments de l'interface homme-machine à notre code Matlab sous-jacent, de façon à personnaliser notre architecture avec les callbacks

## VI. Conclusion

### A. Évaluation de la matière

Le cours d'Acquisition, traitement et modélisation statistique des données ainsi que l'utilisation de l'interface homme-machine (IHM) MATLAB nous a permis d'acquérir des compétences fondamentales pour traiter et modéliser des données. Le contenu de ce cours nous a fourni une base solide pour comprendre les différents aspects de l'analyse statistique, tels que les méthodes d'échantillonnage, les mesures de dispersion et les distributions de probabilité. Nous avons également appris à utiliser MATLAB pour effectuer des analyses de données concrètes et visualiser les résultats.

En termes d'enseignement, le professeur a été très clair et précis dans la transmission des connaissances. Les exemples pratiques et les exercices en classe nous ont aidés à mieux comprendre les concepts théoriques. Nous avons également eu accès à des ressources logiciel en ligne pour nous aider à approfondir notre compréhension.

En ce qui concerne l'évaluation, les travaux pratiques et surtout la partie projet, ils ont été des occasions importantes pour mettre en pratique les connaissances acquises en classe. Ces travaux nous ont permis de travailler en équipe et de partager des idées pour résoudre les problèmes. Les projets ont également été une excellente occasion pour appliquer les compétences que nous avons acquises au cours de ce semestre.

En somme, nous sommes très reconnaissants pour l'opportunité de suivre ce cours. Nous sommes convaincus que les connaissances et les compétences que nous avons acquises nous seront très utiles dans notre future carrière professionnelle.

## B. Conclusion

Par le biais de ce projet, nous avons pu acquérir des connaissances nouvelles sur ce logiciel qu'est MatLab. Nous avons tous commencé avec des connaissances plus ou moins nulles sur ce logiciel et au cours de ces séances de travaux pratiques nous avons pu prendre en main petit à petit ce logiciel. Même si, nous ne sommes pas devenus de super experts du logiciel, nous avons tout de même atteint un niveau nous permettant de créer et développer une interface homme-machine.

Cette matière nous a permis de développer nos capacités à travailler en équipe, à respecter un cahier des charges, à réaliser de nouvelles tâches évolutives, ce qui se rapproche du monde professionnel.

De plus, ce rapport et un travail conséquent, ce qui permet et privilégie organisation, dynamisme et rigueur.

## C. Perceptive d'amélioration

Malgré les résultats positifs de notre projet, il est important de souligner les axes d'amélioration pour une version ultérieure.

Un page d'accueil, permettant à l'utilisateur d'avoir à entrer un nom d'utilisateur et un mot de passe pour une meilleur version optimal serait envisageable.

Ensuite, nous avons également remarqué que la documentation de notre projet pourrait être améliorée. En effet, certaines parties du code et certaines fonctionnalités peuvent ne pas être assez documenté, ce qui peut rendre moyen la compréhension et la maintenance de l'application à long terme. Pour remédier à cela, nous proposons de rédiger une documentation détaillée pour chaque fonctionnalité et de la maintenir à jour tout au long de notre formation.

Nous avons aussi ajouter un bouton start que nous avons pas eu le temps de configurer complètement que nous aimerions configurer prochainement le bouton permettra de lancer l'interface que lorsque tous les paramètres nécessaire ont été saisis afin d'avoir un bon résultat.

En résumé, nous avons identifié plusieurs points d'amélioration pour une version ultérieure de notre projet, notamment l'ajout d'un système de Page d'accueil d'identification, l'amélioration et le maintien d'une bonne



documentation. En prenant en compte ces éléments, nous sommes convaincus que nous pourrons fournir une application encore plus sécurisée, performante et plus conviviale pour les utilisateurs.

## D. Glossaire

`num2str()`: transforme un nombre en une chaîne de caractères.

`guide`: permet d'accéder aux anciens guides ou d'en créer des nouveaux qui ont pour but de créer et configurer une interface homme-machine.

`get()`: permet de récupérer des valeurs sur un slider par exemple

`set()`: permet de configurer ou de donner différents paramètres, valeurs à un paramètre ou variable précis

`disp()`: permet d'afficher dans la fenêtre de commande

`global variable`: rend la variable globale, afin de l'exécuter en dehors de la fonction, et l'utilisée dans sa fonction d'origine ainsi que dans la fonction où l'on veut la ré-utiliser

`axes(handle,graph)`: permet d'afficher un plot dans la fenêtre de graph sélectionnée

`plot(x,y,modèle)`: affiche le vecteur y en fonction du vecteur x selon le modèle choisi

`help fonction`: affiche un mode d'utilisation de la fonction souhaitée.

`function [sorties] = nom(paramètres)`: créer une fonction en définissant les paramètres ainsi que les sorties

`rand(n)`: renvoie un vecteur avec un nombre de lignes n, et un nombre de valeur par ligne v composé de valeurs aléatoires comprises entre 0 et 1 selon une distribution uniforme.

`hist(valeurs,class)`: affiche l'histogramme des valeurs

`randn`: même fonctionnement que `rand` mais avec une distribution normale.

`title('titre')`: renomme un plot.

xlabel - ylabel : renomme les axes d'une graphique.

subplot(a,b,c): découpe l'affichage en a lignes et b colonnes, et affiche la figure suivante en position c.

Int2str : transforme un entier en une chaîne de caractères