## Examen « Matlab »

## Durée : 3H (sur PC)

## 

## Seuls les documents fournis sont autorisés. La calculatrice collège est autorisée.

**DEPÔT DU LIVRABLE SUR MOODLE**

DEPOSER votre unique livrable au format .zip à votre **Nom\_prenom.zip** ou .rar avec 3 fichiers exo1.m exo2.m exo3.m + fichier data suivant le lien suivant :

<https://moodle-ingenieurs.cesi.fr/mod/assign/view.php?id=94400>

|  |  |
| --- | --- |
| **Compétences à valider** | Pondération |
| C1 - Savoir établir un dialogue avec l’utilisateur | x 1 |
| C2 – Savoir calculer une dérivée ou une intégrale | x 1 |
| C3 - Savoir construire un signal bruité | x 1 |
| C4 – Savoir réaliser un filtrage numérique et synthétiser la réponse du filtre | x 1 |
| C5 – Savoir concevoir une fonction Matlab | x 1 |
| C6 – Etre capable de retranscrire un algorithme en Matlab | x 2 |
| C7 – Qualité de l’affichage | x 1 |

Exercice 1 : Dérivées ou intégrales ?

Dans cette exercice, le programme doit :

* Demander à l’utilisateur :
  + Quelle fonction à étudier : y= x^2, y = 1/x ou y = ln(x) ?
  + Quelle opération à effectuer sur cette fonction : dérivation ou intégration ?
  + Une valeur x=xo où calculer la dérivée f’(xo) ou la primitive F(xo) (selon le choix précédent)
* Calculer f’(xo) ou F(xo)
* Afficher la valeur f’(xo) ou F(xo) sur le graphe

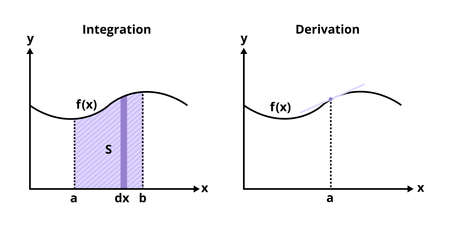


Figure 1 : Dérivation et Intégration (à titre illustratif)

Exercice 2 : Filtrage numérique

Cette exercice comporte 2 parties :

* Une première partie où il s’agit d’afficher le spectre du signal en affichant sur le graphe les fréquences du signal
* Une seconde partie où il s’agit de :

1. construire un signal bruité à partir d’un rapport signal à bruit (SNR)\* demandé à l’utilisateur
2. puis de filtrer le signal bruité par un filtre moyenneur d’ordre n\*\* (demandé à l’utilisateur) en fournissant au passage la réponse du filtre.

\* Formule écart-type en fonction du SNR :

A : amplitude du signal

SNR : Rapport signal à bruit

écart-type

\*\* Filtre moyenne d’ordre n :

y(k) = (1/n). [x(k)+x(k-1)+x(k-2) … +x(k-n)]

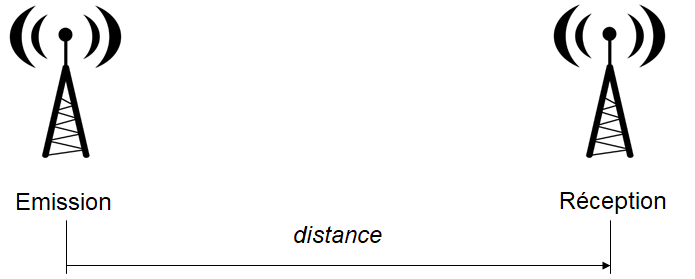
Pour constituer le vecteur B, on pourra utiliser la fonction ones :

* ones(1,n) donne un vecteur constitué de n valeurs de 1 🡺 [1 1 1 1 …. 1]

Exercice 3 : Algorithme de correcteur d’erreur

Dans cette exercice, il s’agit de réaliser un correcteur d’erreur qui fonctionne de la manière suivante :

* On souhaite transmettre un bloc de données [02 09 12] à travers un canal de transmission



* On va ajouter 2 mots de contrôle avant la transmission :
  + Le premier mot de contrôle est la somme des 3 données : 02+09+12=23
  + Le deuxième mot de contrôle est la somme pondérée des 3 données (chaque pondération correspond à la position (= rang) de la donnée dans le bloc):
  + 02 x 1 + 09 x 2 + 12 x 3 = 02+18+36 = 56

Ainsi le bloc final à envoyer est :

Tx = [02 09 12 23 56] ;

Supposons que du bruit survienne sur la ligne de transmission altérant une des données (ici 07 remplace 09), le bloc réellement transmis est le suivant  :

Tx\_err = [02 **07** 12 23 56]

L’objectif est de réaliser un correcteur d’erreur une fois le bloc reçu côté récepteur.

* Vous devez implémenter l’algorithme suivant permettant de retrouver le bloc d’origine :
  + On réalise tout d’abord la somme des 3 données transmises :
    - S\_err = 02 +07 +12 = 21
  + On fait la différence avec le premier mot de contrôle
    - Delta\_err = S\_err – mot de contrôle n°1

= 21 – 23

**Delta\_err = - 2**

* + On réalise ensuite la somme pondérée des 3 données transmises :
    - Sp\_err = 02 x 1 + 07 x 2 + 12 x 3 = 02+14+36 = 52
  + La formule suivante

**(Sp\_err - mot de contrôle n°2) / Delta\_err**

donne la position où la donnée a été altérée :

**Pos\_err = (52 – 56 ) /-2 = -4 / -2 = 2**

**L’altération de la donnée s’est produite à la deuxième position, c’est-à-dire au niveau du 07 : Tx\_err = [02 07 12 23 56]**

La vraie valeur est donc 07-**Delta\_err** = 07- (-2) = **09**

**La correction de l’erreur donne bien Tx = [02 09 12 23 56]**

**3 bloc de 3 données data1, data2, data3 sont disponibles dans le package pour tester et valider votre algorithme**

**Annexe : Algorithme complet (Wikipédia)**

