

Nom de l'étudiant-e : .....

\_\_\_\_\_ *Ce document complété doit être transmis au professeur à la fin du module* \_\_\_\_\_

## Feuille de travail pour l'atelier *Octave*

### Objectifs et exigences du module

Ce document contient la liste des objectifs devant être atteints par chaque étudiant-e à la fin du module (durée: deux fois 3 périodes). Les objectifs sont écrits sous la forme de "compétences" à acquérir. Il faut alors **être capable d'effectuer soi-même** ce qui est demandé, ou, dans certains cas, d'avoir simplement lu les paragraphes mentionnés. Une croix peut être mise dans la case correspondante à droite lorsque l'objectif a été atteint, en particulier lorsque **les exercices sont terminés et les programmes fonctionnent correctement**. La feuille de contrôle sera envoyée au professeur à la fin du module.

Tous les programmes écrits par l'étudiant-e doivent contenir au moins une ligne de commentaire avec le nom de l'auteur et la date de réalisation. Ils doivent finalement être placés dans un répertoire nommé `nom.prenom/`, qui sera finalement archivé et envoyé au professeur par courrier électronique. Pour archiver, on peut utiliser l'instruction Unix

```
> tar cfz nom.prenom.taz nom.prenom
```

ou l'instruction `zip`.

**Le module sera réussi lorsque le professeur aura reçu la feuille de contrôle complétée et le dossier archivé, et que le contenu du dossier sera conforme aux exigences données durant le cours.**

### 1 Utiliser *Octave* dans la ligne de commande

1. Les instructions mathématiques de base: lire la section 3, p. 3
2. Ecrire soi-même en *Octave* les codes de la page 3
3. Chercher des informations grâce à l'instruction `help` sur les constantes `e`, `i` et `pi`
4. Comprendre le rôle du séparateur point-virgule (`;`)

### 2 Fichiers scripts

1. La notion de script-file: lire la section 4, p. 3-4
2. Ecrire soi-même le code du script-file `triangle.m` et créer le script-file correspondant
3. Exécuter le script-file `triangle.m` dans *Octave*

### 3 Vecteurs et matrices

1. Vecteurs et matrices: lire la section 5, p. 4-5
2. Ecrire soi-même les codes de la page 5
3. Comprendre la différence entre un vecteur-ligne et un vecteur-colonne.  
Comprendre la signification de l'opérateur '
4. Effectuer le produit scalaire de deux vecteurs
5. Déclarer une matrice, par exemple de dimension  $3 \times 3$
6. Calculer le déterminant de la matrice
7. Multiplier une matrice avec un vecteur
8. Transformer un système d'équations linéaires en un système matriciel
9. Résoudre un système d'équations linéaires

### 4 Fichiers de fonctions

1. La notion de fonction-file: lire la section 6, p. 6
2. Ecrire soi-même le code du function-file `f01.m` et créer le function-file correspondant
3. Evaluer la fonction pour une valeur de la variable
4. Evaluer la fonction en une seule commande pour toute une liste d'arguments
5. Comment évaluer une fonction sans utiliser un function-file. Avantages, inconvénients ?

### 5 Graphiques 2D et 3D

1. Représentation graphique d'une fonction: lire la section 7, p. 6-7
2. Ecrire soi-même le code pour dessiner la fonction `f01.m`
3. Ecrire soi-même le code du function-file `f02.m` et créer le function-file correspondant
4. Ecrire soi-même le code pour dessiner la fonction à deux variables `f02.m`

### 6 Exercices

1. Exercice No 10.1, p. 13
2. Exercice No 10.2, p. 13
3. Exercice No 10.3, p. 13

## 7 Régression linéaire pour une résistance en fonction de la température

1. Lire et comprendre la description du problème dans la section 11.1
2. Aller chercher le fichier de données `copper.dat` dans la page web et le copier dans son répertoire de travail
3. Lire et comprendre le code permettant de lire et visualiser les données, puis le programmer soi-même
4. Utiliser l'instruction `polyfit()` pour déterminer les paramètres de la droite de régression
5. Ecrire le code permettant de produire un bon graphique, selon la section 11.4
6. Aller chercher le fichier-fonction `LinearRegression.m` dans la page web et le copier dans son répertoire de travail
7. Utiliser l'instruction `LinearRegression()` pour déterminer les paramètres de la droite de régression et les écarts-types correspondants
8. Lire le code correspondant à une parabole de régression, et comprendre le résultat produit par ce code, selon section 11.6

## 8 Régression appliquée à un problème de conduction de chaleur

1. Lire et comprendre la description du problème dans la section 12.1
2. Aller chercher le fichier de données `Messdaten.prn` et le fichier-script `ReadHeatData.m` dans la page web et les copier dans son répertoire de travail
3. Lire et comprendre le code permettant de lire les données, selon section 12.2
4. Comprendre l'algorithme pour la préparation des données et écrire le code correspondant, selon section 12.3
5. Exécuter la régression linéaire et visualiser les résultats, selon section 12.4

\_\_\_\_\_ *Ce document complété doit être transmis au professeur à la fin du module* \_\_\_\_\_

Date: ..... Signature: .....