TP Matlab/Simulink - Introduction

Exercice 1:

Ecrire un programme qui demande deux réels a et b à l'utilisateur, et qui échange les valeurs de a et b si a > b.
Vérifier votre programme en faisant afficher les valeurs finales de a et b, dans les différents cas.

Exercice 2:

a) Ecrire 3 scripts permettant de trouver le minimum de 3 nombres, le maximum de 3 nombres, la valeur médiane de trois nombres. On utilisera les structures **`if`**…**`else`**…**`end`** ou/et **`if`**…**`elsif`**…**`else`**…**`end`**.

b) Créer un script utilisant la commande **`menu`** pour laisser le choix entre les 3 possibles calculs (minimum, maximum, médiane) à l'utilisateur.

Exercice 3:

Le niveau de pollution à l'ozone dans l'atmosphère est calculé de la façon suivante :
Le niveau 1 est atteint à partir d'une concentration en ozone de 140 unités ($\mu g/m^3$), le niveau 2 à partir de 180 unités et le niveau 3 à partir de 220 unités.
Ecrire un programme qui demande la concentration d'ozone dans l'air et qui donne le niveau de pollution.
Petite précision : l'utilisateur doit voir l'affichage " Aujourd'hui, niveau de pollution 0 (resp. 1, 2, ou 3)".

Exercice 4:

Dans chacun des cas suivants, écrire un programme demandant n à l'utilisateur et affichant u_n :

1. $u_0 = 0.1$ et $\forall n \in$ N, $u_{n+1} = 1 - e^{-2*u_n}$ (pour vérifier $u_5 = 0.6976037$).
2. $u_1 = 1$ et $\forall n \in$ N*, $u_{n+1} = 2nu_n + 3$ (pour vérifier $u_7 = 135759$).

Modifier les programmes précédents pour en faire des fonctions.

Exercice 5:

La résistance *R* d'un conducteur électrique est directement proportionnelle à sa longueur *L*, et inversement proportionnelle à sa surface de section *A* : *R* = *P*. *L* /*A*

On appelle résistivité le facteur de proportionnalité *P*. La résistivité du cuivre à 20°C est de

P=0.0170…0.0178 $\Omega\,\mathrm{mm}^2/\mathrm{m}$. La résistivité est fonction de la température et l'on peut écrire la variation par :

$$P_\theta = P\left(1 + a\ \Delta\theta\right)$$

où : P_θ est la résistivité à θ°C,

a un coefficient de température (pour le cuivre a =0.0039 K^{-1}),

$\Delta\ \theta$ est la différence de température au-delà de 20°C.

Wieseman (1989) donne une relation plus détaillée :

$$P_\theta = P_{20}\left(1 + a_{20}\ \Delta\theta + b_{20}\ \Delta\theta^2\right)$$

où : $P_{20} = 0.017\ \Omega\,mm^2/m$
$a_{20} = 4.3 \times 10^{-3}\ K^{-1}$
$b_{20} = 0.6 \times 10^{-6}\ K^{-2}$
Pour comparer graphiquement les deux relations, entre 20°C et 100°C, nous allons les tracer.

- Créez deux constantes *P* et *a*.
- Créez un tableau de températures theta allant de 20 à 100°C avec un pas de 0.5°C.
- Calculez delta ($\Delta\theta$), puis *P*_theta, selon la première formule donnant P_θ.
- Faire de même avec la formule de Weiseman.
- Tracer les deux résultats sur le même graphe.
- Mettez les labels nécessaires aux axes avec les commandes xlabel, ylabel.
- Pour mettre une légende aux courbes, on utilise la commande legend.

Exercice 6:

Ecrire un script qui déclare la variable *R* contenant la valeur 20. Déclarer 3 variables *D*, *P* et *S* et affecter respectivement à ces variables les valeurs du diamètre, du périmètre et de la surface d'un cercle dont le rayon est *R*. On affichera à l'écran le contenu de ces différentes variables selon le format suivant :

Un cercle de rayon **WW** a pour diamètre **XX**, pour circonférence **YY** et pour surface **ZZ**.

Exercice 7:

Les équations paramétriques d'une ellipse centrée à l'origine des coordonnées, le grand axe 2A et le petit axe 2B sont :

```
x = A cos t
y = B sin t
```

où 0 < t < 2 pi

- Ecrire un script qui permet de tracer une ellipse avec, par exemple, les paramètres A = 2 et B = 1.
- Essayez de comprendre le sens de A et B.

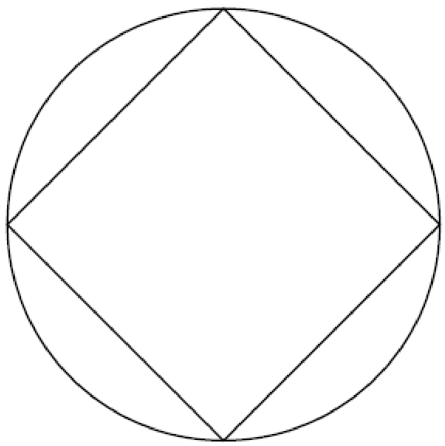
Exercice 8:

Nous cherchons un algorithme pour estimer la valeur de π. Une méthode est basée sur le fait que le périmètre d'un cercle de rayon ½ est π. Pour estimer la valeur de π, il suffit alors d'estimer le périmètre d'un cercle de rayon ½ . L'idée est donc d'inscrire des polygones réguliers dans le cercle et de calculer le périmètre du polygone.

Par exemple, si on inscrit un carré, on vérifie facilement que le périmètre est $2\sqrt{2}$, ce qui est une approximation grossière de π (cf. figure 1). En augmentant le nombre de côtés du polygone, on s'approche de plus en plus du périmètre du cercle et donc
de π. Nous noterons p_n le périmètre du polygone ayant 2^n côtés (par exemple, $p_2 = 2\sqrt{2}$) et on assumera que la formule de récurrence suivante est vraie:

$$\begin{cases} r_{n+1} = \dfrac{r_n}{2+\sqrt{4-r_n}}; \\ p_{n+1} = 2^n\sqrt{r_{n+1}}, \end{cases} \qquad \text{pour} \quad n = 3, 4, \ldots$$

où $r_3 = \frac{2}{2+\sqrt{2}}$.



Écrire un fichier-m qui calculera p_n pour n = 4, 5, ..., 30 en utilisant cet algorithme.
Imprimer, sous forme d'un tableau (utiliser la commande fprintf) et avec toute la précision disponible, les valeurs de n, p_n ainsi que l'erreur absolue commise.

Exercice 9: Simulink

Un système du 2ème ordre est régi par l'équation différentielle linéaire :

$$\frac{d^2y}{dt^2} = 5\left(u + 2\frac{dy}{dt} - y\right)$$

1) A l'aide de Simulink, programmer le système représenté par l'équation ci-dessus.
2) Compléter le schéma Simulink pour pouvoir visualiser la réponse du système à un échelon en entrée u.