

## TD 1-1 Découverte Matlab et conditionnelles

(\* = exercices les plus faciles)

### 1 Compréhension d'algorithme

#### 1.1 Décryptage d'algorithme

Que fait le programme Matlab suivant ? Pour vous en convaincre, choisissez des valeurs initiales pour  $x$  et  $y$  et suivez leur évolution au fur et à mesure des calculs effectués.

```
x = input('valeur de x' ?); % lecture au clavier de la valeur de la variable x
y = input('valeur de y' ?);
x = x + y ;
y = x - y ;
x = x - y ;
disp([x,y]) % écriture à l'écran des valeurs de x et y
```

#### 1.2 Conversion \*

Ecrire dans la fenêtre de commandes les instructions Matlab qui étant donné un temps  $t$  exprimé en secondes, le convertissent en heures, minutes et secondes.

#### 1.3 Charité bien ordonnée \*

A, B et C possèdent respectivement 5, 4 et 3 millions d'euros. Chaque jour :

- A, qui a des loisirs, va porter à B la moitié de sa fortune
- B, qui ne veut pas être en reste, va porter à C la moitié de sa (nouvelle) fortune
- puis C, suivant leur exemple se précipite chez A pour lui donner la moitié de sa (nouvelle) fortune.

Ecrire les instructions qui permettent de connaître les fortunes respectives de A, B et C le soir venu.

## 1.4 De l'importance de l'indentation et des mots clefs

Soit le programme Matlab incomplet suivant :

```
x = 1 ; y = 2 ; z = 5 ; t = 6 ; i = 10 ;
C1 = x < y ;
C2 = z == t ;
if C1
i = i + 2 ;
if C2
i = i - 2 ;
else
i = 0 ;
```

- Quel *type* de valeur est stocké dans la variable C1 ?
- Rappelez le sens différent des opérateurs = et == en Matlab
- En déduire ce que réalise cette ligne et quel est le type de la variable C2
- quel mot clef manque par deux fois dans ce programme Matlab ?
- il existe plusieurs façons de placer les deux occurrences manquantes dans le programme. Proposez deux façons différentes de faire, et suivant chacune indiquez la valeur de la variable i à la fin du programme.
- Pour chaque variante, proposez une version correctement indentée.

## 1.5 Intersection de conditionnelles : ça va chauffer !

Revenez sur les règles vues en cours pour déterminer si un matériau est jugé satisfaisant ou non-satisfaisant en regard de ses propriétés de conduction thermique mesurées. Rappel :

1. la donnée est lmes1,lmes2,lmes3,lmes4 des conductivités thermiques mesurées, lmod1,lmod2,lmod3,lmod4 des conductivités modélisées.
2. On calcule B1,B2,B3,B4 puis S
3. en comparant S à 0.03 et les Bi à 0.06 on prend la décision

Implémentez ces calculs et ces règles de décision et essayez-les à partir de quelques valeurs tests.

Trouvez un jeu de valeurs pour lesquelles on conclut que le test est satisfaisant, et un cas où le test n'est pas satisfaisant.

## 1.6 Problème financier

Supposons que vos économies soient abritées par une banque ayant la politique d'intérêts suivante : si vous disposez de moins de 5000€ en moyenne sur votre compte dans l'année, la banque vous versera

en fin d'année un intérêt de 9%. Par contre, si vous disposez de plus de ce montant, la banque vous applique un taux d'intérêt à 12%.

On veut écrire des instructions Matlab qui vous permettent de savoir quel argent vous allez gagner grâce aux intérêts de votre compte.

1. Combien de variables sont nécessaires pour modéliser ce problème ?
2. Dans la **fenêtre de commande** trouvez une façon de stocker en mémoire la somme d'argent que vous avez laissé sur votre compte cette année.
3. Entrez les instructions permettant à Matlab de calculer le montant de votre intérêt (cela doit fonctionner quelle que soit la somme précédemment indiquée, c-a-d quel que soit le cas de figure).
4. Vérifiez en changeant la somme initiale que Matlab est bien capable de tenir compte des différents cas.
5. Supposons maintenant que votre banque n'applique le taux d'intérêt de 12% que pour les sommes entre 5000€ et 10000€ , et qu'au delà de 10000€ un taux de 12% est appliqué. Indiquez la suite d'instructions qui permet à Matlab de calculer le montant de votre intérêt en fin d'année en fonction de la somme que vous avez sur votre compte.
6. Par défaut cet intérêt est déposé sur votre compte. Quelles instructions faut-il ajouter si vous voulez connaître le montant d'argent disponible sur votre compte après versement des intérêts ?
7. Reportez ci-dessous les instructions une fois qu'elles sont vérifiées.

Calcul d'intérêts :

Un de vos amis propose le code Matlab suivant pour la partie centrale du calcul d'intérêt.

```
if montant < 5000
    taux = 0.09;
```

```
end
if montant < 10000
    taux = 0.12;
end
if montant >= 10000
    taux = 0.15;
end
```

8. Combien de conditionnelles ce code contient-il ? est-ce plus ou moins ce dont vous avez eu besoin précédemment ?
9. Pouvez-vous, sur la base d'un exemple, lui expliquer en quoi son code est incorrect ?
10. Indépendamment du fait que ce code soit correct ou incorrect, quel désavantage a-t-il d'un point de vue opérationnel ?

## 1.7 Racine(s) d'un polynôme du deuxième degré

### 1.7.1 Premiers essais

(15mn)

Dans la fenêtre de commandes, définissez trois variables  $a$ ,  $b$  et  $c$  correspondant aux coefficients d'un polynôme du 2ème degré, par exemple  $2x^2 - x - 1$ .

Tapez ensuite les commandes nécessaires pour calculer les racines de ce polynôme par la méthode du déterminant (on supposera pour l'instant que le polynôme admet deux racines). Rappel : la fonction `sqrt` en Matlab permet de calculer une racine carrée.

## 1.8 Conditionnelles

(20mn)

Nous allons maintenant adapter votre code à tous les cas possibles. Pour cela, nous allons avoir recours aux instructions conditionnelles.

Effectuez les ajustements nécessaires dans votre script pour qu'il distingue les cas où il n'y a pas de solution, de ceux où la solution est indéterminée, du cas où il n'y a qu'une racine, du cas général où deux racines existent. Dans tous ces cas, le programme doit avoir une sortie satisfaisante pour l'utilisateur<sup>1</sup>.

Pour tester votre programme vous pourrez vous appuyer sur les polynômes suivants :

- $2x^2 - 3$
- $3x - 4$
- $x^2 - x + 3$
- $2x^2 - x - 1$

---

1. il faut savoir troquer la casquette "concepteur" pour la casquette "utilisateur" quand c'est nécessaire.

## 1.9 Classement de trois valeurs

Un algorithme réalisé par l'un de vos partenaires a besoin d'une petite routine simple qui ordonne trois nombres distincts  $x, y$  et  $z$ , c-a-d les classe en  $min, med$  et  $max$ .

1. Ecrivez un algorithme qui permet de réaliser ce traitement (sans utiliser une fonction de tri.
2. Au fait, combien de possibilités d'ordres entre les 3 valeurs de départ ?
3. combien de comparaisons contient votre algorithme ?
4. combien seront effectuées dans le meilleur des cas ? et dans le pire des cas ?
5. Votre algorithme est-il optimal en nombre de comparaisons effectuées (une instruction qui peut être coûteuse sur des valeurs de  $x, y, z$  un peu extrêmes) ? Si ce n'est pas le cas, améliorez-le ; si c'est le cas, comment s'en convaincre (sur la base d'un parallèle avec la recherche dichotomique appliquée ici sur les ordres possibles).
6. Dans le cas où l'on dispose de 6 valeurs à trier, combien de conditions seront nécessaires au minimum ?

**Remarque :** SOUS MATLAB VOUS POURREZ DEMANDER LA REPRÉSENTATION DE LA FONCTION INDICANT LE NOMBRE MAXIMUM DE COMPARAISONS NÉCESSAIRES AFIN DE VOIR QUELLE EST SA CROISSANCE QUAND LE NOMBRE DE VALEURS À CLASSER AUGMENTE

Proposez maintenant une variante utilisation la fonction `sort` de Matlab (réflexe : `help sort` ;)

## 2 Un algorithme *glouton* pour rendre la monnaie

Pour un prototype de robot distributeur de boissons dans les cinémas, on vous demande d'écrire un algorithme permettant de rendre la monnaie. Cet algorithme prend en entrée un montant payé par l'utilisateur (on supposera au maximum 5€ et un prix qu'il avait à payer, au maximum aussi 5€). On supposera pour l'instant que le distributeur dispose de larges réserves de pièces de chaque type pour rendre la monnaie. On vous demande de vous occuper de la programmation de ce distributeur pour qu'il rende la monnaie de la façon la plus arrangeante pour le consommateur, c-a-d *en le nombre minimum de pièces*. Votre distributeur dispose de pièces de 2 et 1 euros, de pièces de 50c, 20c, 10c, 5c, 2c et 1c. Attention, votre prototype de programme est écrit en matlab, mais le circuit électronique à réaliser ne disposera pas d'itératives (boucles), il faut donc écrire votre prototype sans (mais vous pourrez utiliser les fonctions division entière (`fix(x, y)`) et reste (`rem(x, y)`)).

### Version 1

1. De combien de variables "données" avez-vous besoin pour traiter ce problème ?
2. Ecrivez en quelques phrases le principe d'un algorithme possible pour réaliser le rendu de monnaie (attention à respecter la contrainte).

3. Vérifiez manuellement que cela fonctionne pour un prix de 2€32c et un montant payé de 5€.
4. Le résultat de votre algorithme est multiple (c-a-d, ce n'est pas une seule valeur isolée). Combien de variables allez-vous choisir comme variables "résultat" ? Quel va être le type de ces variables ?
5. Ecrivez maintenant un algorithme en pseudo-code pour le distributeur
6. Proposez une **trace** de votre algorithme pour un prix de 1€30c et un montant payé de 3€99. Votre algorithme fournit-il la meilleure solution ?

## Version 2

Pour se rapprocher de la réalité, on vous demande maintenant de prendre en compte que le distributeur (à un moment donné) dispose d'un nombre limité de pièces de chaque type. En plus, il s'agit d'un modèle avec trieur : les pièces introduites par le client sont automatiquement déposées dans le compartiment des pièces du même type, c-a-d peuvent elles-aussi servir à rendre la monnaie.

7. Décrivez très précisément comment les "données" de votre algorithme en sont affectées.
8. Modifiez votre algorithme pour tenir compte de ces facteurs.
9. Proposez une version Matlab de votre algorithme.

**Remarque :** LE PRINCIPE MIS EN OEUVRE ICI POUR CHOISIR QUELLES PIÈCES ON REND AU CLIENT EST APPELÉ **GLOUTON** : ON OPTIMISE AUTANT QUE POSSIBLE À CHAQUE ÉTAPE EN ESPÉRANT ABOUTIR À UNE SOLUTION OPTIMALE AU BOUT DES ÉTAPES EFFECTUÉES. CE PRINCIPE MARCHE POUR CERTAINS PROBLÈMES D'OPTIMISATION MAIS PAS POUR TOUS.<sup>2</sup>. MÊME DANS LES CAS OÙ IL NE DONNE PAS LA MEILLEURE SOLUTION POSSIBLE, IL EST EMPLOYÉ POUR EN OBTENIR UNE APPROXIMATION, CAR IL EST RAPIDE À METTRE EN OEUVRE ET EFFICACE EN TEMPS CALCUL GÉNÉRALEMENT. CONCLUSION : RETENEZ BIEN CE PRINCIPE D'ALGORITHME "GLOUTON".

## Références

- [1] Biran A, Breiner M, *Matlab pour l'ingénieur*, 2ème édition, Pearson Education, 2009.
- [2] Lapresté, J.-T., *Introduction à Matlab*, 3ème édition, Ellipses, 2009.

---

2. Par exemple dans le cas du pbm considéré ici, ce principe marche bien pour notre système monétaire, mais pas pour certains autres systèmes ou la valeur des pièces est différentes