TP 4 : Révisions

Dans ce TP, nous vous donnerons quelques exercices de révision pour Matlab, afin de vous préparer au mieux pour le contrôle.

1 Exercices de révision

Le jeu des bâtonnets

Le jeu des bâtonnets a été popularisé par l'émission de télé Fort Boyard. Dans cette émission, où les candidats courent chercher des clefs pour ouvrir des portes où se situent d'autres clefs, l'une des dernières épreuves consiste à jouer au jeu du bâton contre une personne déguisée en lion.

Dans le jeu des bâtons, il y a 21 bâtons disposés devant deux joueurs. Chaque joueur peut prendre, lorsque c'est son tour, entre 1 et 3 bâtons. Le joueur qui récupère le dernier bâtonnet a perdu. Implémentez ce jeu à deux joueurs : à chaque tour, votre programme demandera à chaque joueur de prendre un bâton, et annoncera le vainqueur lorsque la partie sera terminée.

Question subsidiaire : Il existe une stratégie optimale permettant, si on ne commence pas en premier le jeu, de gagner à tous les coups. La connaissez-vous ?

Probabilités

Dans un jeu télévisé, le présentateur vous propose de choisir une boîte parmi trois. Dans l'une d'entre elles se trouve un grand prix de plusieurs milliers d'euros ; dans les deux autres, trois heures de TP Matlab... Au cas où ceci ne serait pas clair car les TPs Matlab sont vraiment formidables, votre but est de trouver la boîte avec l'argent.

Le présentateur, qui sait où se trouve l'argent, vous propose de choisir une boîte parmi les trois. Une fois votre choix fait, il soulève, parmi les deux boîtes que vous n'avez pas choisies, une boîte avec le TP dedans (il y en a nécessairement une parmi les deux qui contient le TP, et le présentateur, qui sait où se situe l'argent, peut soulever une boîte "perdante"). Il vous propose alors de changer de boîte et prendre la boîte restante que vous n'aviez pas choisie au départ : acceptez-vous ce choix ?

1. Pour répondre à la question, vous devrez modéliser ce problème sous Matlab : trois boîtes dont une contient un prix, choisir au hasard une des boîtes, et éliminer une des boîtes restantes. Vous devrez recommencer cette expérience plusieurs milliers de fois afin de calculer dans combien de cas l'échange proposé par le présentateur était bénéfique. La fonction **rand** vous sera très utile.
2. La réponse obtenue vous semble-t-elle satisfaisante ? Dans votre code, le fait que le présentateur soulève une boîte change-t-il quelque chose ?
3. Que se passe-t-il si on modélise le problème en disant que le présentateur ne connaît pas la boîte contenant le prix, mais qu'il a eu de la chance et a soulevé une boîte vide devant vous ?

Le crible d'Eratosthène

Le crible d'Eratosthène est une méthode permettant de trouver tous les nombres premiers dans un intervalle $[2; n]$. Pour ce faire, la méthode commence par supposer que tous les nombres entre 2 et n sont premiers (ce qui est, quand n est supérieur ou égal à 4, faux). Puis, elle répète les étapes suivantes jusqu'à avoir parcouru testé tous les nombres de l'intervalle :

— Parmi les nombre que l'on n'a pas testé, on choisit le plus petit d'entre eux : on le nomme i.
— Si i est toujours marqué comme étant premier, on parcourt tous les nombres de l'intervalle $[2i; n]$, de i en i : chaque nombre de cet intervalle étant un multiple de i, on le marque comme n'étant pas premier.
— Si i est marqué comme non premier, on ne fait rien.

1. Codez cette méthode en Matlab afin d'afficher tous les nombres entre 1 et n qui sont premiers.

2. Récupérez tous les nombres premiers entre 1 et 100000 dans un vecteur **e**. Grâce à la fonction **hist(e,30)**, affichez l'histogramme de **e**, c'est à dire la quantité de nombres premiers dans des intervalles réguliers entre 1 et 100000. Les nombres premiers ont tendance à devenir plus rare, moins rare, ou rester en même quantité quand les nombres deviennent grands ?
3. Notez le temps de calcul de votre algorithme pour différentes valeurs de n : $25000, 50000, 100000, 150000, 200000$. Tracez ces temps de calcul sur un graphe à l'aide d'une feuille de Excel : le temps de calcul augmente-t-il de façon proportionnelle à n ?

Normaliser une matrice

La normalisation d'un ensemble de valeurs est une opération importante en mathématiques, consistant à transformer un ensemble de valeurs dans un intervalle $[v_{min}; v_{max}]$ en un ensemble de valeurs dans $[0; 1]$. Générez une matrice de valeurs aléatoires de cette façon :

```
a = randi(100,1,1);
A = randi([a, randi([a+10, a+400],1,1)], 6, 8);
```

Ecrivez le code permettant de normaliser la matrice A, c'est à dire rapporter toutes ses valeurs entre 0 et 1. Pour vérifier que votre code fonctionne, vérifiez bien que, dans votre matrice normalisée, vous avez au moins un 0 et un 1.

Calculer et afficher une série

Calculer, dans un vecteur, les n premiers termes de la suite $u_n = 0.5^n$. Ensuite, à l'aide d'un code vectorisé ne contenant aucune boucle explicite, calculez les n premiers termes de la série $v_n = \sum_{i=1}^{n} u_n$.

Quelle est la limite de cette série (géométrique) ? En traçant le résultat à l'aide de la fonction **plot**, confirmez-vous la limite que vous aviez conjecturée ?