

---

## Feuille de TP n°2 – Initiation à Matlab

---

Ce second TP porte sur les entrées et sorties, les fonctions et les outils graphiques dont vous disposez sous Matlab.

### 1 Entrées et sorties.

La commande `input` permet de demander à l'utilisateur Matlab d'entrer les valeurs de variables à utiliser. La commande `pause` permet de stopper l'exécution Matlab. Vous pouvez préciser le nombre de secondes de pose ou revenir à Matlab en appuyant sur n'importe quelle touche. La commande `save` permet de sauvegarder dans un fichier, dont le nom par défaut est `matlab.mat`, le contenu de certaines variables dont vous souhaitez garder une trace. Ce fichier peut être appelé par la commande `load` qui restaure toutes les variables que vous avez sauvegardées.

```
% Affectez une valeur à n
n = input('Entrez la valeur de n: ');
% Affectez une valeur à a
a=input('Precisez la valeur de a: ');
% Création de la matrice de Toeplitz A
v=a.^[0:n]; A = toeplitz(v); d = det(A);
% Sauvegarde de n, a, A, d dans restoep.mat
save restoep n a A d;
% Efface toutes les variables de la session
clear
% Restaure les variables de restoep.mat
load restoep
% Vérification
who
```

### 2 Fonctions

Un ensemble de commandes Matlab peut être considéré comme une fonction. On peut voir une fonction comme un sous-programme Matlab dont les paramètres éventuels sont les arguments de la fonction et dont les résultats sont les images de cette fonction. Beaucoup de fonctions Matlab, comme par exemple `mean`, sont déjà écrites en Matlab et le code Matlab correspondant est stocké dans un fichier dont le nom se termine par `.m`. Pour `mean`, il s'agit de `mean.m`. Ajouter de nouvelles fonctions à Matlab revient donc à écrire de nouveaux fichiers de ce type. Il est d'usage d'appeler une fonction du même nom que le fichier correspondant.

#### Simulation de lois discrètes

Dans votre répertoire personnel, éditer le fichier `probadis.m` suivant dont le code Matlab génère une réalisation aléatoire d'une loi discrète à support fini.

```
function realis=probadis(n,x,p)
% Générateur aléatoire d'une loi discrète à support fini.
% n >= 1 est le cardinal du support de la loi.
% p est un vecteur de n nombres réels positifs tel que sum(p) == 1.
% x est un vecteur de n nombres réels donnant le support de la loi.
% Les réalisations successives sont indépendantes.
```

```

r = rand; a = 0; b = p(1);
for i = 1 : n-1,
    if ((r >= a) & (r < b))
        realis = x(i);
        return;
    end
    a = b; b = b + p(i+1);
end
realis = x(n);
return;

```

La commande Matlab `type` permet de lister le contenu d'un fichier. Ainsi, `type probadis` vous montrera le code source Matlab de la fonction `probadis`. Le commentaire ajouté à partir de la seconde ligne constituera l'aide affichée lorsque l'utilisateur tapera `help probadis`. Finalement, la commande `what` liste les fichiers Matlab du répertoire courant.

**Exercice 2.1.** Créer un code Matlab permettant de générer un vecteur aléatoire  $X$  contenant  $N$  réalisations i.i.d. de loi Binomiale  $\mathcal{B}(n, p)$  où les valeurs  $N, n \geq 1$  et  $0 < p < 1$  sont affectées par l'utilisateur. Pour  $N$  assez grand, vérifier la LGN sur les moyennes empiriques successives de  $X$ .

**Exercice 2.2.** Soit  $(X_n)_n$  une suite de variables aléatoires i.i.d. de loi exponentielle  $\mathcal{E}(\lambda)$  avec  $\lambda > 0$ . Si  $S_n = \sum_{k=1}^n X_k$ ,  $N_0 = 0$  et pour tout  $t > 0$ ,  $N_t = \sum_{n=1}^{\infty} \mathbf{1}_{(S_n \leq t)}$ ,  $(N_t)$  est un processus de Poisson d'intensité  $\lambda$ . Montrer que, pour tout  $t > 0$ ,  $N_t$  suit la loi de Poisson  $\mathcal{P}(\lambda t)$ . En déduire un code Matlab permettant de générer un vecteur aléatoire  $Y$  contenant  $N$  réalisations i.i.d. de loi  $\mathcal{P}(\lambda)$  où les valeurs  $N \geq 1$  et  $\lambda > 0$  sont affectées par l'utilisateur. Pour  $N$  assez grand, vérifier la LGN sur les moyennes empiriques successives de  $Y$ .

**Exercice 2.3.** Pour  $N, N_1$  et  $n \geq 1$  avec  $N_1, n \leq N$ , la loi hypergéométrique  $\mathcal{H}(N, N_1, n)$  est donnée, pour tout  $k \in \mathbb{N}$  avec  $0 \leq k \leq n$ , par  $\mathbb{P}(X = k) = \frac{C_{N_1}^k C_{N-N_1}^{n-k}}{C_N^n}$ . Créer un code Matlab permettant de générer un vecteur aléatoire  $Z$  contenant  $M$  réalisations i.i.d. de loi  $\mathcal{H}(N, N_1, n)$  où les valeurs  $M, N \geq 1$  et  $N_1, n \leq N$  sont affectées par l'utilisateur.

- Si  $N$  tend vers l'infini et le rapport  $N_1/N$  tend vers  $p$  avec  $0 < p < 1$ , montrer que  $X$  converge en loi vers la loi Binomiale  $\mathcal{B}(n, p)$ . Pour  $M, N$  assez grand et  $N_1 = pN$  avec  $0 < p < 1$ , tracer l'histogramme de  $Z$  et comparer le à la loi  $\mathcal{B}(n, p)$ .
- Si  $N, N_1$  et  $n$  tendent vers l'infini et le produit  $nN_1/N$  tend vers  $\lambda > 0$ , montrer que  $X$  converge en loi vers la loi de Poisson  $\mathcal{P}(\lambda)$ . Pour  $M, N$  assez grand,  $N_1 = \lambda\sqrt{N}$  et  $n = \sqrt{N}$ , tracer l'histogramme de  $Z$  et comparer le à la loi  $\mathcal{P}(\lambda)$ .

### 3 Représentations graphiques

```

clear; n = 1000; lambda = 0.5;
X = -log(rand(n,1))/lambda;
% Création d'une nouvelle fenêtre graphique.
figure;
% Trace les moyennes empiriques successives de X
plot(cumsum(X) ./ [1:length(X)], 'b')
% Titre de la figure
title('Loi_des_Grands_Nombres')
% Titre des abscisses
xlabel('Nombre_de_réalisations')
% Titre des ordonnées
ylabel('Moyennes_\empiriques')
% Garde la fenêtre graphique
hold on

```

```
% Trace la limite théorique
plot(1/lambda*ones(n,1), 'r—')
% Légende
legend('Empirique', 'Theorique')
```

**Exercice 3.1.** Ajouter à vos codes Matlab les représentations graphiques rencontrées ci-dessus.