

NOM :
PRENOM :

Date :
Groupe :

Feuille-réponses du TP 1 Simulation de trajectoires du modèle CRR

Prise en main de Scilab

Recherchez le programme Scilab et lancez-le. La fenêtre qui s'ouvre s'appelle la *console*. Elle sert essentiellement à l'affichage des résultats. On peut y saisir une commande, telle la demande d'affichage de la valeur d'une variable (on tape le nom de la variable, puis "return"), mais il convient d'éviter de l'utiliser pour des commandes plus compliquées telles les boucles, etc. car il n'est pas facile de demander la réexécution.

Pour saisir une succession de commandes, il convient d'ouvrir une *fenêtre d'édition*. Pour cela utiliser la commande de menu **Application-Editeur**. Il s'agit d'un assez simple éditeur de texte (il reconnaît quand même le nom des commandes et constantes de Scilab qu'il affiche d'une autre couleur). Comme pour tout éditeur, il convient de commencer par donner un nom au fichier édité. Pour cela, utiliser la commande de menu **Fichier-Sauvegarder-sous** et choisir le nom `CalSto1.sce` et éventuellement le disque et répertoire où vous voulez sauvegarder votre séance (votre clé USB par exemple). Il est essentiel de garder les fichiers de vos séances : cela ne vous sera pas seulement utile pour vos révisions ; dès la semaine suivante vous pourrez repartir des idées de la semaine précédente pour développer de nouvelles questions.

A présent, vous allez taper les instructions successives dans votre fenêtre d'édition, puis les marquer et demander l'exécution de la zone marquée par la commande de menu **Executer-Evaluer-la-selection**. Ainsi vous construisez progressivement une succession d'instructions et en testez l'effet en lisant dans la **console** les résultats produits. Vous pourrez ensuite réexécuter tout ou partie de votre code, dans lequel vous aurez peut-être modifié la valeur de quelques constantes.

Avant d'exécuter tout ou partie de votre code, tapez régulièrement **Ctrl-S** (ou exécutez la commande de menu **Fichier-Sauvegarder**) car il arrive que Scilab plante(!) (ou déjante(!)) c'est-à-dire s'arrête complètement et il ne vous reste plus qu'à repartir, en double-cliquant sur le fichier que vous avez fort heureusement sauvegardé ! Notez que la fenêtre d'édition s'affiche souvent par dessus la fenêtre console, ce qui gêne pour lire les résultats sur cette console. Pour cela, **modifiez la place et la taille des deux fenêtres** : à gauche la console et à droite l'éditeur plus large, les deux s'étirant du haut en bas de votre écran. Évitez tout chevauchement.

Exercice 1. : Calcul de la marche CRR On rappelle que la marche CRR S_t est définie par sa valeur initiale S_0 et la relation $S_{t+\delta t} = S_t U_t$, avec $U_t \in \{\text{up}, \text{down}\}$. On choisit ici $up = e^{+\frac{\sigma}{\sqrt{n}}}$ et $down = e^{-\frac{\sigma}{\sqrt{n}}}$, où σ est une constante (la volatilité de l'actif) et n le nombre d'étapes, $T = n\delta t$. Pour les calculer on définit une fonction $S(i, j) = S_0 up^j down^{i-j}$, où $i = 0, \dots, n$ est l'indice du pas de temps $t = i\delta t$ et $j = 0, \dots, i$ le nombre de *up* intervenus avant t . Calculer au moyen de Scilab les valeurs de S_t dans un modèle à $n = 3$ étapes puis représenter sur le dessin suivant les 10 points de coordonnées $(i, S(i, j))$ pour $i = 0, \dots, 3$ et $j = 0, \dots, i$) en les liant pour former le début de l'arbre CRR de ce modèle.

Exercice 2. : Représentation d'une trajectoire Une trajectoire de la marche CRR est caractérisée par une suite de *up* et de *down* que nous représenterons par une suite de 1 et de 0. On désigne par $J(i)$ la fonction qui donne pour une trajectoire le nombre de *up* entre l'instant $t = 0$ et l'instant $t = i\delta t$. Ainsi pour la trajectoire donnée par la suite 0, 1, 0, 0, 0, 1, 0, $J(2) = J(5) = 1$ et $J(6) = 2$.

1. Examiner la seconde partie du code et tâcher d'en comprendre chaque instruction (avec l'aide en ligne au besoin). Exécutez le, décrire la figure obtenue et expliquer.
2. Faire tracer par **Scilab** plusieurs trajectoires en changeant la suite des 0 et 1. Comment obtenir une trajectoire dont les deux extrémités sont S_0 ? Quelle est la valeur la plus petite possible qu'une trajectoire peut atteindre (lorsque $n = 10$)?

Exercice 3. : Simulation d'une suite aléatoire de 0 et de 1

1. La fonction `rand()` de Scilab (comme la touche `random` d'une calculatrice) renvoie un nombre aléatoire compris entre 0 et 1, distribué selon une loi uniforme sur $[0, 1]$. Si l'on précise `rand(m,n)`, on obtient une matrice de taille $m \times n$ dont toutes les composantes sont des nombres aléatoires distribués selon une loi uniforme sur $[0, 1]$. Faire quelques essais pour diverses (petites) valeurs de m et de n . La fonction `2*rand()-3` renvoie encore un nombre aléatoire, mais il n'est plus distribué selon une loi uniforme sur $[0, 1]$. Faire quelques essais puis déterminer quelle loi est simulée par cette fonction. Expliquer.
2. La suite des instructions suivantes permet de vérifier si la loi simulée est bien une loi uniforme :
`x=rand(1,1000);histplot(10,x)`. Que vaut x ? Que vaut `x(1,1:5)`? Expliquer ce que vous voyez sur la figure produite. Indiquer quelle est la classe de plus grand effectif et celle de plus petit? Voyez-vous comment calculer leurs effectifs?

Exercice 1 : la marche aléatoire CRR

```
clear; // supprime les variables, fonctions, ... déjà introduites
n=3;T=1; deltat=T/n;S0=140; sigma=0.2;
up=exp(sigma*sqrt(deltat));down=1/up;
function s=S(i,j); // on doit avoir 0<=j<=i<=n
    s=S0*up.^j.*down.^(i-j);
endfunction;
```

Exercice 2 : Tracé d'une trajectoire de CRR de longueur 10

```
deltaJ=[1 1 0 0 0 1 0 0 1 1];
J=cumsum(deltaJ);
n=10; deltat=T/n;up=exp(sigma*sqrt(deltat));down=1/up;
tttraj=zeros(1,n+1);
tttraj(1)=S(0,0);
for i=1:n
    tttraj(i+1)=S(i,J(i));
end;
scf(1); //crée et active la fenêtre 1
plot2d(1:10,tttraj);
```

Exercice 3 : Simulation d'une suite de quarante 0 et 1

```
x=rand(1,40);
scf(2);
histplot(10,x)
y=grand(1,40, 'bin",1,0.25);
scf(3);
histplot(2,y)
```

Exercice 4 : Simulations de M trajectoires de CRR de longueur n

```
M=20; n=40;up=exp(sigma*sqrt(deltat));down=1/up; p=1/2;
MdeltaJ=grand(M,n, 'bin",1,p);
MJ=cumsum(MdeltaJ,"c");
Mttraj=zeros(M,n+1);
for m=1:M
    Mttraj(m,1)=S0;
    for i=1:n
        Mttraj(m,i+1)=S(i,MJ(m,i));
    end;
end;
scf(4);
for m=1:M
    plot2d(0:n,Mttraj(m,1:n+1));
end;
```

Exercice 5 : Ensemble de trajectoires coïncidant au debut

```
for m=1:M
    MdeltaJ(m,1:10)=deltaJ;
end;
MJ=cumsum(MdeltaJ,"c");
for m=1:M
    Mttraj(m,1)=S0;
    for i=1:n
        Mttraj(m,i+1)=S(i,MJ(m,i));
    end;
end;
scf(4);
for m=1:M
    plot2d(0:n,Mttraj(m,1:n+1));
end;
scf(5);
for m=1:M
    plot2d(0:n,Mttraj(m,1:n+1));
end;
```