

ECS1.1 - Correction du TP N ° 5

Exercice 1

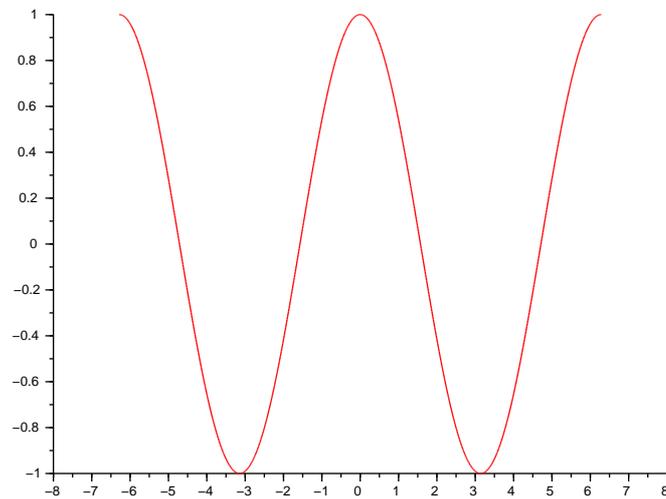
1. Au choix `x=\linspace(-2*pi,2*pi,1000)` ou encore `x=(-2*pi):(4*pi)/999:(2*pi)`.

Le ; sert à ne pas afficher ces 1000 points!

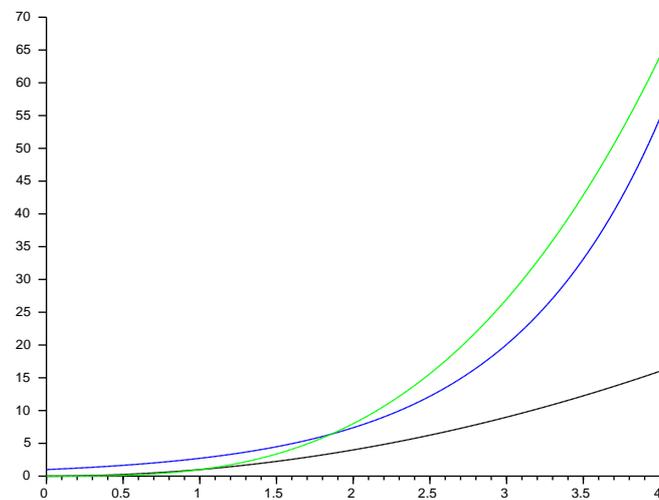
2. Les instructions donnent le même vecteur `y`. La deuxième série d'instruction est préférable en Scilab car elle dispense d'une boucle `for`.
pour créer le vecteur ligne `y` :

```
y=zeros(1,1000); // initialisation facultative  
y=cos(x);
```

3. Pour tracer la courbe en rouge : `plot2d(x,y,5)`



4. On obtient :

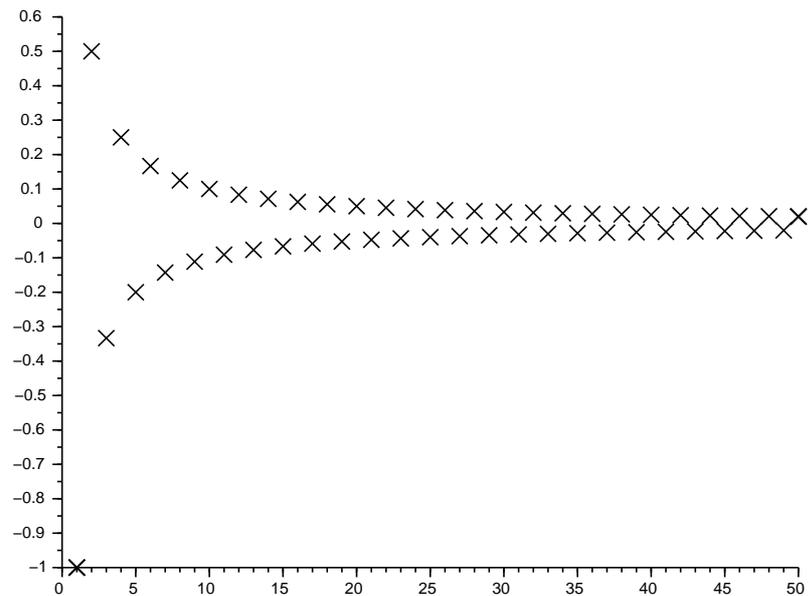


Exercice 2

1. Par exemple :

```
n=50;
x=1:n;
u=(-1)^x./x;
plot2d(x,u,-2) // pour avoir des croix à la place de la ligne brisée
```

On obtient :



2.(a)

```
function[x]=piece(p)
    if rand()<p then x=1;
        else x=0;
    end
```

2.(b)

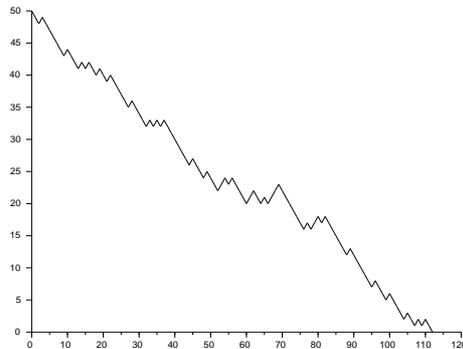
```
function[]=ruine(a,b,p)
    fortune=[a]; // vecteur à une seule composante
    n=1; // temps (décalé de +1)
    while fortune(n)<>0 & fortune(n)<>a+b // tant que personne n'est ruiné
        n=n+1; // temps augmenté de 1
        if rand()<p then fortune(n)=fortune(n-1)+1; // on ajoute une composante au vecteur fortune
            fortune(n)=fortune(n-1)-1; // qui prend la valeur de la nouvelle fortune d
        end
    end
```

À la fin, fortune est un vecteur ligne, tel que fortune(i) donne la fortune du joueur à la fin du (i-1)-ième tour.

Pour afficher un graphique et la durée du jeu :

```
function []=ruine(a,b,p)
    fortune=[a];
    n=1;
    while fortune(n)<>0 & fortune(n)<>a+b
        n=n+1;
        if rand()<p then fortune(n)=fortune(n-1)+1;
            fortune(n)=fortune(n-1)-1;
        end
    end
    x=0:(n-1); // vecteur ligne du temps ou x=linspace(0,n-1,n)
    clf; // efface la figure précédente
    plot2d(x,fortune) // affiche graphiquement le vecteur fortune en fonction du temps
    disp(n-1,'Durée du jeu:')
```

Par exemple avec ruine(50,500,0.3) on obtient une durée de 112 tours et le graphique :



2.(c) Cette fois :

```
function []=ruine(a,b,p,nbrepet)
for i=1:nbrepet
    fortune=[a];
    n=1;
    while fortune(n)<>0 & fortune(n)<>a+b
        n=n+1;
        if rand()<p then fortune(n)=fortune(n-1)+1;
            fortune(n)=fortune(n-1)-1;
        end
    end
    duree(i)=n-1;
end
S=0;
for i=1:nbrepet
    S=S+duree(i); // S sera la somme des composantes de duree
end
disp(S/nbrepet,'Durée moyenne estimée')
if p=0.5 then disp(a*b,'Durée moyenne théorique')
    else q=1-p;
        x=q/p;
        disp(1/(p-q)*((a+b)*(x^a-1)/(x^(a+b)-1)-a),'Durée moyenne théorique')
    end
end
```

Ou sans boucle for :

```
function []=ruine(a,b,p,nbrepet)
for i=1:nbrepet
    fortune=[a];
    n=1;
    while fortune(n)<>0 & fortune(n)<>a+b
        n=n+1;
        if rand()<p then fortune(n)=fortune(n-1)+1;
            fortune(n)=fortune(n-1)-1;
        end
    end
    duree(i)=n-1;
end
S=duree'*ones(nbrepet,1); // attention duree est un vecteur colonne
disp(S/nbrepet,'Durée moyenne estimée')
if p=0.5 then disp(a*b,'Durée moyenne théorique')
    else q=1-p;
        x=q/p;
        disp(1/(p-q)*((a+b)*(x^a-1)/(x^(a+b)-1)-a),'Durée moyenne théorique')
    end
end
```

Par exemple avec `ruine(50,500,0.3,1000)` on obtient une durée moyenne estimée de 125.076 et une durée moyenne théorique de 125.