
TP_N°3_BRALET_DURET_EXERCICE_N°2 :

```
% Lancer des impacts

%Tracé du grillage de la zone à considérer pour les tirs hasardeux
figure(1)
%Nombre de cases qui divisent chaque ligne et colonne
n_cases=24;
%Tracé du grillage
cx=[0:n_cases];
cy=[0:n_cases];
[ CX,CY]=meshgrid(cx,cy);
mesh(CX,CY,zeros(n_cases+1,n_cases+1))
%adaptation en 2D
axis([0 n_cases 0 n_cases])

%Tracée du grillage de la zone à considérer pour les tirs ciblés
figure(3)
%Nombre de cases qui divisent chaque ligne et colonne
n_cases=24;
%Tracé du grillage
cx=[0:n_cases];
cy=[0:n_cases];
[ CX,CY]=meshgrid(cx,cy);
mesh(CX,CY,zeros(n_cases+1,n_cases+1))
%adaptation en 2D
axis([0 n_cases 0 n_cases])

%Matrices comptabilisant le nombre d'impacts par case de la grille
M = zeros(n_cases,n_cases);
M_cible = zeros(n_cases,n_cases);

%Nombre d'impacts à considérer
n_impacts = 537;

%Probabilité qu'un impact touche la zone ciblée
p = 0.1;

% Lancer des impacts

for impact = 1:n_impacts
    %Coordonnée d'un impact non ciblé
    x = n_cases*rand();
    y = n_cases*rand();

    %Coordonnées d'un impact ciblé
    %Probabilité que l'impact tombe sur la cible
    a = rand();

    %Coordonnées modifiées si la cible est touchée
    %La mutiplication permet de réduire la zone à cibler
    %L'addition permet de déplacer cette zone
```

```

if a < p
    x_cible = 0.5*x+10;
    y_cible = 0.5*y+3;
    %Coordonnées inchangées dans le cas contraire
else
    x_cible = x;
    y_cible = y;
end

%Tracer des impacts hasardeux
figure(1)
hold on
plot(x,y,'*r')

%Tracer des impacts ciblés
figure(3)
hold on
plot(x_cible,y_cible,'*r')

%Comptabilisation des impacts
for ligne = 1:n_cases      %Repertoriage par ligne
    %Hasardeux
    if ceil(y) == ligne    %Teste si l'impact est dans la ligne
        for colonne = 1:n_cases      %Repertoriage par colonne
            if ceil(x) == colonne    %Teste si l'impact est
dans la colonne
                %Actualisation de la matrice de comptage
                M(colonne,ligne) = M(colonne,ligne) + 1;
            end
        end
    end
end

%Ciblés
if ceil(y_cible) == ligne    %Teste si l'impact est dans la
ligne
    for colonne_cible = 1:n_cases      %Repertoriage par
colonne
        if ceil(x_cible) == colonne    %Teste si l'impact est
dans la colonne
            M_cible(colonne_cible,ligne) =
M(colonne_cible,ligne) + 1;
        end
    end
end
end

end

%Vectorialisation de la matrice M
M = M(:);
%Nombre d'impact maximal sur une seule case
maxi=max(M);
%Repertoriage du nombre de cases touchées par n impacts
hcases_impact = hist(M',0:maxi);

```

```

%Tracé du nombre de cases touchées par n impacts
figure(2)
bar(0:maxi,hcases_impact/n_cases^2)

%Vectorialisation de la matrice M_cible
M_cible;
M_cible = M_cible(:);
%Nombre d'impact maximal sur une seule case
maxi_cible=max(M_cible);
%Repertoriage du nombre de cases touchées par n impacts
hcases_impact_cible = hist(M_cible',0:maxi_cible);

%Tracé du nombre de cases touchées par n impacts
figure(4)
bar(0:maxi_cible,hcases_impact_cible/n_cases^2)

% Calcul des lois de poisson théoriques

%Nombre d'impact moyen selon la simulation due au hasard
nb_imp_moy = mean(M);
%Résultats de la loi de Poisson de paramètre le nombre précédent
Poi = [];

%Calcul de la loi de Poisson théorique
for k = 0:maxi_cible
    Poi = [Poi, exp(-nb_imp_moy)*nb_imp_moy^k*1/factorial(k)];
end

%Superposition de la loi théorique avec l'étude empirique
figure(2)
hold on
bar(0:maxi_cible,Poi, 0.3,'r')

%Calcul de l'écart type théorique et empirique
ecarttheo = sqrt(nb_imp_moy)
ecartempi = std (M)

%Calcul de la théorie pour les tirs ciblés
%Nombre d'impact moyen selon la simulation due aux tirs ciblés
nb_imp_moy_cible = mean(M_cible);
%Résultats de la loi de Poisson de paramètre le nombre précédent
Poi_cible = [];

%Calcul de la loi de Poisson théorique
for k = 0:maxi_cible
    Poi_cible = [Poi_cible, exp(-
nb_imp_moy_cible)*nb_imp_moy_cible^k*1/factorial(k)];
end

ecarttheo_cible = sqrt(nb_imp_moy_cible)
ecartempi_cible = std (M_cible)

```

```

%Superposition de la loi théorique avec l'étude empirique de la loi de
Poisson
figure(4)
hold on
bar(0:maxi_cible,Poi_cible, 0.3,'r')

%Paramètre des tracés
figure(1)
xlabel('abscisse des impacts ennemis')
ylabel('ordonnee des impacts ennemis')
title('Resultat des tirs hasardeux de l''ennemei')

figure(2)
xlabel('Nombre d''impact')
ylabel('Probabilite associee')
title('Comparaison de la loi theorique et empirique hasardeuse')
legend('Probabilite empirique hasardeuse','Probabilite theorique')

figure(3)
xlabel('abscisse des impacts ennemis')
ylabel('ordonnee des impacts ennemis')
title('Resultat des tirs cibles de l''ennemei')

figure(4)
xlabel('Nombre d''impact')
ylabel('Probabilite associee')
title('Comparaison de la loi theorique et empirique cible')
legend('Probabilite empirique ciblee','Probabilite theorique')

ecarttheo =

    0.9656

ecartempi =

    0.9676

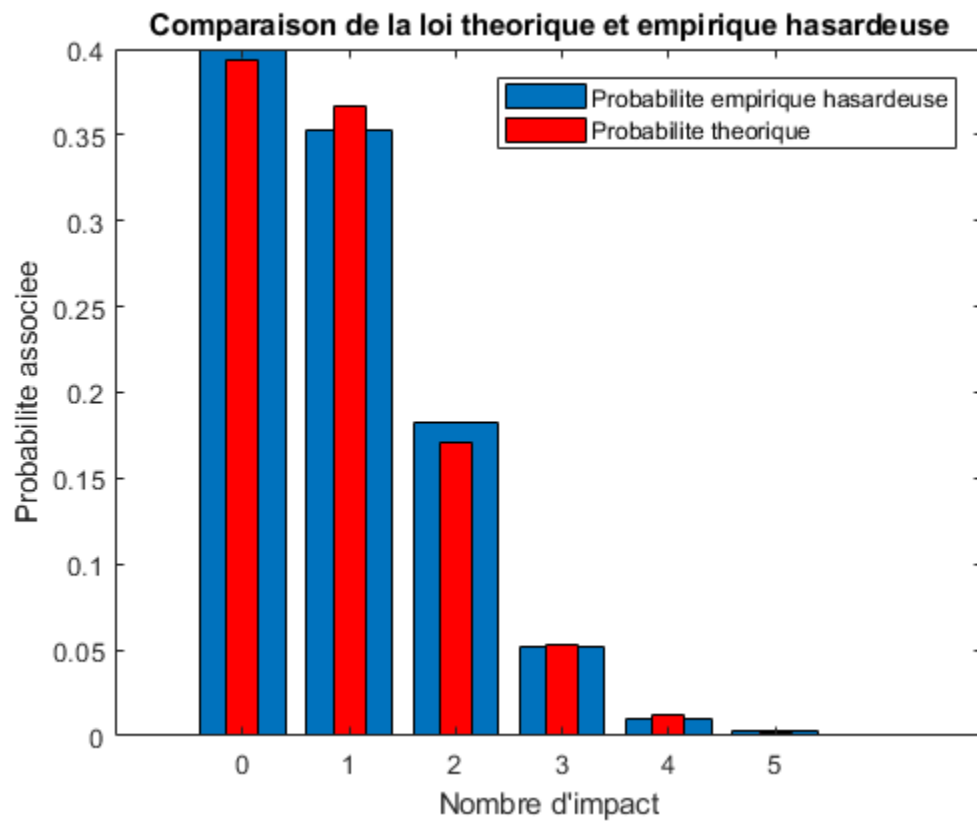
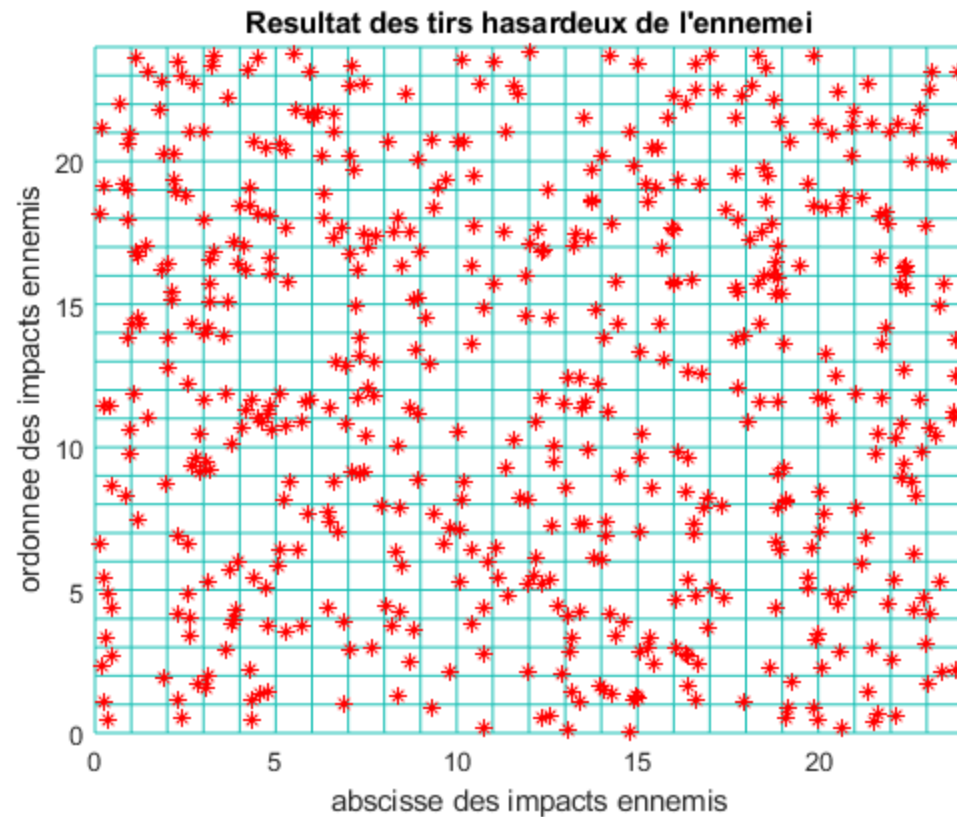
ecarttheo_cible =

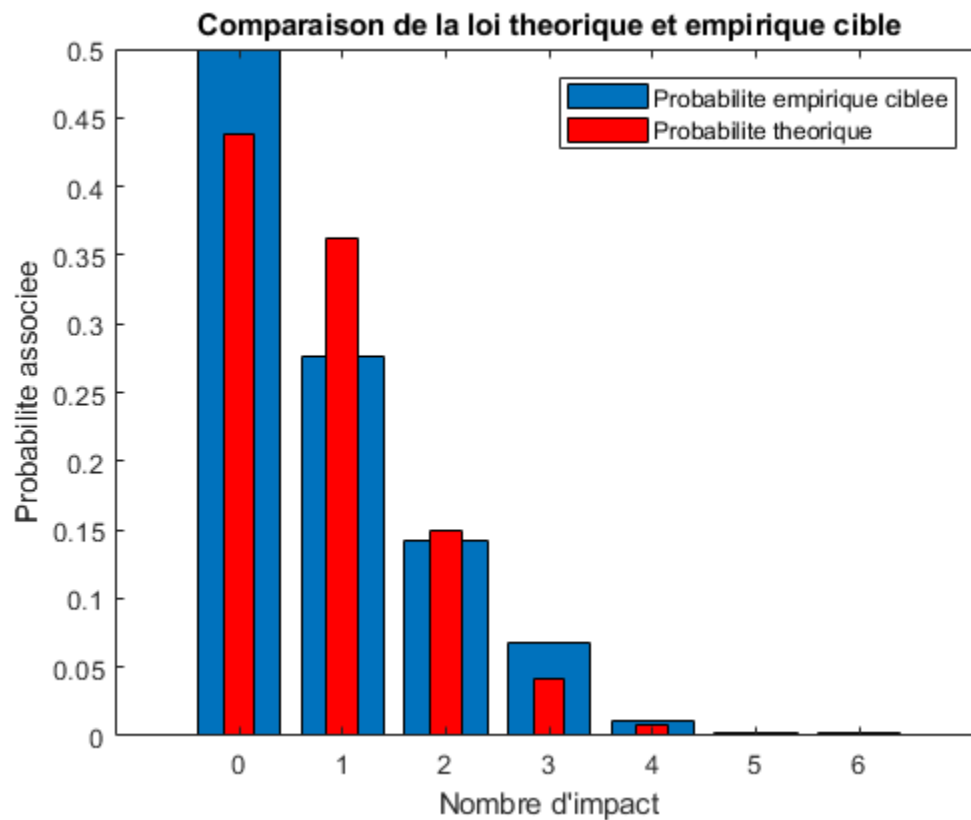
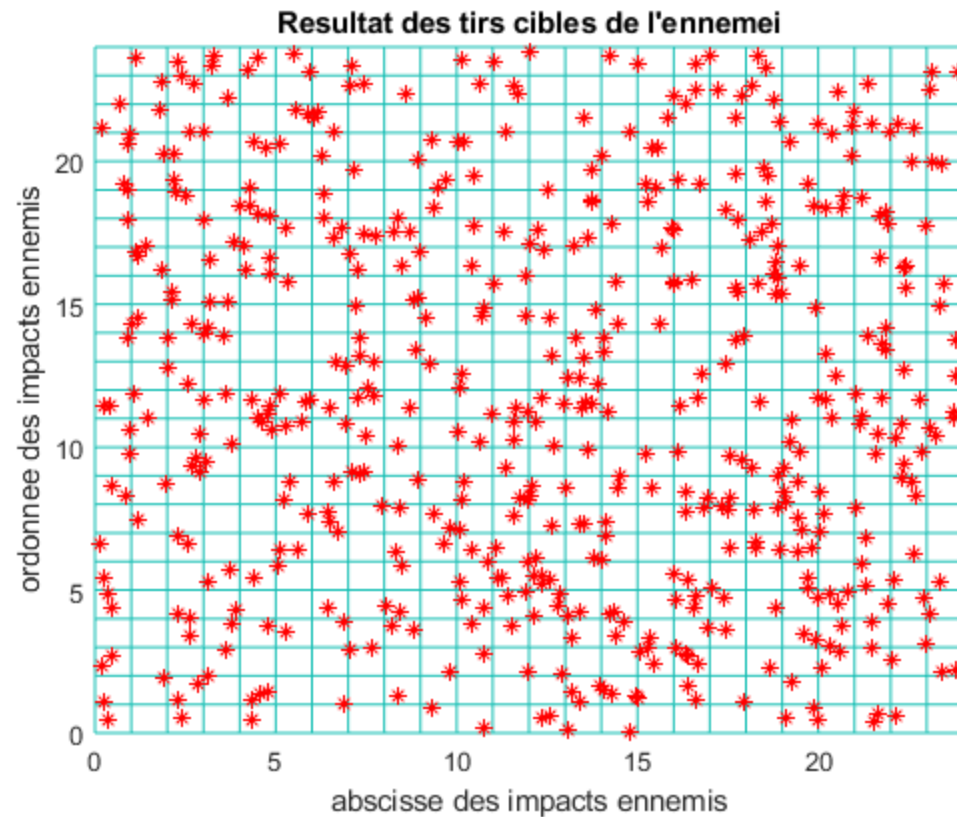
    0.9081

ecartempi_cible =

    1.0243

```





Commentaires

On peut remarquer au fil de cette étude que la simulation empirique d'un lancer de tirs au hasard relève bien de la loi de Poisson comme cela a pu être constaté dans le premier histogramme tandis que lorsque le lancer est ciblé, la loi semble s'écarter significativement de la loi de Poisson qui devrait théoriquement lui correspondre. Ceci permet donc bien de confirmer les dires de l'énoncé, c'est à dire que si la variable aléatoire qui compte le nombre d'impacts par case suit une loi de Poisson, dans ce cas le tir est bel et bien hasardeux, dans le cas contraire il faut alors se pencher plus en avant sur les cases qui ont été touchées car le tir n'a pas été fait au hasard et il y avait bien une cible que l'ennemi s'efforçait de toucher.

Published with MATLAB® R2017b