MODÉLISATION ET CONTRÔLE OPTIMAL D'UNE ÉPIDÉMIE

Présenté par : DEHMANI Mohammed
Thème: Santé et prévention

Plan:

1. Elaboration du modèle SIR
2. Amélioration du modèle SIR
3. Problème de contrôle optimal
4. Résolution théorique
5. Résolution numérique

MODÈLE SIR

r: taux d'infection
γ: taux de guérison
N: population totale

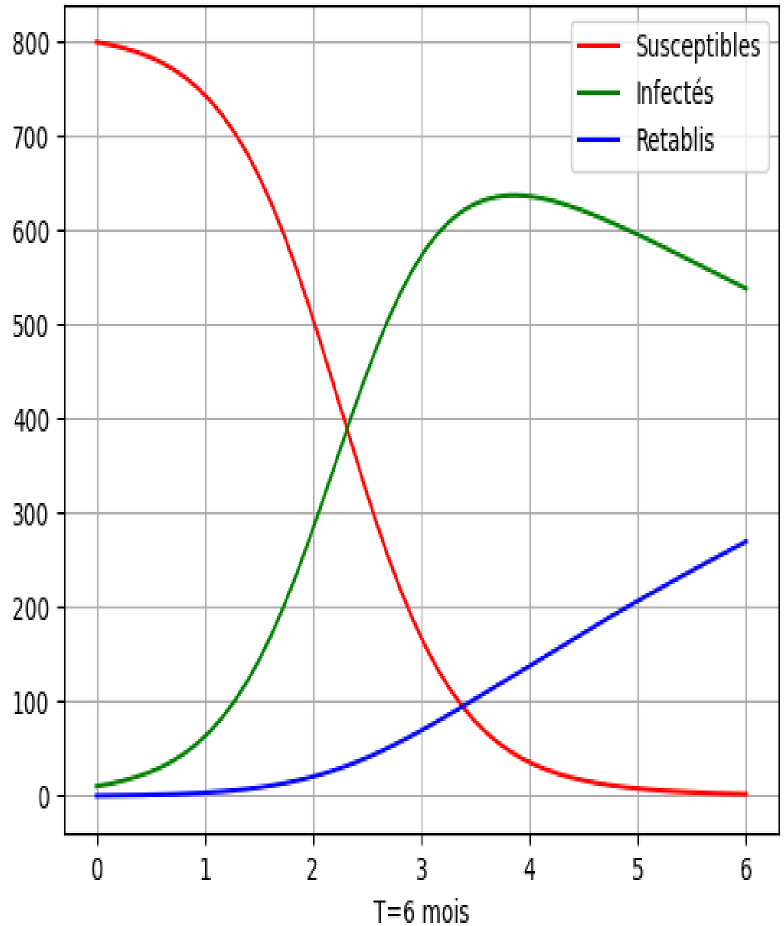
Formulation du système

$$\begin{cases} \dfrac{dS}{dt} = -rI(t)S(t) \\ \dfrac{dI}{dt} = rI(t)S(t) - \gamma I(t) \\ \dfrac{dR}{dt} = \gamma I(t) \\ N = S(t) + I(t) + R(t) = cste \end{cases}$$

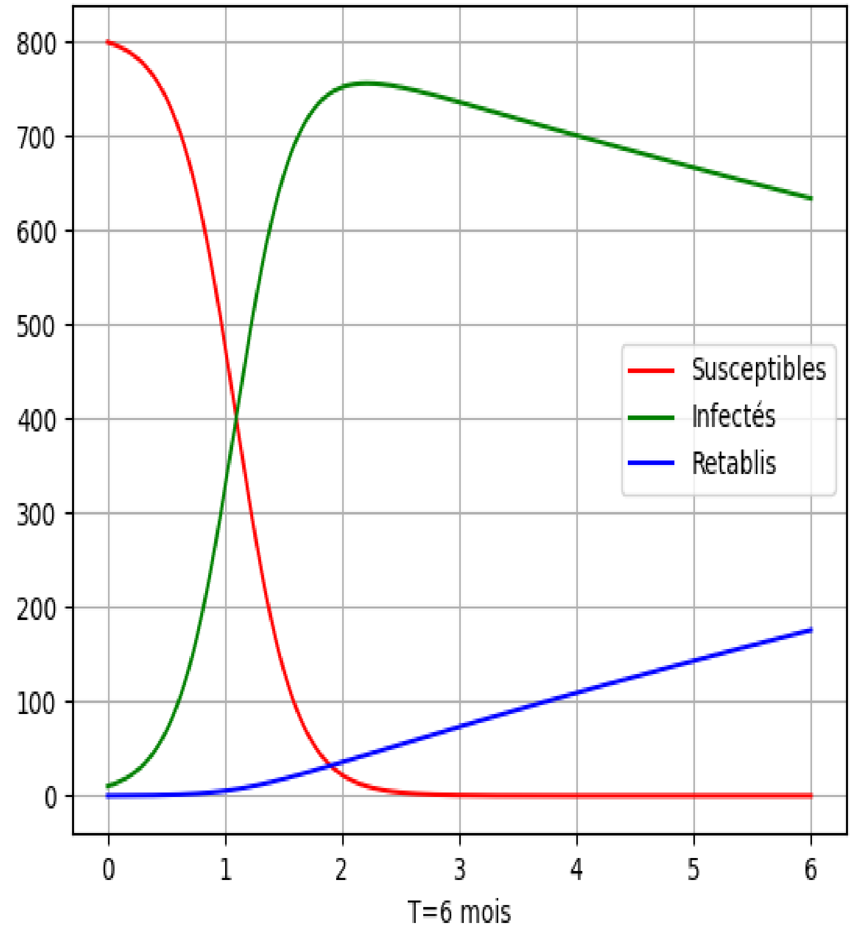
Représentation graphique:

On pose à t=0:
S0=800
I0=10
R0=0
On varie le taux de guérison γ et le taux d'infection r:

Modèle SIR



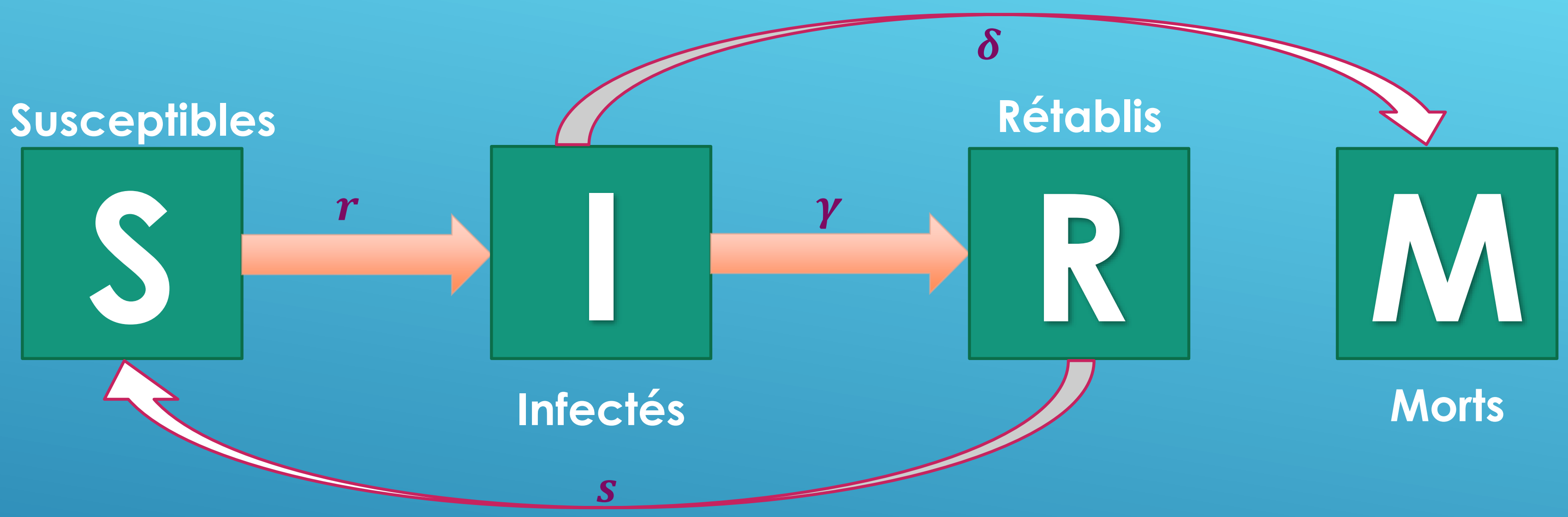
Modèle SIR



r=0,0025
γ=0,111

r=0,005
γ=0,05

Amélioration du modèle SIR



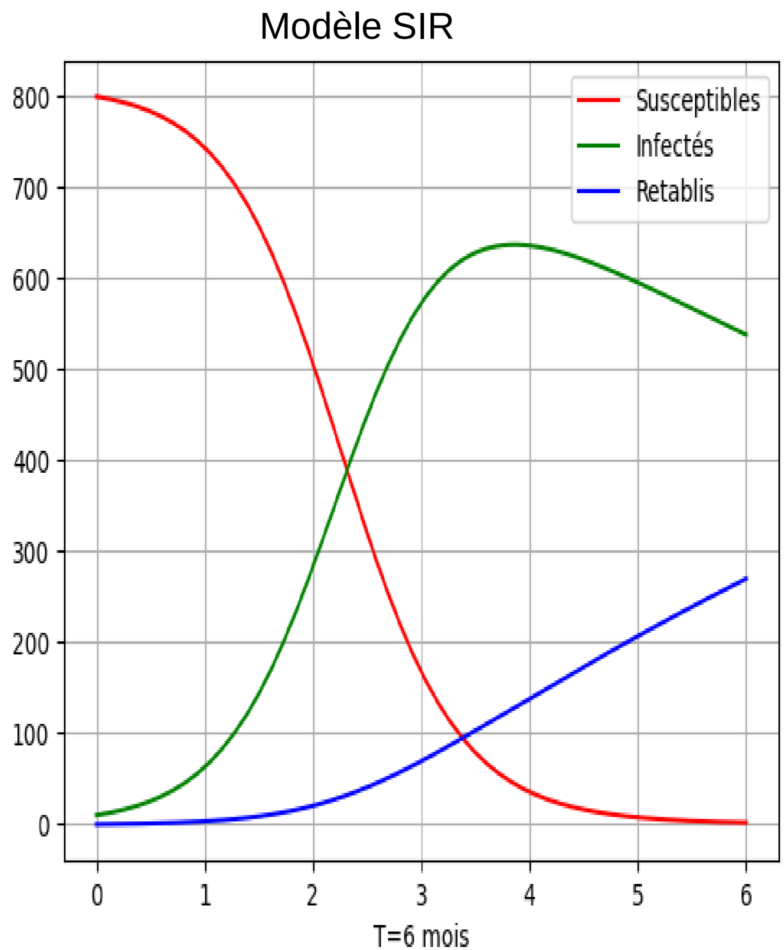
r: taux d'infection
γ: taux de guérison
N: population totale
s: taux d'immunité
δ:taux de mortalité

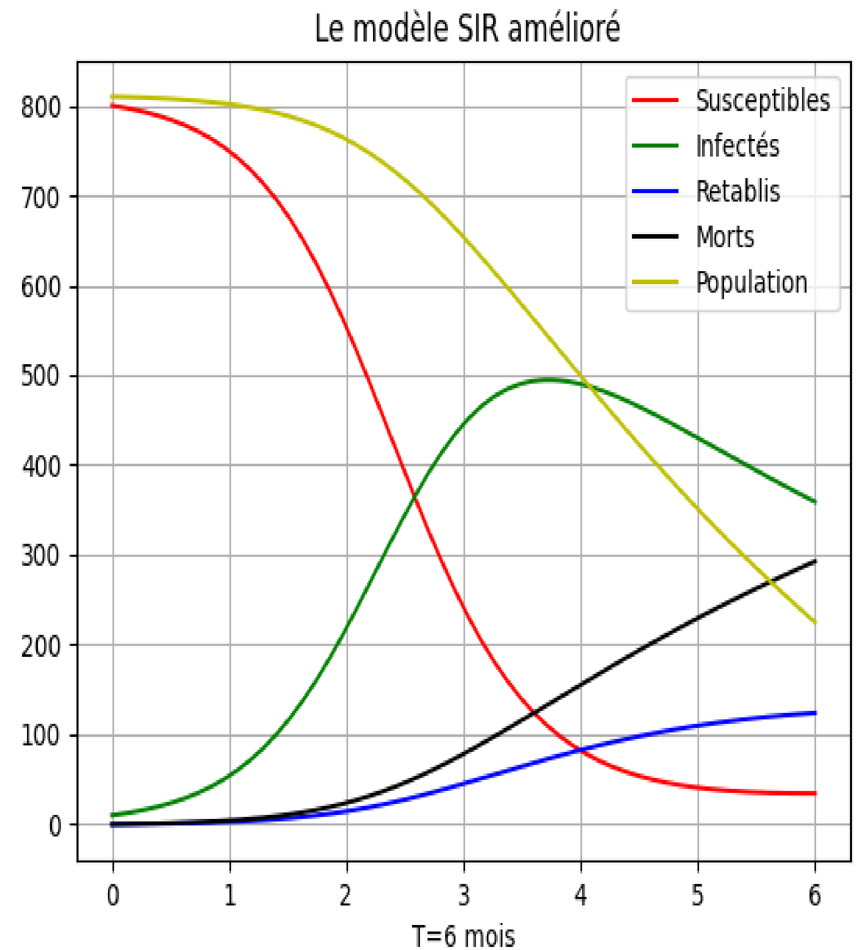
Formulation du système

$$\begin{cases} \dfrac{dS}{dt} = -rI(t)S(t) + sR(t) \\ \dfrac{dI}{dt} = rI(t)S(t) - \gamma I(t) - \delta I(t) \\ \dfrac{dR}{dt} = \gamma I(t) - sR(t) \\ \dfrac{dM}{dt} = \delta I(t) \\ N(t) = S(t) + I(t) + R(t) - M(t) \end{cases}$$

Représentation graphique:

On garde les mêmes conditions initiales.
On pose:
r=0,0025
γ=0,111
s=0,25
δ=0,16





On rappelle que c'est la figure obtenue sans amélioration.

Voici le modèle amélioré:

PROBLÈME DU CONTRÔLE OPTIMAL

Formulation du problème:

$$\begin{cases} \dfrac{dS}{dt} = -rI(t)S(t) + sR(t) - u(t)S(t) \\ \dfrac{dI}{dt} = rI(t)S(t) - \gamma I(t) - \delta I(t) \\ \dfrac{dR}{dt} = \gamma I(t) - sR(t) + u(t)S(t) \\ \dfrac{dM}{dt} = \delta I(t) \\ N(t) = S(t) + I(t) + R(t) - M(t) \\ 0 \leq t \leq T \end{cases}$$

u(t): taux de vaccination

Le problème de contrôle optimal est de minimiser, en un temps T fixé, le coût :

$$J(u) = \alpha I(T) + \int_0^T u^2(t)dt \to min_u$$

$$\begin{cases} J(u) = \alpha I(T) + \int_0^T u^2(t)dt \to min_u \\ \dfrac{dS}{dt} = -rI(t)S(t) + sR(t) - u(t)S(t) \\ \dfrac{dI}{dt} = rI(t)S(t) - \gamma I(t) - \delta I(t) \\ \dfrac{dR}{dt} = \gamma I(t) - sR(t) + u(t)S(t) \\ 0 \le t \le T \qquad 0 \le u(t) \le a \end{cases}$$

α: paramètre de poids

a: le taux maximal de la vaccination

Résolution théorique:

On définit le Hamiltonien pour le problème de contrôle optimal par :

$$\boldsymbol{H(S,I,R,P_s,P_I,P_R,u,t)} \\ = \boldsymbol{p^0u^2 + P_s(-\mathrm{rSI} + \mathrm{sR} - \mathrm{uS}) + P_I(rSI - (\gamma+\delta)I)} \\ \boldsymbol{+ P_R(\gamma I - sR + uS)}$$

Avec: $\boldsymbol{P_s, P_I, P_R}$, sont les vecteurs adjoints

Application du principe du maximum de Pontryagin au problème de l'épidémie

On obtient Les équations adjointes :

$$\begin{cases} \dot{P}_S \stackrel{\text{def}}{=} -\dfrac{\partial H}{\partial S} = rIP_s - rIP_I + uP_s - uP_R \\ \dot{P}_I \stackrel{\text{def}}{=} -\dfrac{\partial H}{\partial I} = rSP_s - rSP_I + (\gamma + \delta)P_I - \gamma P_R \\ \dot{P}_R \stackrel{\text{def}}{=} -\dfrac{\partial H}{\partial R} = -sP_s - sP_R \end{cases}$$

Conditions de transversalité:

$$\begin{cases} P_S(T) = p^0 \dfrac{\partial \alpha I(T)}{\partial S} = 0 \\ P_I(T) = p^0 \dfrac{\partial \alpha I(T)}{\partial I} = p^0 \alpha \\ P_R(T) = p^0 \dfrac{\partial \alpha I(T)}{\partial R} = 0 \end{cases}$$

On maximise le Hamiltonien:

$$\max_{0\le u\le a} H = \max_{0\le u\le a}[p^0u^2 + P_s(-\mathrm{rSI} + \mathrm{sR} - \mathrm{uS}) + P_I(rSI - (\gamma + \delta)I) + P_R\,(\gamma I - sR + uS)]$$

C’est équivalent à maximiser:

$$f(u) = p^0u^2 + u(P_R - P_S)S$$

On pose : $\boldsymbol{p^0 = -\frac{1}{2}}$

Et, en prenant en compte les limites sur u, la caractérisation du contrôle optimal est :

$$u^*(t) = \begin{cases} 0 & \big(P_R(t) - P_S(t)\big)S(t) < 0 \\ \big(P_R(t) - P_S(t)\big)S(t) & 0 \le \big(P_R(t) - P_S(t)\big)S(t) \le a \\ a & \big(P_R(t) - P_S(t)\big)S(t) > a \end{cases}$$

$$\begin{cases} \dot{S}(t) = -rI(t)S(t) + sR(t) - u(t)S(t) & S(0) = S_0 \\ \dot{I}(t) = rI(t)S(t) - \gamma I(t) - \delta I(t) & I(0) = I_0 \\ \dot{R}(t) = \gamma I(t) - sR(t) + u(t)S(t) & R(0) = R_0 \\ \dot{P}_S = rIP_s - rIP_I + uP_s - uP_R & P_S(T) = 0 \\ \dot{P}_I = rSP_s - rSP_I + (\gamma + \delta)P_I - \gamma P_R & P_I(T) = -\frac{1}{2}\alpha \\ \dot{P}_R = -sP_s - sP_R & P_R(T) = 0 \end{cases}$$

Avec :

$$u^*(t) = \begin{cases} 0 & \big(P_R(t) - P_S(t)\big)S(t) < 0 \\ \big(P_R(t) - P_S(t)\big)S(t) & 0 \leq \big(P_R(t) - P_S(t)\big)S(t) \leq a \\ a & \big(P_R(t) - P_S(t)\big)S(t) > a \end{cases}$$

Résolution numérique:

Les résultats d'exécutions sous MATLAB (pour différentes valeurs de α) sont représentés par les figures suivantes.

On considére les mêmes conditions initiales avec :

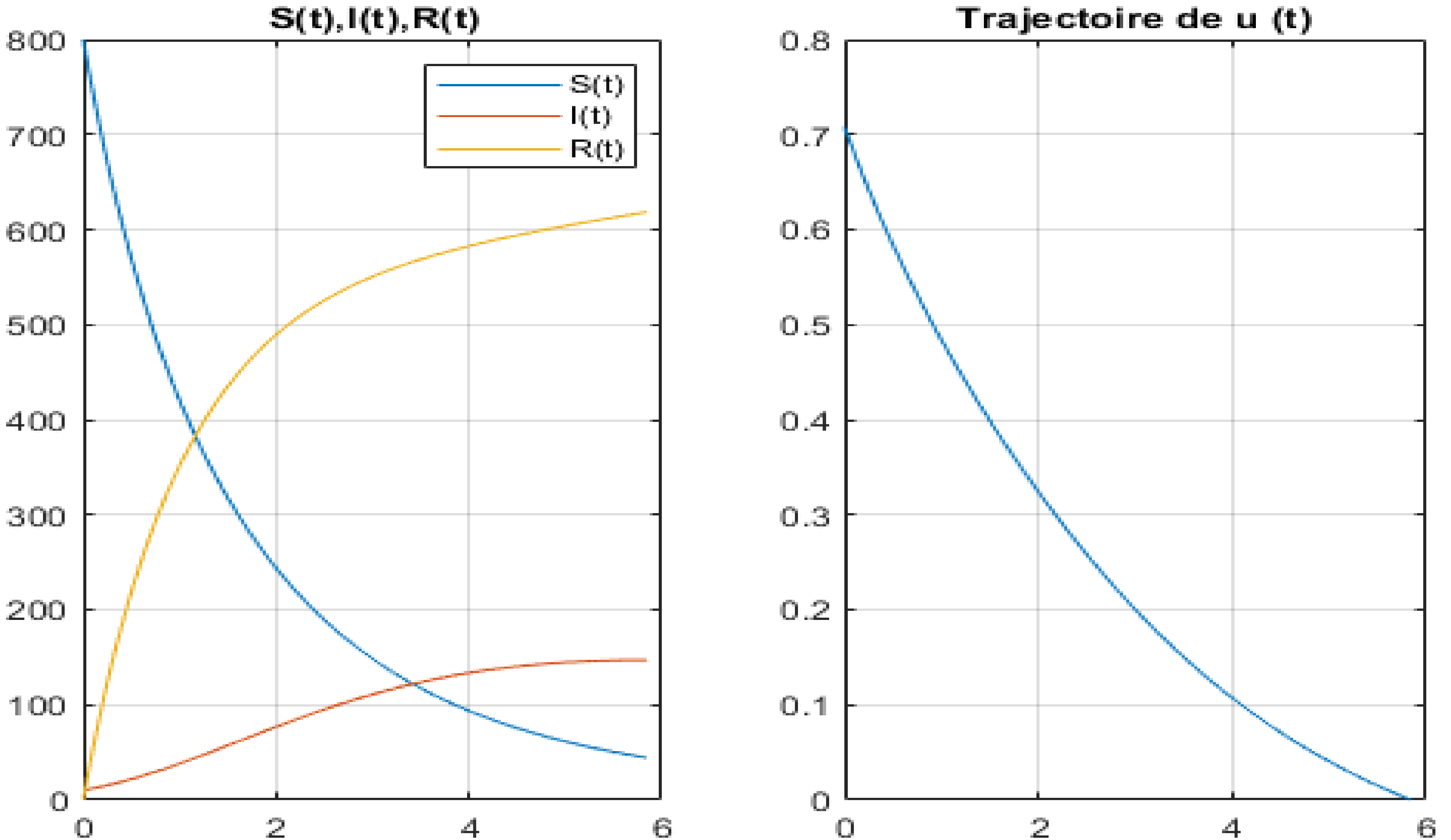
$r = 0.0025$

$gamma = 0.111$

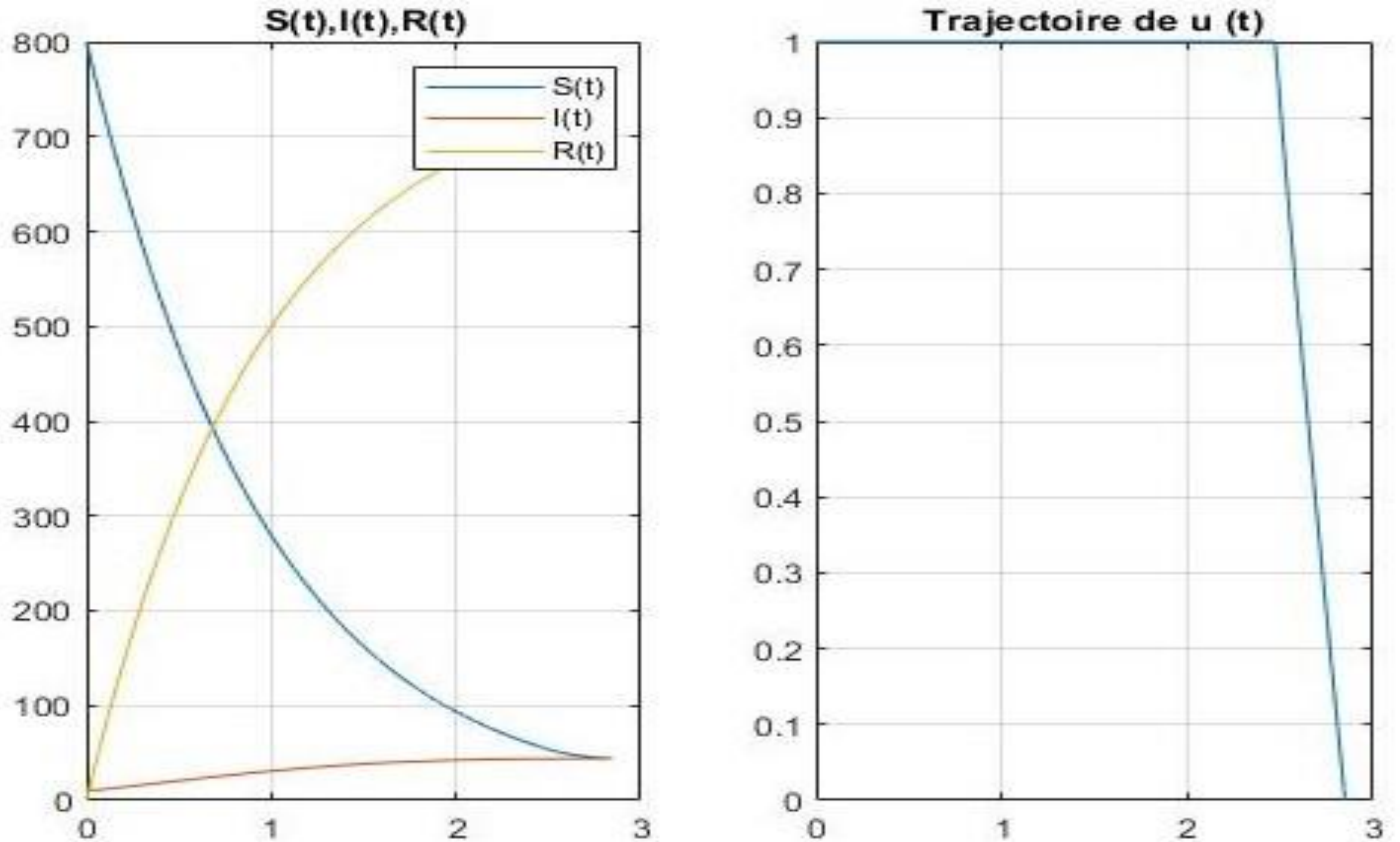
$s = 0.5$

$delta = 0.025$

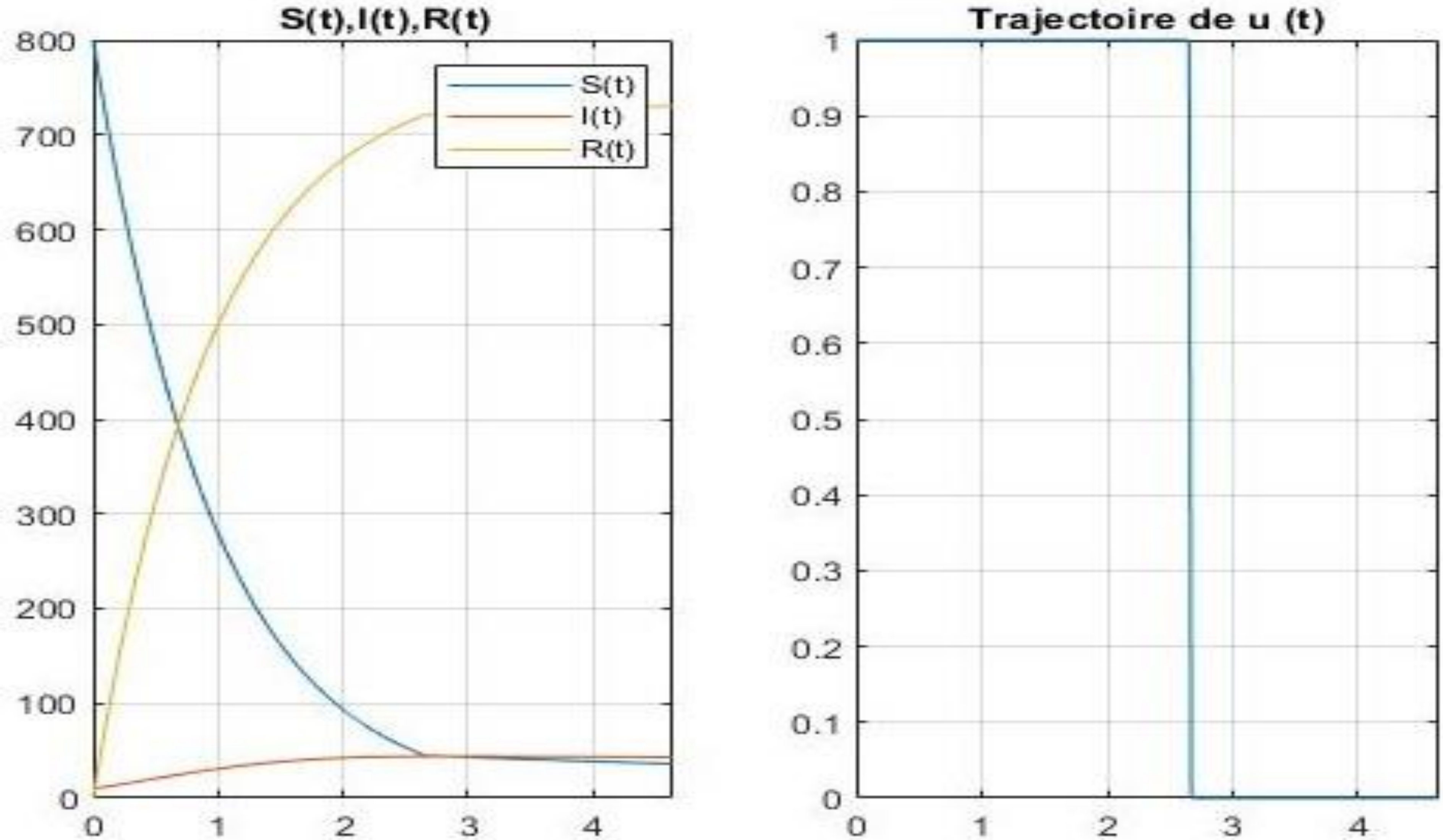
$a = 1$



Solution optimale obtenue avec α=0,005



Solution optimale obtenue avec **α =1**



Solution optimale obtenue avec α =200

ANNEXE:

Code python modèle SIR:

```python
import numpy as np
import scipy.integrate as spi
import matplotlib.pyplot as pl
# Paramètres du modèle
r = float(input("Le taux d'infection est:" ))
gamma = float(input("Le taux de guérison est:"))

def deriv(SIR, t):
    """
    SIR : liste contenant les 3 fonctions inconnus
    t : le temps
    r, gamma : les deux facteurs du modèle
    """
    Y=np.zeros((3))
    V = SIR
    Y[0] = - r * V[0] * V[1]
    Y[1] = r * V[0] * V[1] - gamma * V[1]
    Y[2] = gamma * V[1]
    return Y   # For odeint

# Au temps t0,  800 sains, 10 infécté, 0 guéri
V0 = 800, 10, 0

# Evolution sur 30 jours
t = np.linspace(0, 6,30*3600)

# Resolution des équations differentielles
solu = spi.odeint(deriv, V0, t)

#Ploting
pl.grid(True)
pl.plot(t,solu[:,0], '-r', label='Susceptibles')
pl.plot(t,solu[:,1], '-g', label='Infectés')
pl.plot(t,solu[:,2], '-b', label='Retablis')
pl.legend(loc=0)
pl.xlabel('T=6 mois')
pl.show()
```

Code python modèle SIR amélioré:

```
import numpy as np
import scipy.integrate as spi
import matplotlib.pyplot as pl
# Paramètres du modèle
r = 0.0025
gamma =0.111
s=float(input("Le taux d'immunité' est:"))
delta=float(input("Le taux de mortalité est:"))

def deriv(SIRM, t):
    """
    SIRM : liste contenant les 4 fonctions inconnus
    t : le temps
    r, gamma,s,delta : les quatres facteurs du modèle
    """
    Y=np.zeros((4))
    V = SIRM
    Y[0] = - r * V[0] * V[1] + s * V[2]
    Y[1] = r * V[0] * V[1] - gamma * V[1] - delta * V[1]
    Y[2] = gamma * V[1] -  s * V[2]
    Y[3] = delta * V[1]
    return Y   # For odeint

# Au temps t0,  800 sains, 10 infécté, 0 guéri , 0 morts
V0 = 800, 10,0,0

# Evolution sur 30 jours
t = np.linspace(0, 6, 3600*24)


# Resolution des équations differentielles
sol = spi.odeint(deriv, V0, t)

#Ploting
pl.grid(True)
pl.plot(t,sol[:,0], '-r', label='Susceptibles')
pl.plot(t,sol[:,1], '-g', label='Infectés')
pl.plot(t,sol[:,2], '-b', label='Retablis')
pl.plot(t,sol[:,3], '-k', label='Morts')
pl.plot(t,sol[:,0]+sol[:,1]+sol[:,2]-sol[:,3], '-y', label='Population')
pl.legend(loc=0)
pl.title(' Le modèle SIR amélioré')
pl.xlabel('T=6 mois')
pl.show()
```

Démonstration des passages de la page 21 à 22

$$max_{0\le u\le a}H = max_{0\le u\le a}[p^0u^2 + P_s(-\mathrm{rSI} + \mathrm{sR} - \mathrm{uS}) + P_I(rSI - (\gamma+\delta)I) + P_R(\gamma I - sR + uS)]$$

On maximise par rapport à u(t)

Ainsi il suffit de maximiser la fonction :

$$f(u) = p^0u^2 + u(P_R - P_S)S$$

Conditions nécessaire :

$$f'(u) = 0 \Leftrightarrow 2\,p^0u + (P_R - P_S)S = 0$$

$$\Leftrightarrow u^* = \frac{(P_R - P_S)S}{2\,p^0}$$

On pose: $p^0 = -\frac{1}{2}$, on aura : $u^* = (P_R - P_S)S$

Conditions suffisante :

$f''(u) = 2p^0 = -1 < 0 \quad u^* = (P_R - P_S)S$ est un maximum

Code MATLAB résolution du problème d'optimisation:

```
clear all ; clf ; clc ;
global x0 T r gamma alpha;
x0=[800,10,0] ; T=6 ; r=0.0025 ; gamma=0.111 ; s=0.25 ; delta=0.16 ;
% p0=[-1/2,-1/2,0] for alpha = 0.005  ;
p0=[-1/2,0,1/2];
alpha= 0.005 ;
%===============Calcul du zéro de la fonction de tir===============
options=optimset('Display','iter','LargeScale','on') ;
[p0tf,FVAL,EXITFLAG] = fsolve(@G,[p0,T],options);
EXITFLAG
%===============Tracer les trajectoires optimales=================
options = odeset('AbsTol',1e-9,'RelTol',1e-9) ;
[t,y] = ode45(@sys,[0,p0tf(4)],[x0,p0tf(1),p0tf(2),p0tf(3)],options);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
subplot(1,2,1) ;
plot(t,y( :,1)) ;  hold on
plot(t,y( :,2)) ; hold  on
plot(t,y( :,3)) ; hold on;
grid on
legend("S(t)","I(t)","R(t)");
subplot(1,2,2)
%%================ Trajectoire du contrôle optimal=====================
for i=1:length(y(:,1))
    if (y(i,6) - y(i,4))*y(i,1) < 0
        y(i,7) = 0 ;
    elseif (y(i,6) - y(i,4))* y(i,1) > 1
        y(i,7) = 1 ;
    else
        y(i,7) = (y(i,6) - y(i,4)) * y(i,1);
    end
end
plot(t,y(:,7)) ; title('Trajectoire de u (t)') ;grid on
```

```
%=============== Définition de la fonction de tir ======================
function Yzero=G(y)
global x0 r gamma alpha s delta;
p_puissance_0 = -1/2;
options = odeset('AbsTol',1e-9,'RelTol',1e-9);
[t,y] = ode45(@sys,[0,y(4)],[x0,y(1),y(2),y(3)],options);
if (y(end,6) - y(end,4)) * y(end,1) < 0
    u = 0;
elseif (y(end,6) - y(end,4)) * y(end,1) > 1
    u = 1;
else
    u = (y(end,6) - y(end,4)) * y(end,1);
end
hamend = p_puissance_0*u^2*0 ...
        -r*y(end,1)*y(end,2)*y(end,4)...
        +s*y(end,3)*y(end,4)...
        - u*y(end,1)*y(end,4)...
        +r*y(end,1)*y(end,2)*y(end,5)...
        -(gamma+delta)*y(end,2) * y(end,5) ...
        + gamma * y(end,2) * y(end,6)...
        -s*y(end,3)*y(end,6)...
        + u*y(end,1)*y(end,6);
Yzero=[y(end,4)  y(end,5)+(alpha/(2)) y(end,6) hamend] ;
```

```
%    [ Ps                 PI                 PR     H_max
end
%================ Système extrémal ======================
function ydot=sys(t,y)
global r gamma s delta;
if (y(6) - y(4)) * y(1) < 0
    u = 0 ;
elseif (y(6) - y(4)) * y(1) > 1
    u = 1 ;
else
    u = (y(6) - y(4)) * y(1);
end
ydot = [- r * y(1) * y(2) - u * y(1) + s * y(3);
r * y(1) * y(2) - gamma * y(2) - delta * y(2);
gamma * y(2) + u * y(1) - s * y(3);
r * y(2) * y(4) - r * y(2) * y(5) + u * y(4) - u * y(6);
r * y(1) * y(4) - r * y(1) * y(5) + (gamma + delta) * y(5) - gamma * y(6);
- s * (y(4) + y(6))] ;
end
```