

Capacité d'un vaccin à prévenir une épidémie dans une population

Mouad Obeidi
N° 21106



Plan

1

Le modèle S.I.R

2

Le modèle S.V.I.R

- Étude de la stabilité de l'équilibre sans maladie

3

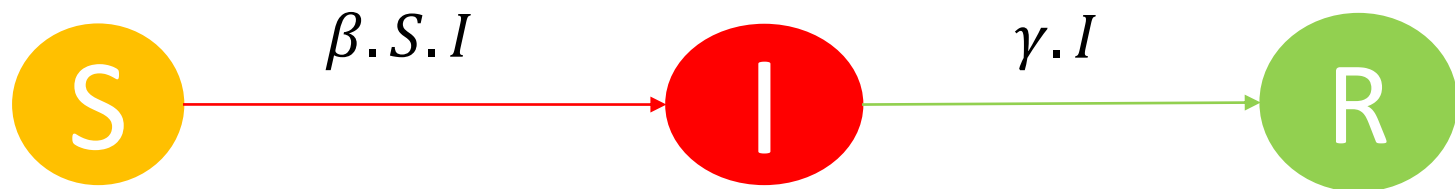
Étude d'une situation plus complexe

- Étude de la stabilité de l'équilibre sans maladie
- Le contrôle de l'épidémie par les mesures sanitaires

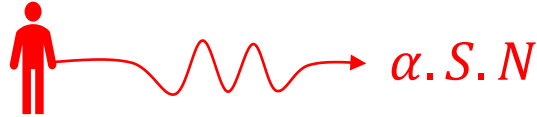
Hypothèses du modèle:

- ❑ La population garde un nombre constant
- ❑ Les gens rétablis ne s'infectent pas une deuxième fois

compartiment	Désignation
N	Nombre total de personnes $N = S + I + R$ ($N = 10^7$)
S	(nombre de personnes susceptibles à être infectées à l'instant t)/ N
I	(nombre de personnes infectées à l'instant t)/ N
R	(nombre de personnes rétablies à l'instant t)/ N



Le paramètre β



Une personne infectée

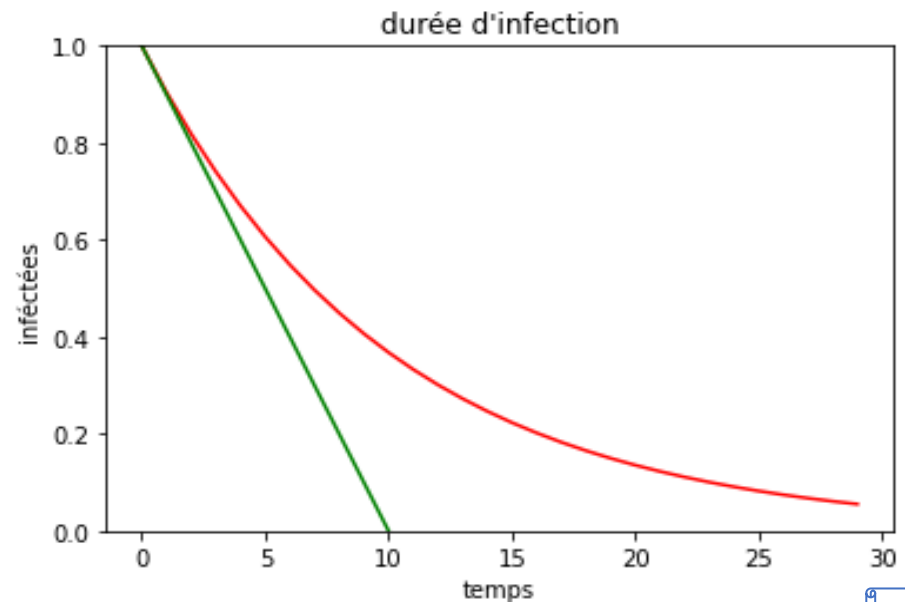
α	$0,25 \times 10^{-7}$	1/(jour x personne)
----------	-----------------------	---------------------

En posant $\beta = \alpha \cdot N$ le nombre total d'infectées engendrées par jour est $\alpha \cdot S \cdot N \cdot I \cdot N = \beta \cdot S \cdot I \cdot N$
le nombre total d'infectées engendrées par jour/N est $\beta \cdot S \cdot I$

Le paramètre γ

C'est l'inverse de la durée d'infection

Si on a $\frac{dI}{dt} = -\gamma \cdot I$



paramètre	Désignation	valeur	dimension
β	le taux d'infection	0,25	$[T]^{-1}$
γ	le taux de rétablissement	1/10	$[T]^{-1}$

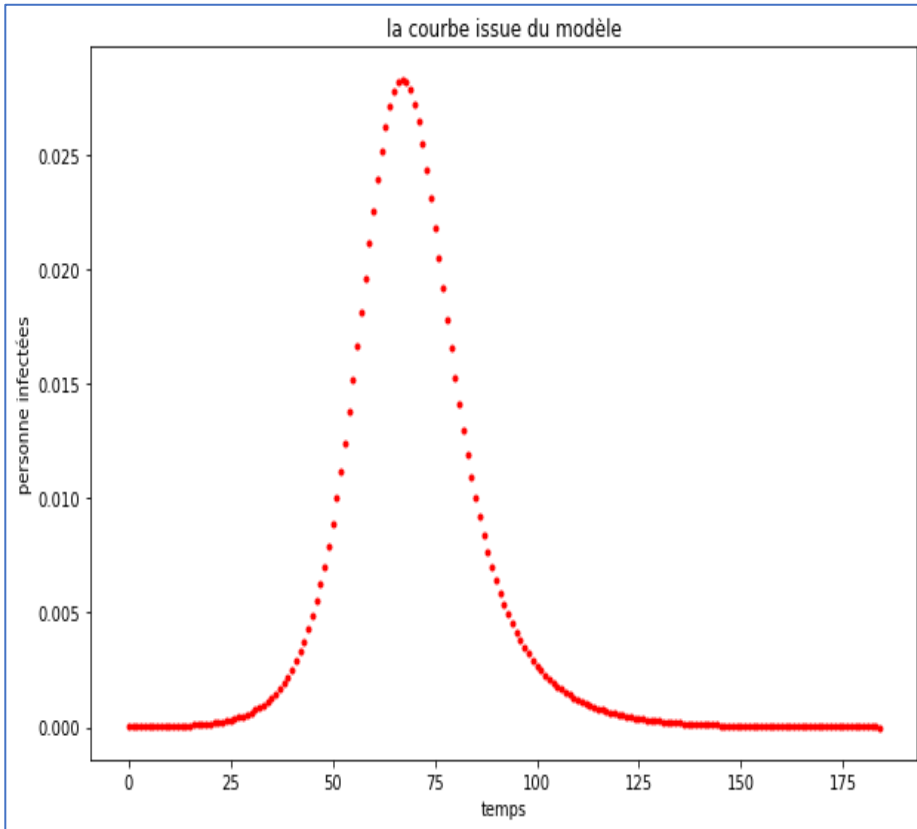
$$\frac{dS}{dt} = -\beta \cdot S \cdot I$$

$$\frac{dI}{dt} = \beta \cdot S \cdot I - \gamma \cdot I$$

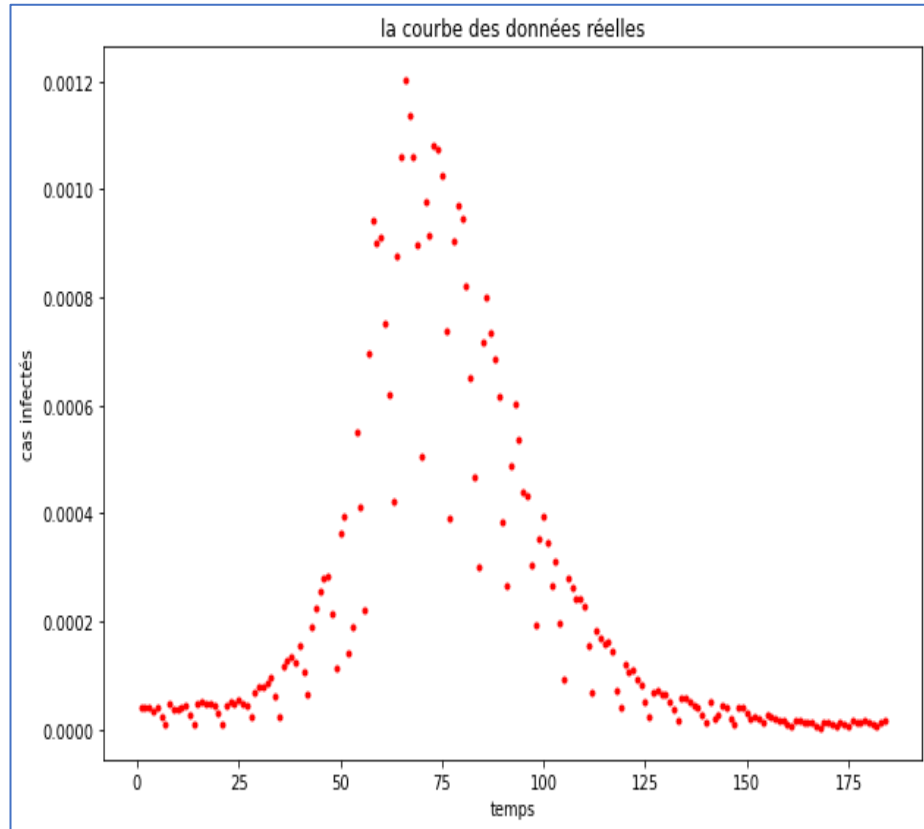
$$\frac{dR}{dt} = \gamma \cdot I$$

Comparaison du modèle avec la réalité

Courbe issue du modèle



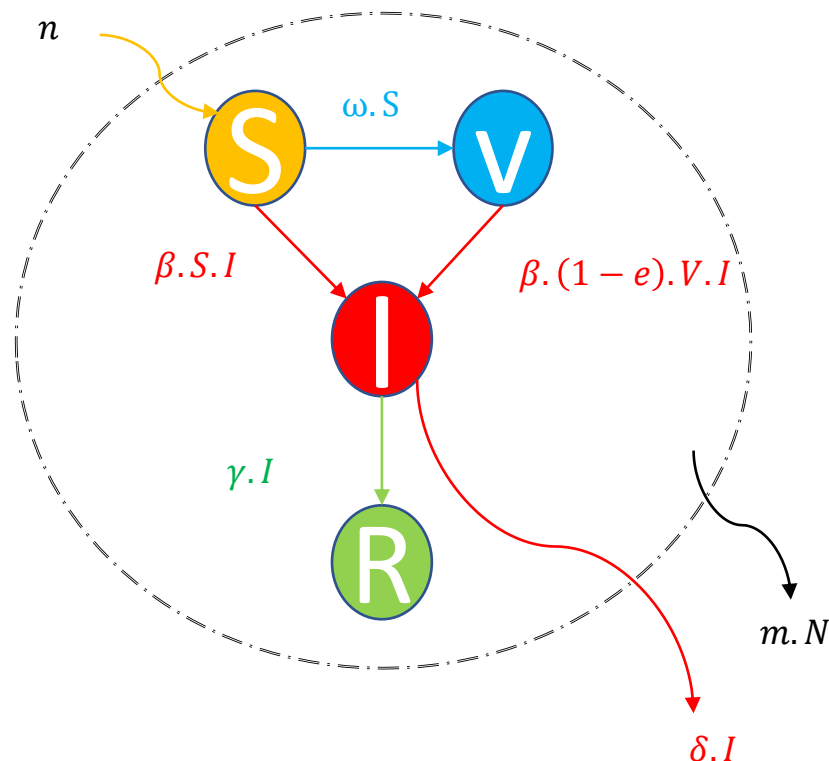
Courbe des données réelles



Code en annexe 1
les données sont fournies par « our world in data »

Modèle SVIR_[2]

ω	taux de vaccination	0,01	$[T]^{-1}$
e	Efficacité du vaccin	0,5	sans dimension
δ	Taux de mortalité du virus	1/50	$[T]^{-1}$
n	personne nais par jour divisé par N	2197×10^{-7}	$[T]^{-1}$
m	Taux de mortalité	$1/(80 \times 365)$	$[T]^{-1}$



$$\frac{dS}{dt} = n - \beta.S.I - \omega.S - m.S$$

$$\frac{dV}{dt} = \omega.S - \beta.(1 - e).V.I - m.V$$

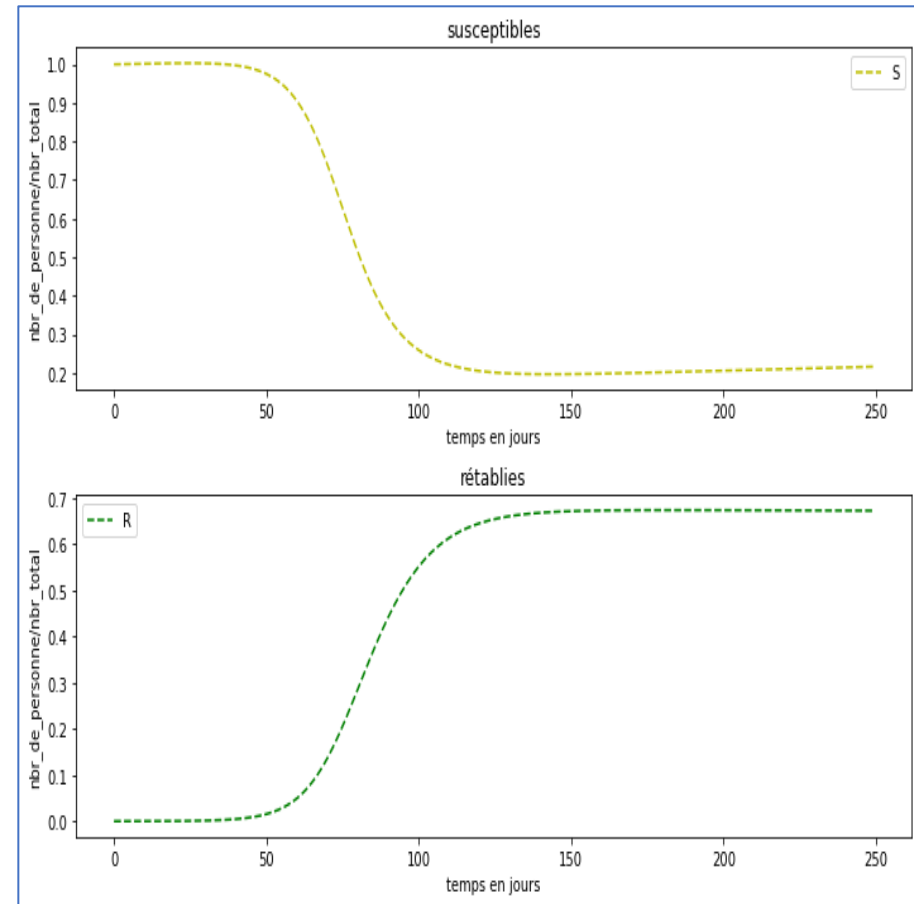
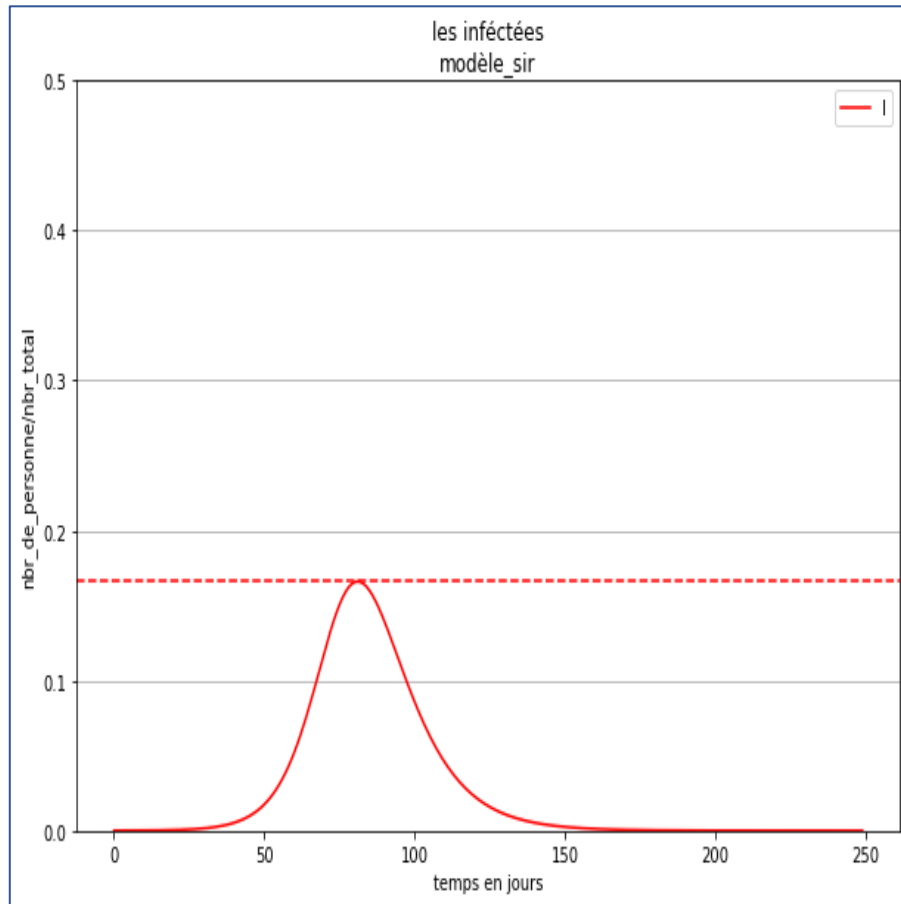
$$\frac{dI}{dt} = \beta.S.I + \beta.(1 - e).V.I - \gamma.I - \delta.I - m.I$$

$$\frac{dR}{dt} = \gamma.I - m.R$$

Mise en évidence de l'effet de la vaccination

Simulation du modèle SIR

Annexe 2

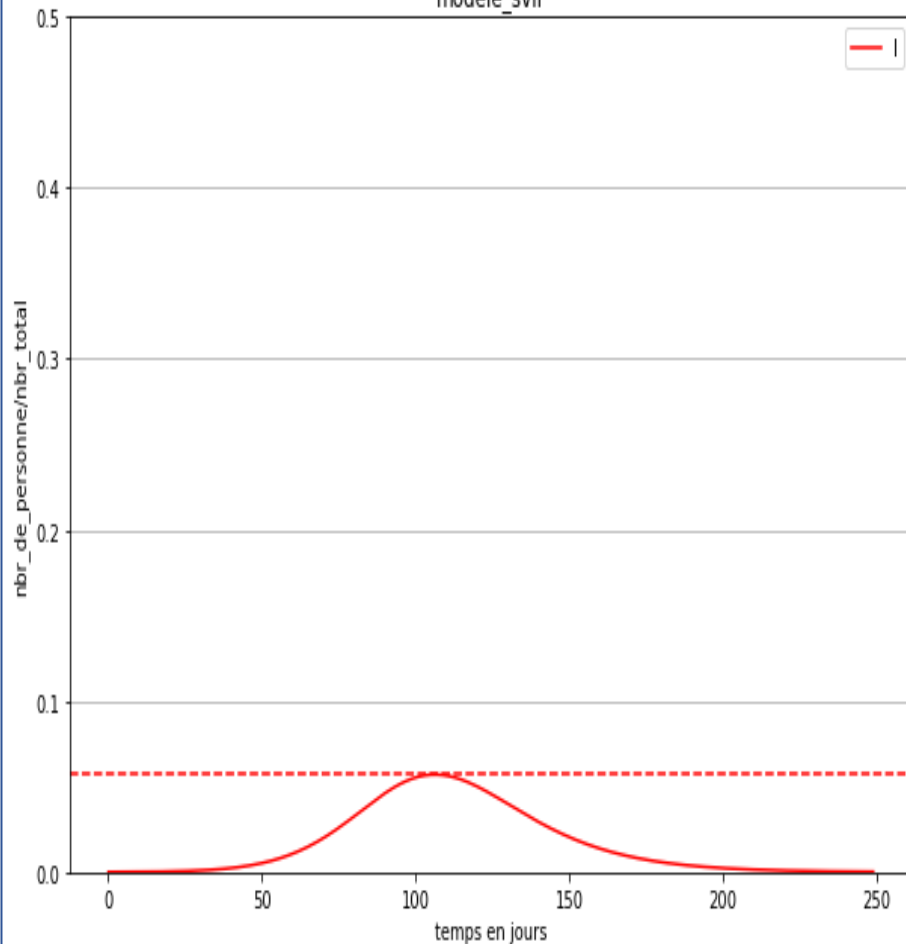


Mise en évidence de l'effet de la vaccination

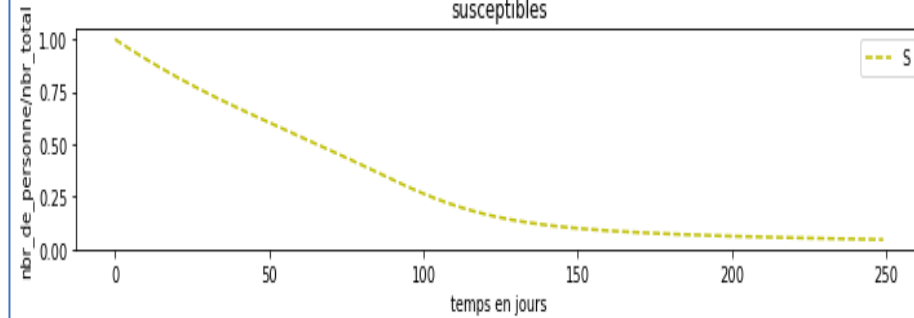
Simulation du modèle SVIR

Annexe 3

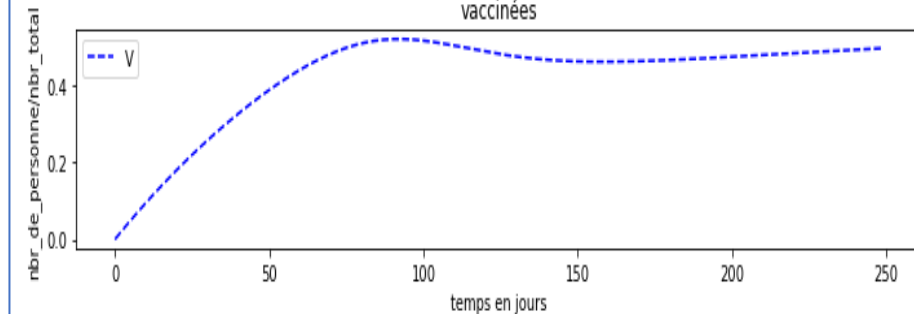
les infectées
modèle_svir



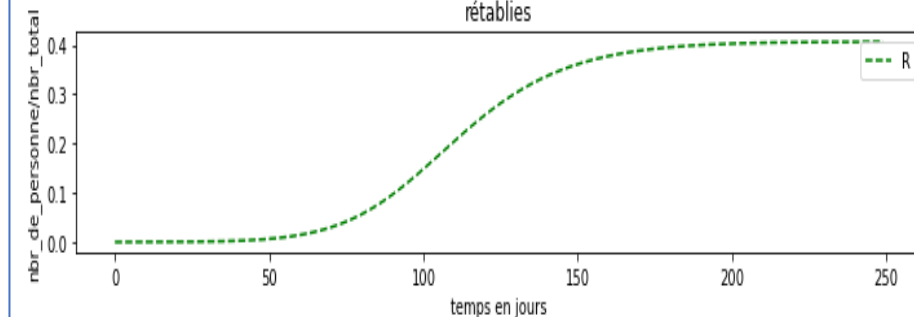
susceptibles



vaccinées



rétablies



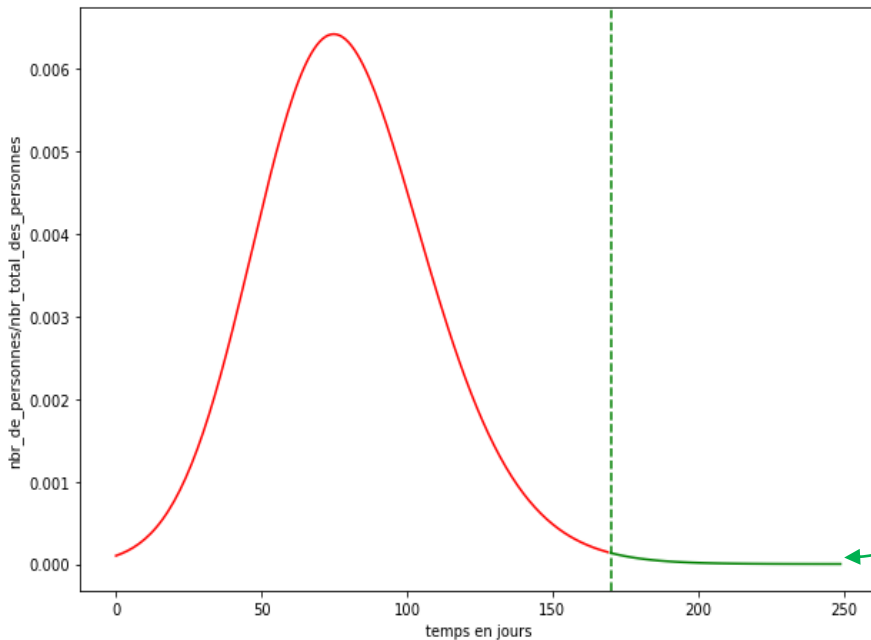
Problématique

Quelle est l'efficacité minimale d'un vaccin pour exterminer une épidémie ?

Il suffit de maintenir cet équilibre, c'est:

« l'Équilibre sans maladie (DFE) »

$$\frac{dS}{dt} = 0, \frac{dV}{dt} = 0, \frac{dI}{dt} = 0, \frac{dR}{dt} = 0$$



Equilibre sans maladie (DFE)

À l'équilibre:

$$\text{On a } \begin{cases} \frac{dS}{dt} = 0 \\ \frac{dV}{dt} = 0 \\ \frac{dI}{dt} = 0 \\ \frac{dR}{dt} = 0 \end{cases} \Rightarrow \begin{cases} n - \beta.S.I - \omega.S - m.S = 0 \\ \omega.S - \beta.(1-e).V.I - m.V = 0 \\ \beta.S.I + \beta.(1-e) - \gamma.I - \delta.I - m.I = 0 \\ \gamma.I - m.R = 0 \end{cases}$$

La solution qui correspond à l'équilibre sans maladie est telle que $I = 0$, donc:

$$S_{\text{éq}} = \frac{n}{\omega+m} \quad , \quad V_{\text{éq}} = \frac{\omega.n}{m.(\omega+m)} \quad \text{et} \quad I_{\text{éq}} = R_{\text{éq}} = 0$$

Pas d'infection

Equilibre sans maladie (DFE)

On veillera à ce que cet état d'équilibre soit stable c'est-à-dire que: pendant cet état d'équilibre, et après une petite perturbation (ajout d'une personne infectée) le système regagne l'équilibre sans maladie.

- Au voisinage de l'équilibre on peut considérer que :

$$S = S_{eq} \quad et \quad V = V_{eq} \quad et \quad R = R_{eq}$$

Donc au voisinage de l'équilibre on a:

$$\frac{dI}{dt} = \underbrace{(\beta \cdot S_{eq} + \beta \cdot (1 - e) \cdot V_{eq} - m - \gamma - \delta)}_{\text{Il suffit que ce coefficient soit négatif}} \cdot I$$

Il suffit que ce coefficient soit négatif

Où: $S_{eq} = \frac{n}{\omega + m}$ et $V_{eq} = \frac{\omega \cdot n}{m \cdot (\omega + m)}$

Ceci est équivalent à:

$$\frac{\beta n}{(m + \omega)(m + \gamma + \delta)} \left(1 + (1 - e) \frac{\omega}{m} \right) < 1$$

et note:

$$R = \frac{\beta n}{(m + \omega)(m + \gamma + \delta)} \left(1 + (1 - e) \frac{\omega}{m} \right)$$

C'est le taux de reproduction de base

Ou à:

$$e > \left[1 - \frac{(m + \omega)(m + \delta + \gamma)}{\beta n} \right] \frac{m}{\omega} + 1$$

Application numérique:

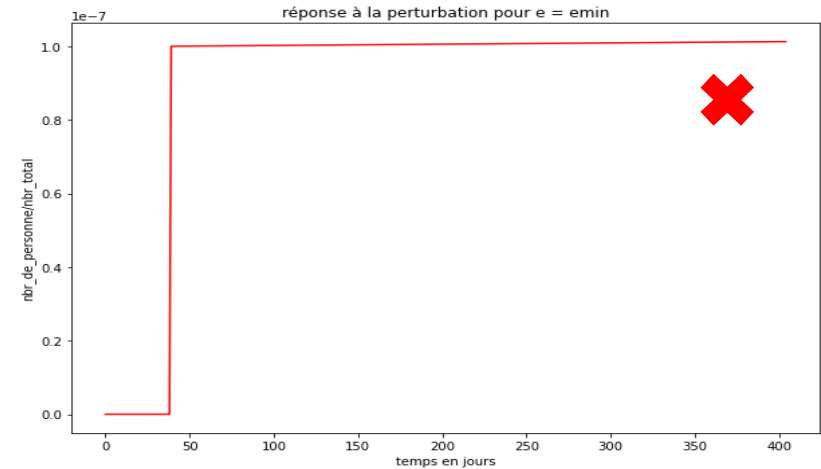
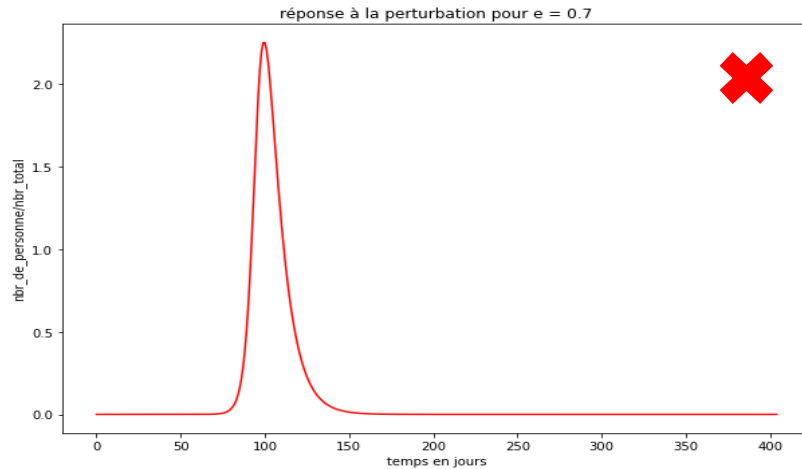
$$e > e_{min} = 0,92$$

Equilibre sans maladie (DFE)

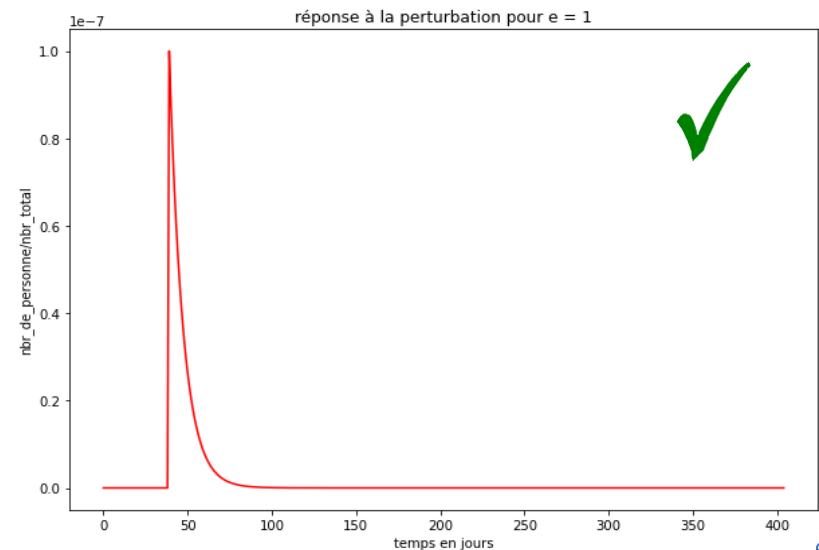
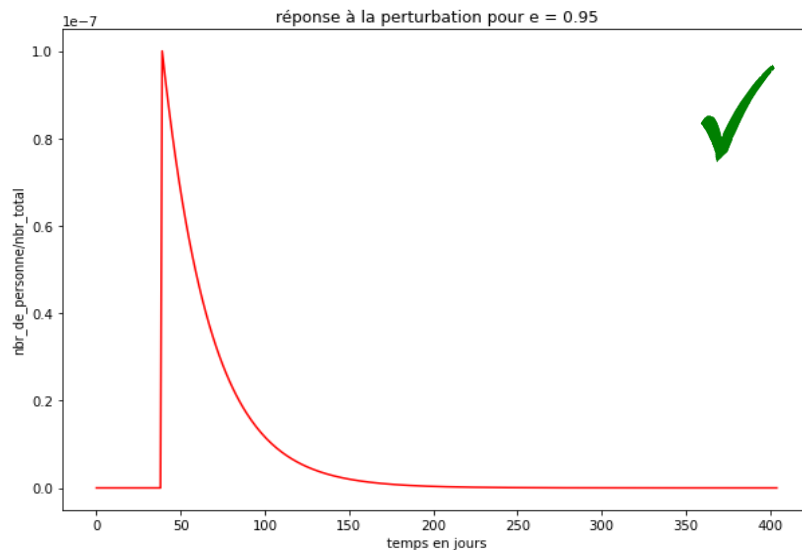
Simulation de la réponse du système en DFE à un ajout
d'une personne infectée

Annexe 4

pour $e \leq e_{min}$



pour $e > e_{min}$



Equilibre sans maladie (DFE)

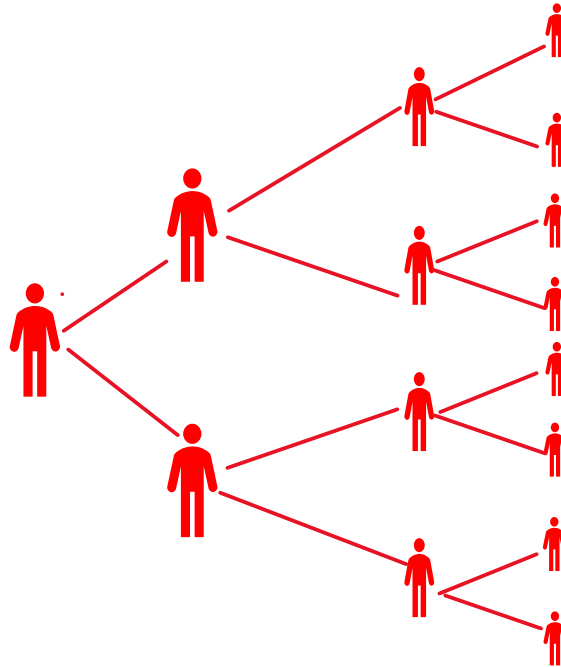
Interprétation

Le taux de reproduction de base R

Définition

C'est le nombre moyen de personnes infectées engendrées par une personne infectée

Exemple Pour $R_o = 2$:



Equilibre sans maladie (DFE)

Interprétation

Le taux de reproduction de base R

Pour que l'épidémie disparait il faut que $R < 1$

- La durée en jour d'infection d'une personne est : $\frac{1}{m+\delta+\gamma}$
- chaque jour elle infecte: $\beta \cdot S_{\text{éq}} + \beta \cdot (1 - e) \cdot V_{\text{éq}}$
- donc elle infecte: $\frac{1}{m+\delta+\gamma} \cdot (\beta \cdot S_{\text{éq}} + \beta \cdot (1 - e) \cdot V_{\text{éq}})$
- donc
$$R = \frac{\beta n}{(m+\omega)(m+\gamma+\delta)} \left(1 + (1 - e) \frac{\omega}{m} \right)$$

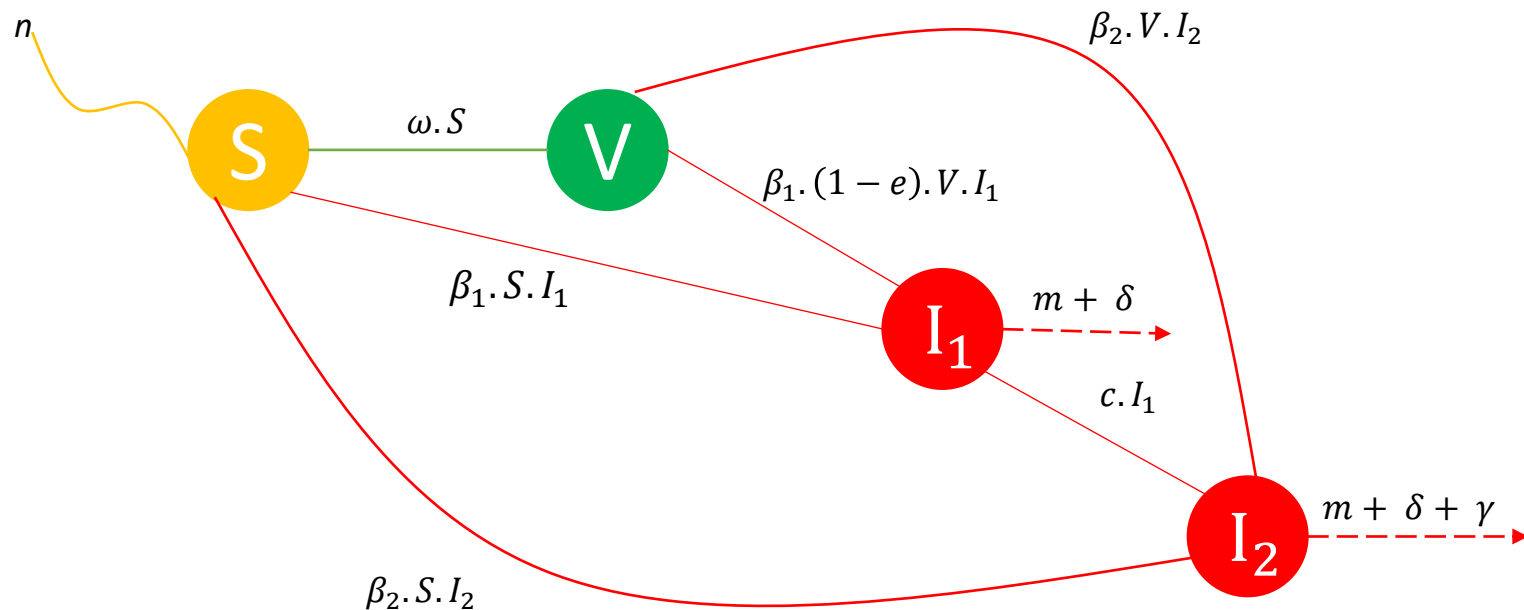
e	0,5	e_{\min}	0,85	1
R	2,18	1	0,79	0,19



Une situation plus complexe

C'est une situation où il existe deux compartiments de personnes infectées : car le virus, parfois, apparaît selon deux aspects, ceci aggrave la situation car le vaccin ne peut empêcher que contre un seul état du virus.

Modélisation



Une situation plus complexe

paramètre	Désignation	valeur	dimension
β_1	Taux d'infection relatif à I_1	0,25	$[T]^{-1}$
β_2	Taux d'infection relatif à I_2	0,025	$[T]^{-1}$
c	taux de transmission du compartiment I_1 vers I_2	0,1	$[T]^{-1}$

Le modèle élaboré

$$\frac{dS}{dt} = n - \omega \cdot S - (\beta_1 \cdot I_1 + \beta_2 \cdot I_2) \cdot S - m \cdot S$$

$$\frac{dV}{dt} = \omega \cdot S - \beta_1 \cdot (1 - e) \cdot V \cdot I_1 - \beta_2 \cdot V \cdot I_2 - m \cdot V$$

$$\frac{dI_1}{dt} = \beta_1 \cdot (S + (1 - e) \cdot V) \cdot I_1 - (m + \delta + c) \cdot I_1$$

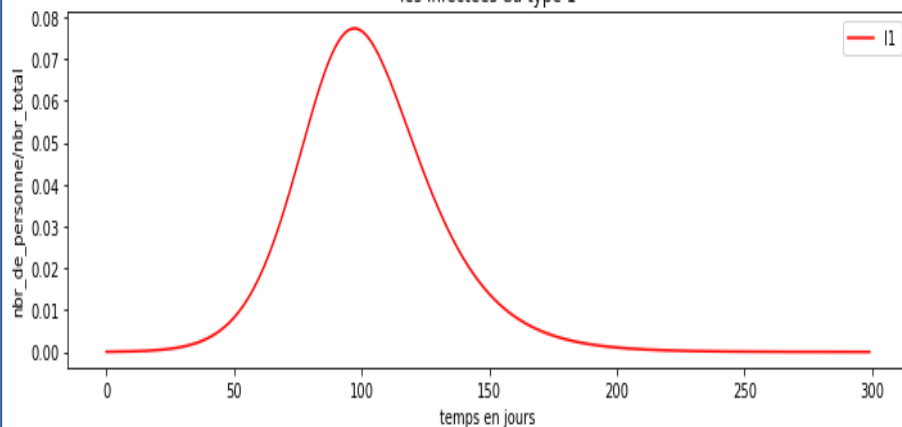
$$\frac{dI_2}{dt} = c \cdot I_1 + \beta_2 \cdot (S + V) \cdot I_2 - (m + \delta + \gamma) \cdot I_2$$

Une situation plus complexe

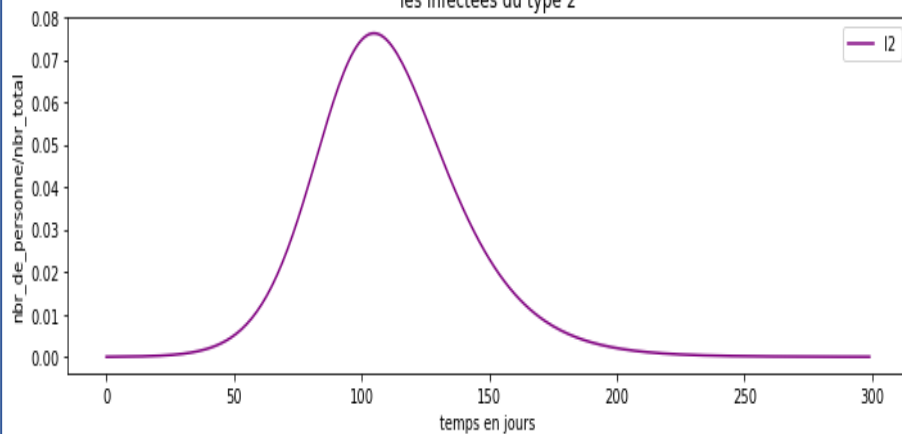
Simulation des courbes issues du modèle

Annexe 5

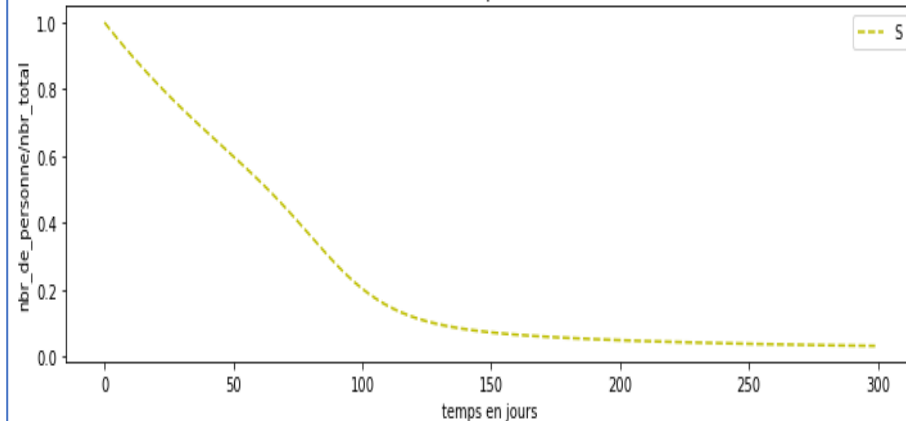
les infectées du type 1



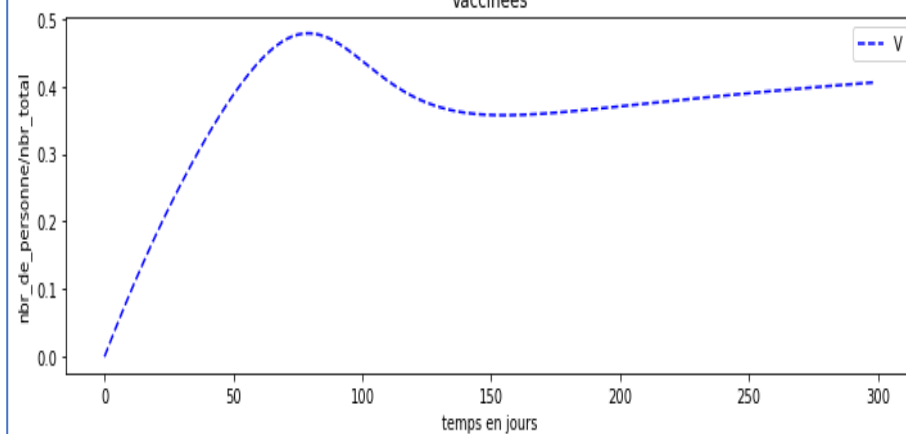
les infectées du type 2



susceptibles



vaccinées



Equilibre sans maladie (DFE)

Il existe encore un état d'équilibre sans maladie on cherchera aussi à ce qu'il soit stable

À l'équilibre:

$$\begin{cases} \frac{dS}{dt} = 0 \\ \frac{dV}{dt} = 0 \\ \frac{dI_1}{dt} = 0 \\ \frac{dI_2}{dt} = 0 \end{cases} \Rightarrow \begin{cases} n - \omega \cdot S - (\beta_1 \cdot I_1 + \beta_2 \cdot I_2) \cdot S - m \cdot S = 0 \\ \omega \cdot S - \beta_1 \cdot (1 - e) \cdot V \cdot I_1 - \beta_2 \cdot V \cdot I_2 - m \cdot V = 0 \\ \beta_1 \cdot (S + (1 - e) \cdot V) \cdot I_1 - (m + \delta + c) \cdot I_1 = 0 \\ c \cdot I_1 + \beta_2 \cdot (S + V) \cdot I_2 - (m + \delta + \gamma) \cdot I_2 = 0 \end{cases}$$

La solution qui correspond à l'équilibre sans maladie est telle que $I_1 = I_2 = 0$

Donc la solution correspondante est :

$$S_{\text{éq}} = \frac{n}{\omega + m} \quad , \quad V_{\text{éq}} = \frac{\omega \cdot n}{m \cdot (\omega + m)} \quad \text{et} \quad I_{1,\text{éq}} = I_{2,\text{éq}} = 0$$

Equilibre sans maladie (DFE)

Il suffit que le taux de reproduction R soit inférieur à 1

Problématique : comment calculer le taux de reproduction R dans ce cas ?

on appliquera la méthode de ***p. Van Den Driessche*** et ***James Watmough*** [3],[4]

On déterminera une matrice \mathbf{M} qui s'appelle la matrice de la prochaine génération puis:

$$R = \max_{\lambda \in \text{sp}(\mathbf{M})} |\lambda|$$

R est le rayon spectrale de la matrice \mathbf{M} on note $R = \rho(\mathbf{M})$

p. van den Driessche et James Watmough deux spécialistes en mathématiques appliquées leurs sujets de recherche comprennent la biomathématique, l'analyse matricielle et la théorie de la stabilité.

Notations

Soit $x = (v_1, v_2, v_3, v_4)$ et $x_{\text{éq}} = (S_{\text{éq}}, V_{\text{éq}}, 0, 0)$:

Le modèle peut être écrit de la forme suivante:

$$\frac{dx}{dt} = g(x)$$

Où

$$g(x) = (g_1(x), g_2(x), g_3(x), g_4(x))$$

Et

$$g_1(x) = n - \omega \cdot S - (\beta_1 \cdot I_1 + \beta_2 \cdot I_2) \cdot S - m \cdot S$$

$$g_2(x) = \omega \cdot S - \beta_1 \cdot (1 - e) \cdot V \cdot I_1 - \beta_2 \cdot V \cdot I_2 - m \cdot V$$

$$g_3(x) = \beta_1 \cdot (S + (1 - e) \cdot V) \cdot I_1 - (m + \delta + c) \cdot I_1$$

$$g_4(x) = c \cdot I_1 + \beta_2 \cdot (S + V) \cdot I_2 - (m + \delta + \gamma) \cdot I_2$$

La fonction g peut être décomposée de la façon suivante:

$$g = f + v$$

$$f = (f_1, f_2, f_3, f_4)$$

$$v = (v_1, v_2, v_3, v_4)$$

Où f_1, f_2, f_3, f_4 décrivent l'apparition des nouveaux infectés respectivement dans les compartiments S, V, I_1, I_2 ($f_1 = f_2 = 0$)

Et v_1, v_2, v_3, v_4 décrivent la sortie ou l'entrée par toute autre manière (mort, vaccination...) respectivement du ou vers les compartiments v_1, v_2, v_3, v_4

Matrice de la prochaine génération

1^{ère} étape

Identifier les fonctions f_i et v_i nécessaires

Au voisinage de l'équilibre on considère que : $S = S_{\text{éq}}$, $V = V_{\text{éq}}$, $I_1 \ll 1$ et $I_2 \ll 1$

Pour x au voisinage de $x_{\text{éq}}$:

$$g_3(x) = \underbrace{\beta_1 \cdot (S_{\text{éq}} + (1 - e) \cdot V_{\text{éq}}) \cdot I_1}_{f_3(x)} - \underbrace{(m + \delta + c) \cdot I_1}_{v_3(x)}$$

$$g_4(x) = \underbrace{\beta_2 \cdot (S_{\text{éq}} + V_{\text{éq}}) \cdot I_2}_{f_4(x)} + \underbrace{c \cdot I_1 - (m + \delta + \gamma) \cdot I_2}_{v_4(x)}$$

$$f_3(x) = \beta_1 \cdot (S_{\text{éq}} + (1 - e) \cdot V_{\text{éq}}) \cdot I_1$$

$$v_3(x) = -(m + \delta + c) \cdot I_1$$

$$f_4(x) = \beta_2 \cdot (S_{\text{éq}} + V_{\text{éq}}) \cdot I_2$$

$$v_4(x) = c \cdot I_1 - (m + \delta + \gamma) \cdot I_2$$

Matrice de la prochaine génération

2^{ème} étape

Calculer les matrices jacobiennes des fonctions f et v au point $x_{éq}$

Les matrices jacobienne de f et de v ont les formes suivantes

$$J_f(x_{éq}) = \begin{pmatrix} 0 & 0 \\ 0 & F \end{pmatrix} \quad J_v(x_{éq}) = \begin{pmatrix} J_1 & J_2 \\ 0 & V \end{pmatrix}$$

$$F = \begin{pmatrix} \beta_1 \cdot (S_{éq} + (1 - e) \cdot V_{éq}) & 0 \\ 0 & \beta_2 \cdot (S_{éq} + V_{éq}) \end{pmatrix}$$

et

$$V = \begin{pmatrix} -(m + \delta + c) & 0 \\ c & -(m + \delta + \gamma) \end{pmatrix}$$

Matrice de la prochaine génération

3^{ème} étape

Calculer la matrice $-FV^{-1}$

$$V^{-1} = \begin{pmatrix} \frac{-1}{(m + \delta + c)} & 0 \\ \frac{-c}{(m + \delta + c)(m + \delta + \gamma)} & \frac{-1}{(m + \delta + \gamma)} \end{pmatrix}$$

$$-FV^{-1} = \begin{pmatrix} \frac{\beta_1 \cdot (S_{\text{éq}} + (1 - e) \cdot V_{\text{éq}})}{(m + \delta + c)} & 0 \\ \frac{\beta_2 \cdot (S_{\text{éq}} + V_{\text{éq}}) \cdot c}{(m + \delta + c)(m + \delta + \gamma)} & \frac{\beta_2 \cdot (S_{\text{éq}} + V_{\text{éq}})}{(m + \delta + \gamma)} \end{pmatrix}$$

*Cette matrice
s'appelle la
matrice de la
prochaine
génération*

Matrice de la prochaine génération

Interprétation

Si on a des nombres I_1 et I_2 d'infectées

$$\begin{aligned} -FV^{-1} \begin{pmatrix} I_1 \\ I_2 \end{pmatrix} &= \begin{pmatrix} \frac{\beta_1 \cdot (S_{\text{éq}} + (1-e) \cdot V_{\text{éq}})}{(m + \delta + c)} & 0 \\ \frac{\beta_2 \cdot (S_{\text{éq}} + V_{\text{éq}}) \cdot c}{(m + \delta + c)(m + \delta + \gamma)} & \frac{\beta_2 \cdot (S_{\text{éq}} + V_{\text{éq}})}{(m + \delta + \gamma)} \end{pmatrix} \begin{pmatrix} I_1 \\ I_2 \end{pmatrix} \\ &= \begin{pmatrix} \frac{\beta_1 \cdot (S_{\text{éq}} + (1-e) \cdot V_{\text{éq}})}{(m + \delta + c)} \cdot I_1 \\ \frac{\beta_2 \cdot (S_{\text{éq}} + V_{\text{éq}}) \cdot c}{(m + \delta + c)(m + \delta + \gamma)} \cdot I_1 + \frac{\beta_2 \cdot (S_{\text{éq}} + V_{\text{éq}})}{(m + \delta + \gamma)} \cdot I_2 \end{pmatrix} \end{aligned}$$

Les infectées engendrées par I_1

Les infectées engendrées par I_2

Stabilité du DFE

La condition de stabilité est $R < 1$

Les valeurs propres de $-FV^{-1}$ sont :

$$\lambda_1 = \frac{\beta_1 \cdot (S_{\text{éq}} + (1 - e) \cdot V_{\text{éq}})}{(m + \delta + c)}$$

et

$$\lambda_2 = \frac{\beta_2 \cdot (S_{\text{éq}} + V_{\text{éq}})}{(m + \delta + \gamma)} = 1,45$$

$$\text{Et } R = \max(|\lambda_1|, |\lambda_2|)$$

Donc il faut que $|\lambda_1| < 1$ et $|\lambda_2| < 1$

Donc il faut que

$$e > \frac{1}{V_{\text{éq}}} \left(S_{\text{éq}} - \frac{(m + \delta + c)}{\beta_1} \right) + 1 \quad \text{A.N} \quad e > 0,93$$

λ_2 dépasse 1 on devra faire appelle au mesure sanitaire à savoir le confinement

$$\frac{dI_1}{dt} = \beta_1 \cdot (S + (1 - e) \cdot V) \cdot I_1 - (m + \delta + c) \cdot I_1$$

$$\frac{dI_2}{dt} = c \cdot I_1 + \beta_2 \cdot (S + V) \cdot I_2 - (m + \delta + \gamma + q) \cdot I_2$$

donc

$$\lambda_2 = \frac{\beta_2 \cdot (S_{\text{éq}} + V_{\text{éq}})}{(m + \delta + \gamma + q)}$$

Dans ce cas il faut que $q > \beta_2 \cdot (S_{\text{éq}} + V_{\text{éq}}) - (m + \delta + \gamma)$ A.N $q > 0,05$

Stabilité du DFE

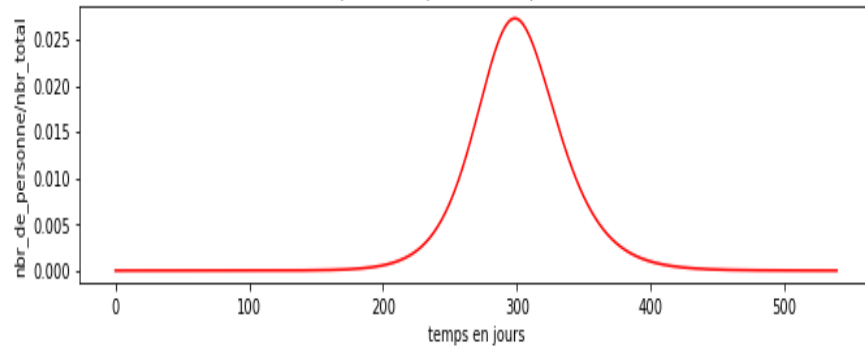
Simulation de la réponse du système en DFE à un ajout d'une personne infectée

Annexe 6

sans confinement

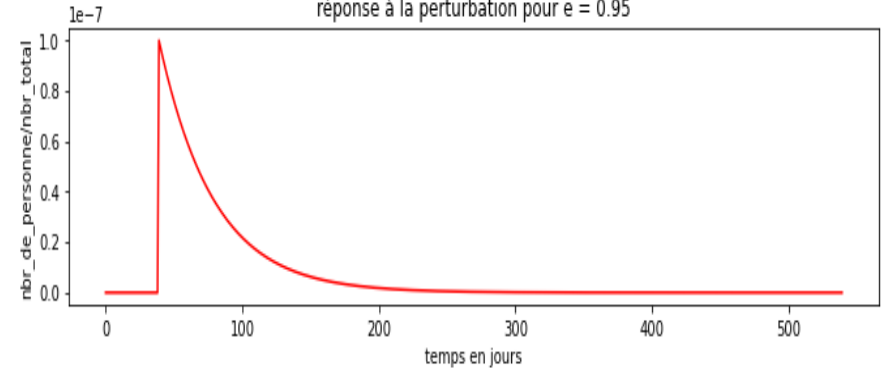
pour $e \leq e_{min}$

réponse à la perturbation pour $e = 0.9$

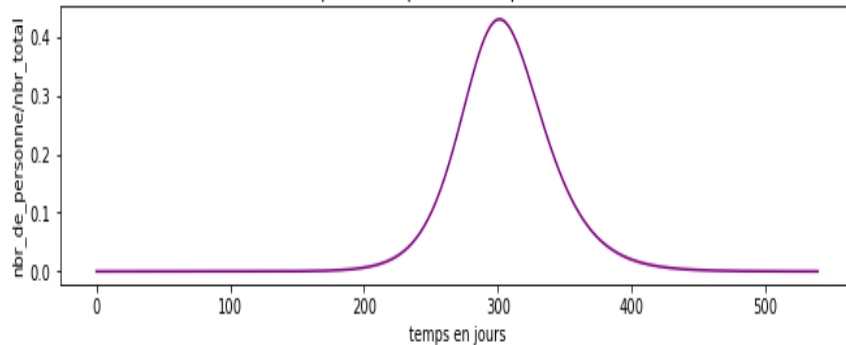


pour $e > e_{min}$

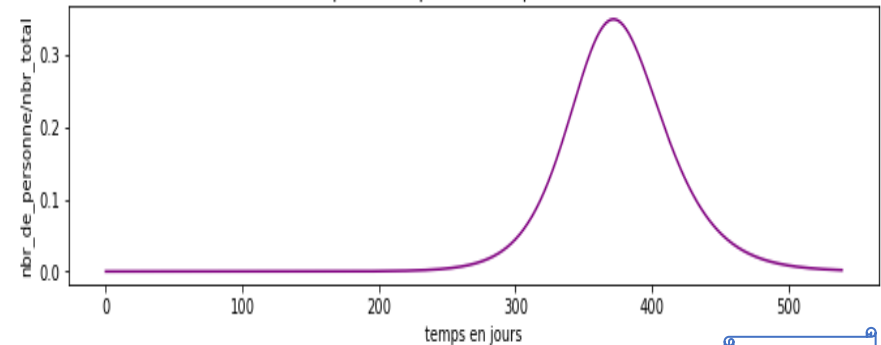
réponse à la perturbation pour $e = 0.95$



réponse à la perturbation pour $e = 0.9$



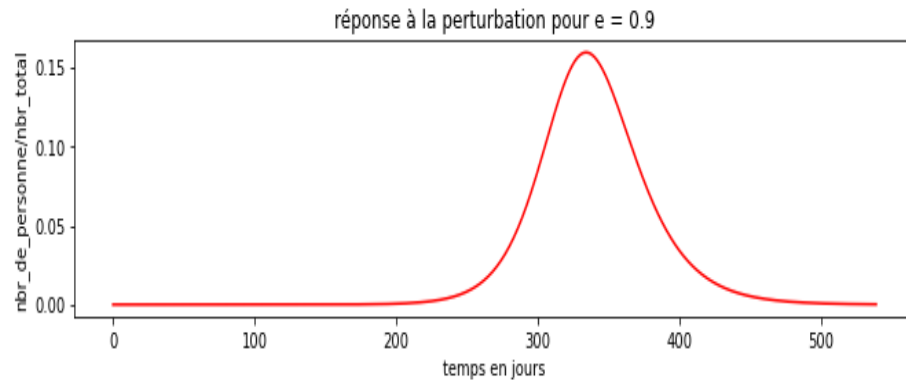
réponse à la perturbation pour $e = 0.95$



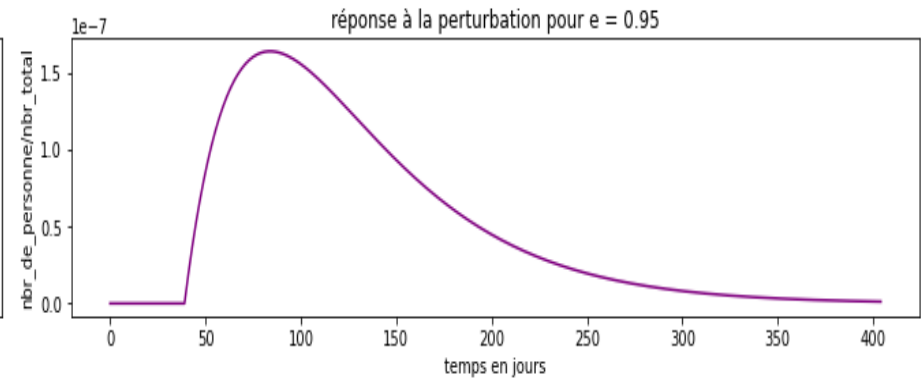
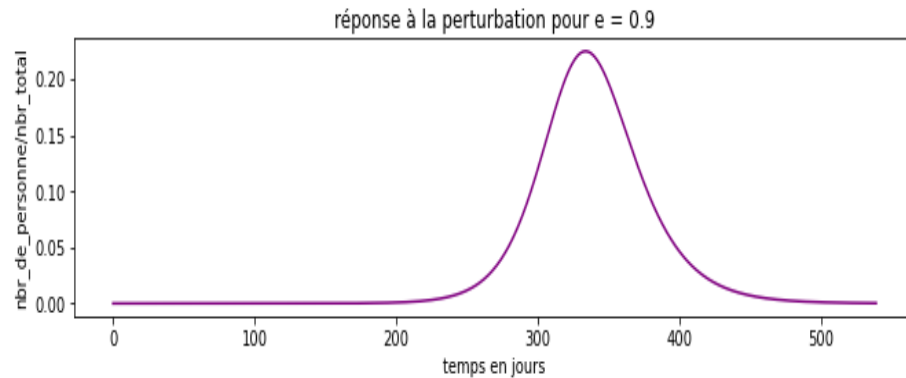
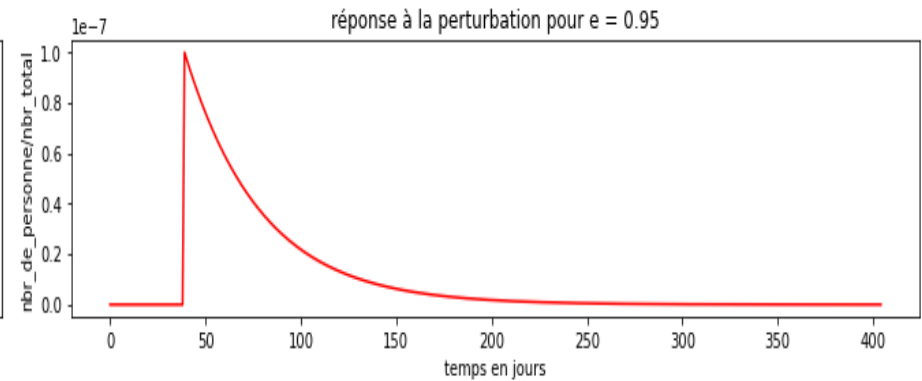
Stabilité du DFE

Avec confinement $q = 0,06$

pour $e \leq e_{min}$



pour $e > e_{min}$



Synthèse

Le contrôle de l'épidémie peut se faire par la modélisation mathématique, il s'agit en premier lieu à modéliser sa propagation, d'estimer ses paramètres, et d'appliquer la méthode de **p. Van Den Driessche** et **James Watmough** pour calculer R et essayer de le maintenir inférieur à 1

Contraintes

Le cas réel est plus complexe:

- Les paramètres ne peuvent pas être parfaitement estimés
- De plus ces paramètres ne sont pas constants le long de la période d'étude
- La propagation du virus est plus complexe, les personnes rétablies peuvent se réinfecter

Fin

Merci pour votre attention

```
#courbe_des_infectés_par_modélisation_sir
```

```
import numpy as np
import matplotlib.pyplot as plt
S = [] ; V = [] ; I = [] ; NI = [] ; R = []
Sn = 1 - 394*10**-7 ; In = 394*10**-7 ; NIn = 394*10**-7 ; Rn = 0
beta = 0.25 ; gamma = 1/10
for i in range(185):
    S.append(Sn)
    I.append(In)
    NI.append(NIn)
    R.append(Rn)
    Sn = Sn - beta*Sn*In
    In = In + beta*Sn*In - gamma*In
    NIn = beta*Sn*In
    Rn = Rn + gamma*In
t = np.arange(0,185,1)
plt.plot(t,NI,'.',color='r')
plt.xlabel('temps')
plt.ylabel('personne infectées')
plt.title('la courbe issue du modèle')
fig = plt.gcf()
fig.set_size_inches(10,7)
plt.show()
```

```
#courbe réelle
```

```
import sqlite3
#connection à la base et accès aux données
db = sqlite3.connect("C:/Users/smart asus/Desktop/mon_tipe/BD_1.db")
cur = db.cursor()
R1 = '''SELECT new_cases*0.0000001
        FROM Bd_1
        WHERE date >= '2021-06-01' and date<= '2021-12-01' '''
cur.execute(R1)
#traçage de la courbe réelle
import numpy as np
import matplotlib.pyplot as plt
NI = []
for e in cur:
    NI.append(e[0])
t = np.arange(1,len(NI)+1)
plt.plot(t,NI,'.',color='r')
plt.xlabel('temps')
plt.ylabel('cas infectés')
plt.title('la courbe des données réelles')
fig = plt.gcf()
fig.set_size_inches(10,7)
plt.show()
```

```

#modèle_sir:
import numpy as np
import matplotlib.pyplot as plt
S = [] ; I = [] ; R = []
Sn = 1-394*10**(-7) ; In = 394*10**(-7) ; Rn = 0
beta = 0.25 ; gamma = 1/10 ; m = 1/(80*365) ; n = 2197*10**-7;
delta = 1/50
for i in range(250):
    S.append(Sn)
    I.append(In)
    R.append(Rn)
    Sn = Sn - beta*Sn*In - m*Sn + n
    In = In + beta*Sn*In - gamma*In - m*In - delta*In
    Rn = Rn + gamma*In - m*Rn
t = np.arange(0,250,1)
#I
plt.plot(t,I,'r',label='I')
plt.axhline(y = max(I),ls = '--',color='r')
plt.xlabel('temps en jours')
plt.ylabel('nbr_de_personne/nbr_total')
plt.legend()
plt.ylim([0,0.5])
plt.grid(which='major', axis='y')
plt.title('les infectées\nmodèle_sir')
fig = plt.gcf()
fig.set_size_inches(10,7)
plt.show()

```

```

#S
plt.subplot(211)
plt.plot(t,S,'y',ls = '--',label='S')
plt.title('susceptibles')
plt.xlabel('temps en jours')
plt.ylabel('nbr_de_personne/nbr_total')
plt.legend()
#R
plt.subplot(212)
plt.plot(t,R,'g',ls = '--',label='R')
plt.xlabel('temps en jours')
plt.ylabel('nbr_de_personne/nbr_total')
plt.title('rétablies')
plt.legend()
fig = plt.gcf()
fig.set_size_inches(10,7)
fig.tight_layout()
plt.show()

```

```

#modèle_svir:
import numpy as np
import matplotlib.pyplot as plt
S = [] ; V = [] ; I = [] ; R = []
Sn = 1-394*10**(-7) ; In = 394*10**(-7) ; Vn = 0 ; Rn = 0
beta = 0.25 ; gamma = 1/10 ; m = 1/(80*365) ; n = 2197*10**-7;
delta = 1/50 ; e = 0.5 ; w = 0.01
for i in range(250):
    S.append(Sn)
    V.append(Vn)
    I.append(In)
    R.append(Rn)
    Sn = Sn - beta*Sn*In - w*Sn - m*Sn + n
    Vn = Vn + w*Sn - beta*(Vn - e*Vn)*In - m*Vn
    In = In + beta*Sn*In + beta*(Vn - e*Vn)*In - gamma*In - m*In - delta*In
    Rn = Rn + gamma*In - m*Rn
t = np.arange(0,250,1)
#I
plt.plot(t,I,'r',label='I')
plt.axhline(y = max(I),ls = '--',color='r')
plt.xlabel('temps en jours')
plt.ylabel('nbr_de_personne/nbr_total')
plt.legend()
plt.ylim([0,0.5])
plt.grid(which='major', axis='y')
plt.title('les infectées\nmodèle_svir')
fig = plt.gcf()
fig.set_size_inches(10,7)
plt.show()

```

```

#S
plt.subplot(311)
plt.plot(t,S,'y',ls = '--',label='S')
plt.title('susceptibles')
plt.xlabel('temps en jours')
plt.ylabel('nbr_de_personne/nbr_total')
plt.legend()
#V
plt.subplot(312)
plt.plot(t,V,'b',ls = '--',label='V')
plt.title('vaccinées')
plt.xlabel('temps en jours')
plt.ylabel('nbr_de_personne/nbr_total')
plt.ylabel('nbr_de_personne/nbr_total')
plt.legend()
#R
plt.subplot(313)
plt.plot(t,R,'g',ls = '--',label='R')
plt.xlabel('temps en jours')
plt.ylabel('nbr_de_personne/nbr_total')
plt.title('rétablies')
plt.legend()
fig = plt.gcf()
fig.set_size_inches(10,7)
fig.tight_layout()
plt.show()

```

#étude de l'équilibre du DFE

```

import numpy as np
import matplotlib.pyplot as plt
beta = 0.25 ; gamma = 1/10 ; m = 1/(80*365) ; n = 2197*10**-7;
delta = 1/50 ; w = 0.01
Seq = n/(m+w)
Veq = (w/m)*Seq
emin = (1-(m+w)*(m+delta+gamma)/(beta*n))*(m/w)+1
e = ?
R0 = (1/(m+delta+gamma)*(m+w))*(beta*n*(1+(1-e)*w/m))
S = [Seq for k in range(40)]
V = [Veq for k in range(40)]
I = [0 for k in range(39)]+[10**-7]
R = [0 for k in range(40)]
for l in range(365):
    s = len(S) ; v = len(V) ; i = len(I) ; r = len(R)
    Sn = S[s-1] ; Vn = V[v-1] ; In = I[i-1] ; Rn = R[r-1]
    Sn = Sn - beta*Sn*In - w*Sn - m*Sn + n
    Vn = Vn + w*Sn - beta*(Vn - e*Vn)*In - m*Vn
    In = In + beta*Sn*In + beta*(Vn - e*Vn)*In - gamma*In - delta*In
    Rn = Rn + gamma*In - m*Rn
    S.append(Sn)
    V.append(Vn)
    I.append(In)
    R.append(Rn)
t = np.arange(0,405,1)
fig = plt.gcf()
plt.plot(t,I,'r',label='I')
plt.xlabel('temps en jours')
plt.ylabel('nbr_de_personne/nbr_total')
plt.title('réponse à la perturbation pour e = ?')
fig.set_size_inches(10,7)

```

```
#une situation plus complexe : deux compartiments infectées
```

```
import numpy as np
import matplotlib.pyplot as plt
S = [] ; V = [] ; I1 = [] ; I2 = [] ; R = []
Sn = 1-394*10**(-7) ; I1n = 394*10**(-7) ; I2n = 0 ; Vn = 0 ; Rn = 0
beta1 = 0.25 ; beta2 = 0.025 ; gamma = 1/10 ; m = 1/(80*365) ; n = 2197*10**-7
delta = 0.01 ; e = 0.5 ; w = 0.01 ; x = 300 ; c = 1/10
for i in range(x):
    S.append(Sn)
    V.append(Vn)
    I1.append(I1n)
    I2.append(I2n)
    Sn = Sn + n - (beta1*I1n + beta2*I2n)*Sn - w*Sn - m*Sn
    Vn = Vn + w*Sn - beta1*(1-e)*Vn*I1n - beta2*Vn*I2n - m*Vn
    I1n = I1n + beta1*(Sn + (1-e)*Vn)*I1n - (m + delta + c)*I1n
    I2n = I2n + c*I1n + beta2*(Sn + Vn)*I2n - (m + delta + gamma)*I2n
t = np.arange(0,x,1)
#I1
plt.subplot(211)
plt.plot(t,I1,'r',label='I1')
plt.xlabel('temps en jours')
plt.ylabel('nbr_de_personne/nbr_total')
plt.title('les infectées du type 1')
plt.legend()
---
```

```
#I2
plt.subplot(212)
plt.plot(t,I2,color = 'purple',label='I2')
plt.xlabel('temps en jours')
plt.ylabel('nbr_de_personne/nbr_total')
plt.legend()
plt.title('les infectées du type 2')
fig = plt.gcf()
fig.set_size_inches(10,7)
plt.tight_layout()
plt.show()
#S
plt.subplot(211)
plt.plot(t,S,'y',ls = '--',label='S')
plt.title('susceptibles')
plt.xlabel('temps en jours')
plt.ylabel('nbr_de_personne/nbr_total')
plt.legend()
#V
plt.subplot(212)
plt.plot(t,V,'b',ls = '--',label='V')
plt.title('vaccinées')
plt.xlabel('temps en jours')
plt.ylabel('nbr_de_personne/nbr_total')
plt.ylabel('nbr_de_personne/nbr_total')
plt.legend()
fig = plt.gcf()
fig.set_size_inches(10,7)
plt.tight_layout()
plt.show()
```

```
#stabilité du dfe pour la situation complexe
import numpy as np
import matplotlib.pyplot as plt
beta1 = 0.25 ; beta2 = 0.025 ; gamma = 1/10 ; m = 1/(80*365) ; n = 2197*10**-7
delta = 0.01 ; w = 0.01 ; c = 1/10
Seq = n/(m+w)
Veq = (w/m)*Seq
emin = (Seq - (m+delta+c)/beta1)/Veq + 1
e = 0.9
S = [Seq for k in range(40)]
V = [Veq for k in range(40)]
I1 = [0 for k in range(39)]+[10**-7]
I2 = [0 for k in range(40)]
for l in range(500):
    s = len(S) ; v = len(V) ; i1 = len(I1) ; i2 = len(I2)
    Sn = S[s-1] ; Vn = V[v-1] ; I1n = I1[i1-1] ; I2n = I2[i2-1]
    Sn = Sn + n - (beta1*I1n + beta2*I2n)*Sn - w*Sn - m*Sn
    Vn = Vn + w*Sn - beta1*(1-e)*Vn*I1n - beta2*Vn*I2n - m*Vn
    I1n = I1n + beta1*(Sn + (1-e)*Vn)*I1n - (m + delta + c)*I1n
    I2n = I2n + c*I1n + beta2*(Sn + Vn)*I2n - (m + delta + gamma + 0.07)*I2n
    S.append(Sn)
    V.append(Vn)
    I1.append(I1n)
    I2.append(I2n)
t = np.arange(0,540,1)
```

```
plt.subplot(211)
plt.plot(t,I1,'r',label='I')
plt.xlabel('temps en jours')
plt.ylabel('nbr_de_personne/nbr_total')
plt.title('réponse à la perturbation pour e = 0.9')
plt.subplot(212)
plt.plot(t,I2,color='purple',label='I')
plt.xlabel('temps en jours')
plt.ylabel('nbr_de_personne/nbr_total')
plt.title('réponse à la perturbation pour e = 0.9')
plt.tight_layout()
fig = plt.gcf()
fig.set_size_inches(10,7)
```