Modélisation d'un comportement addictif

Projet S6

Célia ROESS, Adam SEKKAT, Matthew SANCHEZ

Avril 2024

Sommaire

Sommaire

Introduction

Définition d'une addiction

Une **addiction** est une pathologie cérébrale définie par une dépendance à une substance avec des conséquences délétères.

Objectif

Le projet consiste en la modélisation d'un comportement addictif au cours du temps avec $t \geq 0$ avec $\Delta t = 1$ *semaine*.

Travail à réaliser

- Code Python pour afficher graphiquement les variables
- Création de Sliders pour changer les paramètres du modèle
- Création d'une interface Tkinter

Sommaire

Variables utilisées

Intensité de désir

$C(t)$: Intensité de désir

- si $C(t) < 0$, il n'y a pas de désir
- si $C(t) > 0$, il y a un peu de désir
- si $C(t) \gg 0$, il y a beaucoup de désir

Intensité de self-contrôle

S(t) : Intensité de self-contrôle

Intensité de l'influence extérieure

E(t) : Intensité de l'influence extérieure (sociale, économique, légale...)

Lois de comportement

Loi de comportement

$\Psi(t) = C(t) - S(t) - E(t)$

Si $\Psi(t) < 0$, alors le risque de devenir addict est nul.

Vulnérabilité

$V(t) = max\{0, min\{\Psi(t), 1\}\}$ donc $0 \leq V(t) \leq 1$

On est dans un état addictif lorsque : $V \sim 1$ (par exemple : $V \geq 0.85$).

Passage à l'acte

Le passage à l'acte $A(t)$ correspond à la fréquence de consommation.

Les paramètres du modèle

Paramètre	Valeur	Représentation
C_0	$= 0$	intensité de fringale ou de désir initiale
S_0	$= S_m$	intensité de Self-Control initiale
E_0	$= 1$	représentation de l'influence sociétale initiale
λ_0	$= 0.2$	paramètre initial de la loi de Poisson

Paramètre	Ordre de grandeur	Représentation
d	~ 0.2	effet
q	~ 0.8	quantité maximale physiologique (ce que le corps de l'individu est capable d'ingérer)
p	~ 0.2 à 0.8	résilience psychologique
h	$\sim pS_m$	coefficient de la compétition directe entre S et C
k	$\sim \frac{p}{q} S_m$	coefficient de la résultante de S de la consommation
S_m	~ 0.5	S maximal : maximum de self-contrôle
b	$\sim \frac{2d}{q}$	influence du passage à l'acte : constante de proportionnalité de l'impact A, soit la sensibilité à l'indice
λ	~ 0.1 à 0.5	paramètre de la loi de Poisson
R_m	~ 7	maximum défini de la loi de Poisson
m_E	$= 0.01$	pas de l'influence sociétale : modification apportée à l'influence sociétale pour chaque pas de temps ($-m_E < 0$: individu rentrant progressivement dans un milieu défavorable, poussant à l'addiction)
m_λ	$= 0.001$	pas du paramètre de la loi de Poisson : modification apportée au paramètre de la loi de Poisson pour chaque pas de temps

Modèle mathématique dynamique

- $C(t+1) = (1-d) \times C(t) + b \times min\{1, 1-C(t)\} \times A(t)$
  avec $C(0) = C_0$ et $0 \leq d \leq 1$
- $S(t+1) = S(t) + p \times max\{0, S_{max} - S\} - h \times C(t) - k \times A(t)$
  avec $S(0) = S_0$ et $0 \leq p$
- $E(t+1) = E(t) + m_e$
- $A(t) = q \times V(t) + \frac{R(\lambda)}{R_{max}} \times q \times (1 - V(t))$
  avec $R(\lambda) \sim \mathcal{P}(\lambda)$: Poisson
- $\Psi(t) = C(t) - S(t) - E(t)$
- $V(t) = max\{0,\ min\{\Psi, 1\}\}$

Modèle appliqué à un groupe de 2 personnes

Hypothèse

On fait l'hypothèse que tous les paramètres (d, b, h, k, p, S_{max}) sont les mêmes.

Chaque patient i observe la patient j à travers ses passages à l'acte avec $(i, j) = (1, 2)$ *ou* $(2, 1)$.

Interactions

On a :

$$\begin{cases} A_i(t) > 0 \Rightarrow C_j(t) \nearrow \\ A_i(t) \sim 0 \Rightarrow S_j(t) \nearrow \end{cases}$$

Modèle appliqué à un groupe de 2 personnes

Modèle

$$C_i(t+1) = (1-d)C_i(t) + \alpha A_j(t)C_i(t) + bmin\{1, 1 - C_i(t)\}A_i(t)$$

$$S_i(t+1) = S_i(t) + P_j max\{0, S_{max} - S_i(t)\} - hC_i(t) - kA_i(t)$$

avec $P_j = p + \beta e^{-\gamma A_j(t)}$

Etude

On observe l'action du patient j sur le patient i.

On a alors :
$\{C, S, E, V, A\} \rightarrow \{C_1, S_1, E, V_1, A_1\}$ et $\{C_2, S_2, E, V_2, A_2\}$
avec $A_i(t) = qV_i(t) + \frac{R(\lambda)}{R_{max}}q(1 - V_i(t))$

Sommaire

```python
def Addiction_sans_exposition_sociale(d,q,p,S_max,R_max,lambda_0,C0,E0,m_e,m_lambda,N):

    h = p*S_max
    k = (p/q)*S_max
    b = 2*d/q

    t = np.arange(0,N+1)
    C = np.zeros(N+1)
    S = np.zeros(N+1)
    E = np.zeros(N+1)
    psi = np.zeros(N+1)
    V = np.zeros(N+1)
    A = np.zeros(N+1)

    C[0] = C0
    S[0] = S0
    E[0] = E0

    random_poisson = np.random.poisson(lambda_0)

    for i in range(N+1):
        psi[i] = C[i] - S[i] - E[i]
        V[i] = min(1,max(psi[i],0))
        A[i] = q*V[i] + (random_poisson/R_max)*q*(1-V[i])
        if i < N :
            C[i+1] = (1-d)*C[i] + b*min(1,1-C[i])*A[i]
            S[i+1] = S[i] + p*max(0,S_max - S[i]) -h*C[i] -k*A[i]
            E[i+1] = E[i] - m_e
    return t,C,S,E,V,A
```

```python
def Addiction_avec_exposition_sociale(d,q,p,S_max,R_max,lambda_0,C0,E0,m_e,m_lambda,N):

    h = p*S_max
    k = (p/q)*S_max
    b = 2*d/q

    t = np.arange(0,N+1)
    C = np.zeros(N+1)
    S = np.zeros(N+1)
    E = np.zeros(N+1)
    psi = np.zeros(N+1)
    V = np.zeros(N+1)
    A = np.zeros(N+1)

    random_poisson = np.zeros(N+1)

    C[0] = C0
    S[0] = S0
    E[0] = E0

    for i in range(N+1):
        psi[i] = C[i] - S[i] - E[i]
        V[i] = min(1,max(psi[i],0))
        random_poisson[i] = np.random.poisson(lambda_0)
        A[i] = q*V[i] + (random_poisson[i]/R_max)*q*(1-V[i])
        if i < N :
            C[i+1] = (1-d)*C[i] + b*min(1,1-C[i])*A[i]
            S[i+1] = S[i] + p*max(0,S_max - S[i]) -h*C[i] -k*A[i]
            E[i+1] = E[i] - m_e

    return t,C,S,E,V,A
```

Figure: Code python pour le modèle à 1 patient

```python
for i in range(N+1):
    psi_patient1[i] = C_patient1[i] - S_patient1[i] - E[i]
    psi_patient2[i] = C_patient2[i] - S_patient2[i] - E[i]
    V_patient1[i] = min(1,max(psi_patient1[i],0))
    V_patient2[i] = min(1,max(psi_patient2[i],0))
    random_poisson[i] = np.random.poisson(lambda_0)
    P_patient1[i] = p + beta*np.exp(-gamma*A_patient1[i])
    P_patient2[i] = p + beta*np.exp(-gamma*A_patient2[i])
    A_patient1[i] = q*V_patient1[i] + (random_poisson[i]/R_max)*q*(1-V_patient1[i])
    A_patient2[i] = q*V_patient2[i] + (random_poisson[i]/R_max)*q*(1-V_patient2[i])
    if i < N :
        C_patient1[i+1] = (1-d)*C_patient1[i] + alpha*A_patient2[i]*C_patient1[i] + b*min(1,1-C_patient1[i])*A_patient1[i]
        C_patient2[i+1] = (1-d)*C_patient2[i] + alpha*A_patient1[i]*C_patient2[i] + b*min(1,1-C_patient2[i])*A_patient2[i]
        S_patient1[i+1] = S_patient1[i] + P_patient2[i]*max(0,S_max - S_patient1[i]) - h*C_patient1[i] - k*A_patient1[i]
        S_patient2[i+1] = S_patient2[i] + P_patient1[i]*max(0,S_max - S_patient2[i]) - h*C_patient2[i] - k*A_patient2[i]
        E[i+1] = E[i] - m_e
return t,C_patient1,C_patient2,S_patient1,S_patient2,E,V_patient1,V_patient2,A_patient1,A_patient2
```

Figure: Partie du code python modifiée pour le modèle à 2 patients

Code principal (toutes les courbes)

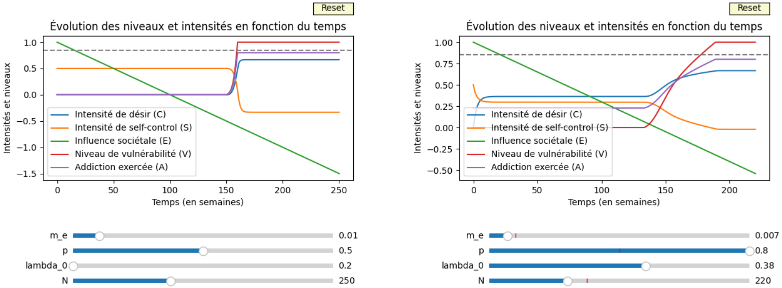


Figure: Variation des paramètres avec le code principal

Choix des courbes

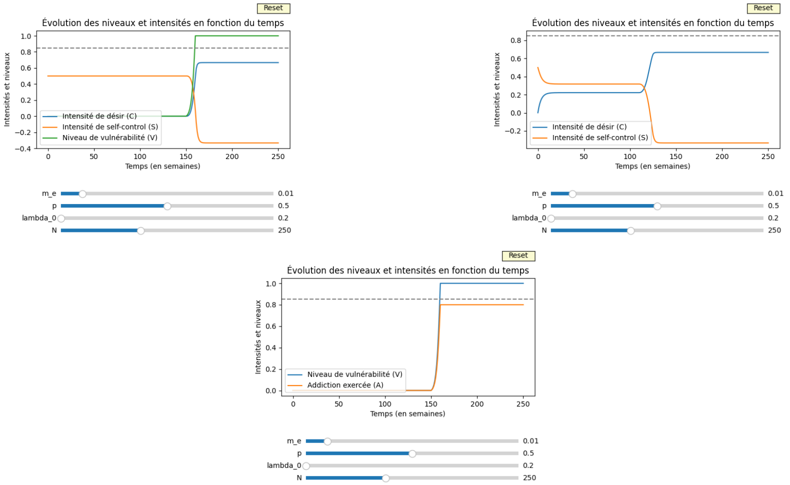


Figure: Test avec le choix des courbes

Variation des paramètres

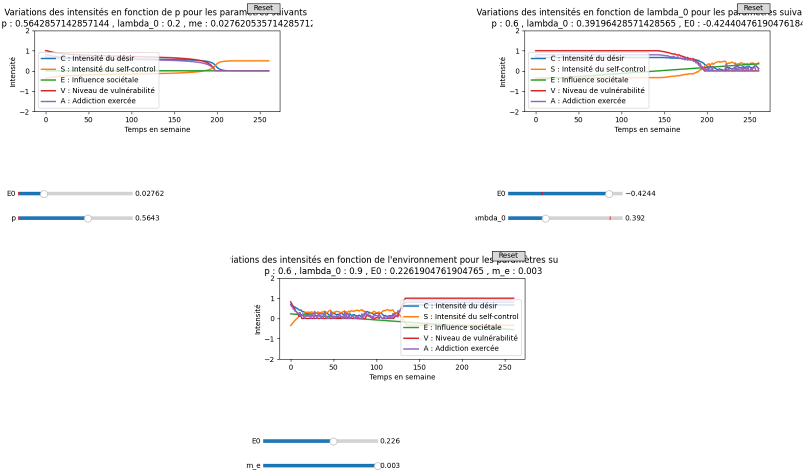


Figure: Test avec variation des paramètres

Résultats pour un groupe de 2 patients

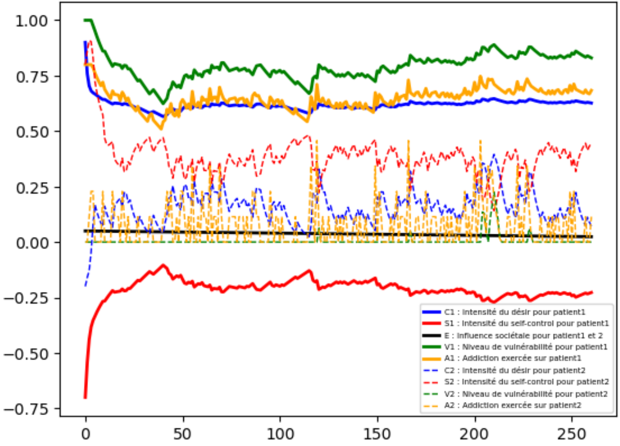
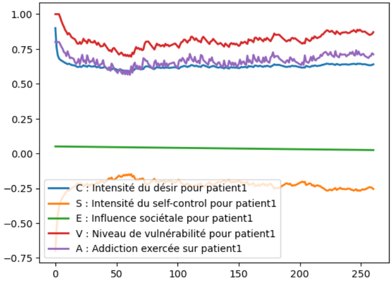
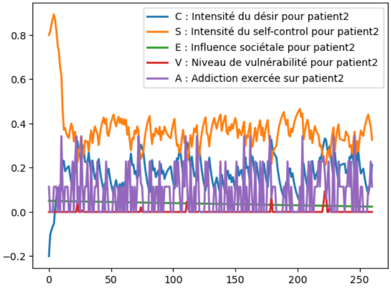


Figure: Résultats pour un groupe de 2 patients

Résultats pour un groupe de 2 patients



(a) Résultats du patient n°1



(b) Résultats du patient n°2

Figure: Résultats pour chacun des 2 patients

Interface Tkinter

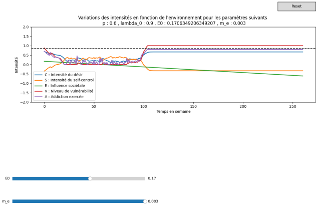
Afin de faciliter l'utilisation et la navigation entre les différents programmes, nous avons créé une interface avec Tkinter.

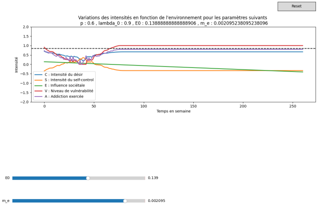
Sommaire

Modification de E_0 et m_e

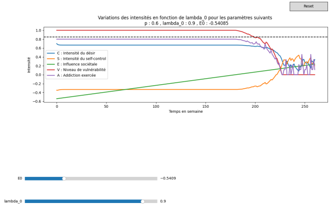
- $E_0 \sim 1$ et $m_e \sim 0.003$
- $E_0 \sim 1$ et $m_e \sim -0.003$
- $E_0 \sim -1$
- $E_0 \sim 0$ et $m_e \searrow -0.003$
- $E_0 \sim 0$ et $m_e \nearrow 0.003$

Un bon environnement est important pour le maintien de la sobriété.



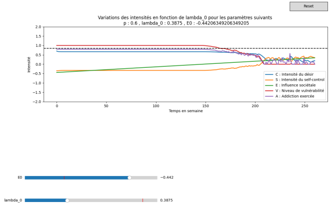


Modification de E_0 et λ_0

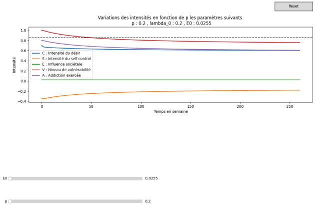


Ces résultats vont dépendre de m_e

- $m_e > 0$ et $\forall \lambda_0$
- $m_e < 0$

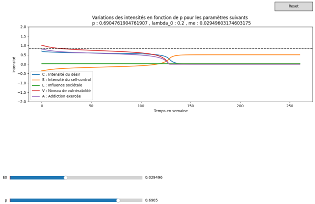


Modification de E_0 et p (résilience / thérapie)

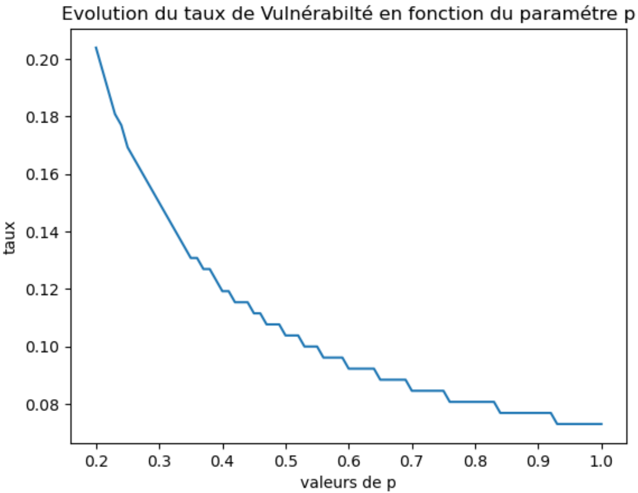


p considère la résilience psychologique et l'effet de la thérapie.

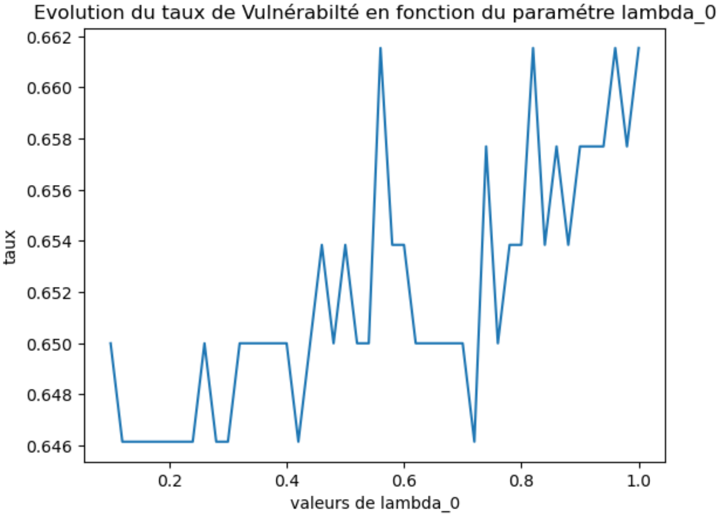
- $m_e > 0$, p $\varnothing$ influence
- $m_e < 0$, $\forall p, \forall E_0$
- $m_e = 0$, E constante
  - $E_0 < 0$: p $\varnothing$ influence
  - $E_0 > 0$



Variation du taux de vulnérabilité en fonction de p



Variation du taux de vulnérabilité en fonction de λ



Sommaire

Conclusion

Modèle

Les équations données permettent de définir un modèle fort pour l'étude d'un comportement addictif appliquée à 1 personne.

Il a fallu adapter ce modèle pour un groupe de 2 patients.

Observations effectuées

On peut déduire de nos différentes courbes que le facteur est d'avoir un bon environnement afin d'avoir un self-contrôle important. Les chances d'atteindre un état addictif sont moindres.

Différences entre 1 patient et 2 patients

En fonction des passages à l'acte d'un patient, soit le désir, soit le self-contrôle de l'autre patient est affecté.

Les limites du projet

Les améliorations possibles du code

- Ajout de Sliders sur les fenêtres graphiques pour tous les paramètres
- Diminution du bruit pour les graphiques de variation et pour le modèle à 2 patients

Les limites

- Difficulté à évaluer les intensités de désir et de self-contrôle
- Difficulté à l'intensité d'influence extérieure, d'autant plus qu'elle prend en compte beaucoup de paramètres
- Chaque personne réagit de façon différente face aux composants addictifs

Fin de la présentation

Merci !

On reste à votre écoute pour toute question.

Le fichier Jupyter Notebook est disponible dans le dossier ZIP.