

Du neurone au réseau de neurones : entre modélisation physique et informatique

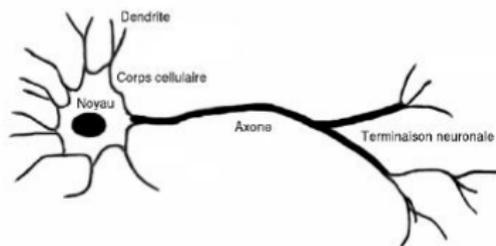
LANAUD Elsa

2016-2017

- 1 Introduction
- 2 Modèle de Hodgkin et Huxley
 - Création des équations du modèle
 - Simulation numérique
 - Limite du modèle de Hodgkin et Huxley
- 3 Modèle du soliton
 - Forme théorique
 - Simulation numérique
- 4 Conclusion

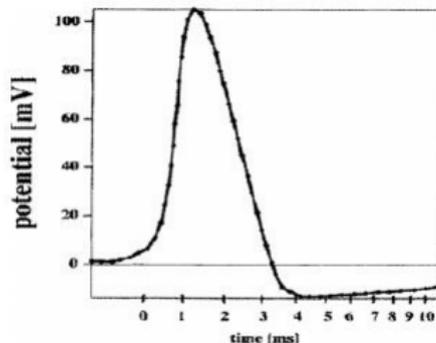
Introduction I

Objectif : comprendre le fonctionnement d'un neurone c'est-à-dire étudier la propagation de l'information dans celui-ci.



Source: <https://perso.univ-rennes1.fr/coralie.renault/L3.pdf>, Coralie Renault

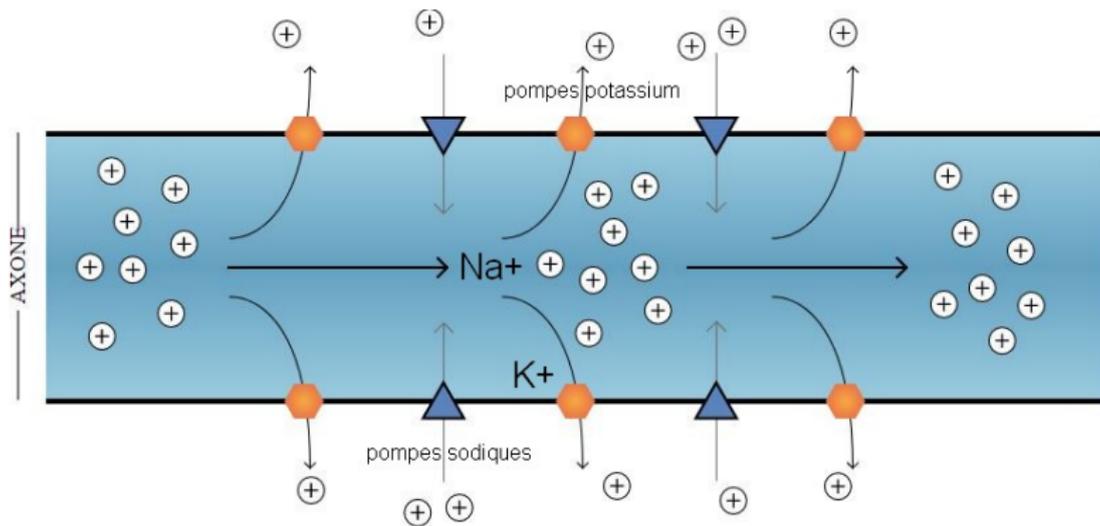
Figure – Composition d'un neurone



Source: 2012-AppalivanRienenHamburg-APLBL.pdf, Revathi Appali-Ursula von Rienen-Thomas Heimburg

Figure – Propagation d'un potentiel d'action

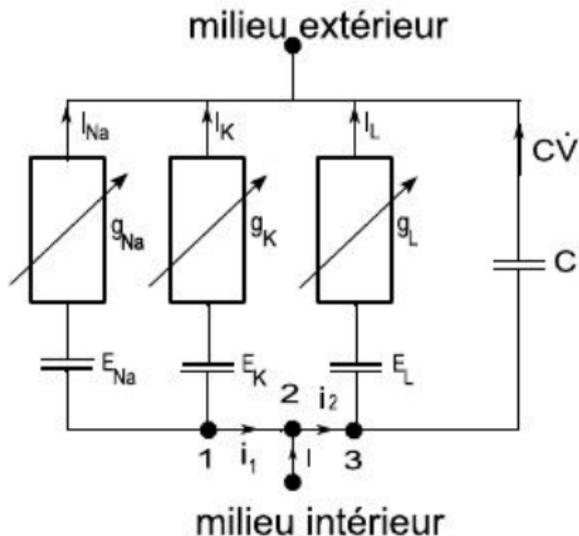
Modélisation physique I



Source: <http://www.techno-science.net>

Figure – Propagation du potentiel dans l'axone

Modélisation physique II



Source: <https://perso.univ-rennes1.fr/coralie.renault/L3.pdf>, Coralie Renault

Figure – Modélisation par un circuit RC parallèle

Equations du modèle I

Système de Hodgkin et Huxley

$$\left\{ \begin{array}{l} \frac{dn}{dt} = \alpha_n \cdot (1 - n) - \beta_n \cdot n \\ \frac{dm}{dt} = \alpha_m \cdot (1 - m) - \beta_m \cdot m \\ \frac{dh}{dt} = \alpha_h \cdot (1 - h) - \beta_h \cdot h \\ I = C \frac{dV}{dt} + m^3 h g_{Na} (V - E_{Na}) + n^4 g_K (V - E_K) \end{array} \right. \quad (1)$$

où

$$\left\{ \begin{array}{l} \alpha_{indice}, \beta_{indice} : \text{fonctions du potentiel} \\ m, n : \text{probabilités d'ouverture} \\ h : \text{probabilité d'activation} \\ I : \text{courant de l'axone} \\ C : \text{capacité de l'axone} \\ G_{indice} : \text{conductance des ions} \end{array} \right. \quad (2)$$

Equations du modèle II

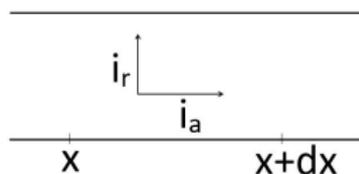


Figure – Décomposition en courants axial et radial

$$\begin{cases} i_a(x + dx) - i_a(x) = -i_r \\ V(x + dx) - V(x) = -\frac{2R}{a} \cdot i_a \end{cases} \quad (3)$$

$$\implies i_r = \frac{a}{2R} \cdot \frac{\partial^2 V}{\partial^2 x}$$

Equations du modèle III

Equation d'évolution

$$\frac{a}{2R} \cdot \frac{\partial^2 V}{\partial x^2} = C \cdot \frac{\partial V}{\partial t} + m^3 \cdot h \cdot g_{Na} \cdot (V - E_{Na}) + n^4 \cdot g_K \cdot (V - E_K)$$

où

$$\left\{ \begin{array}{l} a : \text{rayon de l'axone} \\ R : \text{résistance de l'axone} \\ C : \text{capacité de l'axone} \end{array} \right. \quad (4)$$

Implémentation Python I

$$V[k, t + 1] = V[k, t] + \text{coeff} * (V[k + 1, t] + V[k - 1, t] - 2 * V[k, t]) - \text{coeff1} * ((V[k, t] - ENa) - n * *4 * GK * (V[k, t] - EK))$$

Annexe 1

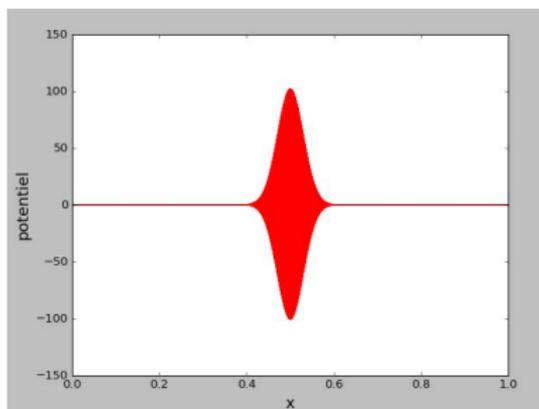


Figure – Courbes obtenues pour $N_x = 1000$, $\text{coeff} = 0.5$ et $\text{coeff1} = 0.01$ avec $V[N_x/2, 0] = 1$

Implémentation Python II

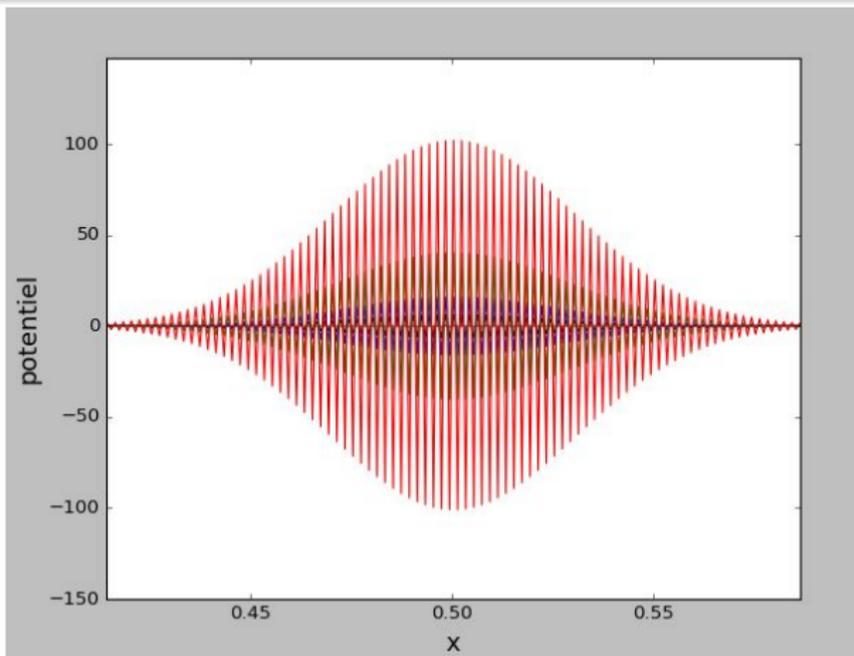


Figure – Courbes précédentes zoomées

Les limites du modèle I

Modèle de Hodgkin et Huxley :

- modèle fréquentiel donc électrique
 - aucune référence à des variables thermodynamiques
- ⇒ il s'agirait en fait d'une onde solitaire

Définition et propriétés I

Définition

Une onde solitaire se caractérise par :

- une unique oscillation qui se propage sans se déformer,
- dans un milieu non linéaire et dispersif,
- solution d'équations aux dérivées partielles non linéaires.

Propriété

Un soliton peut naître à la suite d'un gain d'énergie et est particulièrement stable vis-à-vis de perturbations.

Soliton I

Equation d'Euler avec dispersion

$$\frac{\partial^2}{\partial \tau^2} \Delta \rho = \frac{\partial}{\partial z} \left(c^2 \cdot \frac{\partial \Delta \rho}{\partial z} \right) - h \cdot \frac{\partial^4}{\partial z^4} \Delta \rho$$

où

- { $\Delta \rho$: variation de la densité de la membrane
 - { c : vitesse du son
 - { h : paramètre de mise en place de l'échelle linéaire de l'impulsion
- (5)

Soliton II

Equation différentielle obtenue

$$\beta^2 \cdot \frac{\partial^2 u}{\partial \xi^2} = \frac{\partial}{\partial \xi} \cdot \left(B(u) \cdot \frac{\partial u}{\partial \xi} \right) - \frac{\partial^4 u}{\partial \xi^4}$$

où

$$\begin{cases} \beta : \text{vitesse de propagation du potentiel d'action sans dimension} \\ \xi = x - ct \\ B(u) = 1 + \frac{\rho_0}{c^2} \cdot p \cdot u + \frac{\rho_0^2}{c^2} \cdot q \cdot u^2 \end{cases} \quad (6)$$

Soliton III

$$u(\xi) = \frac{2 \cdot a_+ \cdot a_-}{(a_+ + a_-) + (a_- - a_+) \cdot \cosh(\xi \cdot \sqrt{1 - \beta^2})}$$

où

$$\begin{cases} a_{\pm} = \frac{-B_1}{B_2} \cdot \left(1 \pm \sqrt{\frac{\beta^2 - \beta_0^2}{1 - \beta_0^2}} \right) \\ B_{indice}, \beta_0 : \text{coefficients adimensionnés} \end{cases} \quad (7)$$

Implémentation Python I

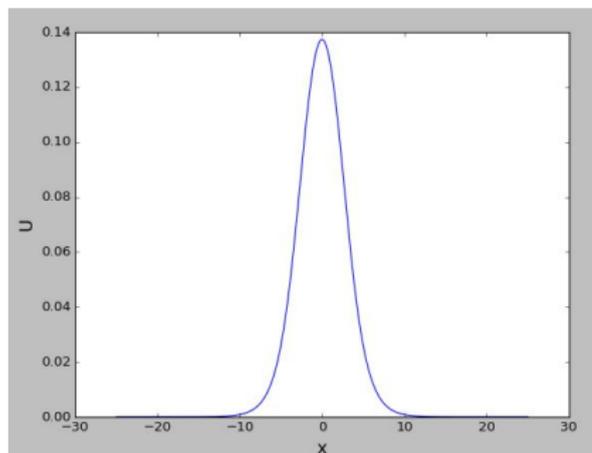


Figure – $\beta = 0.7$

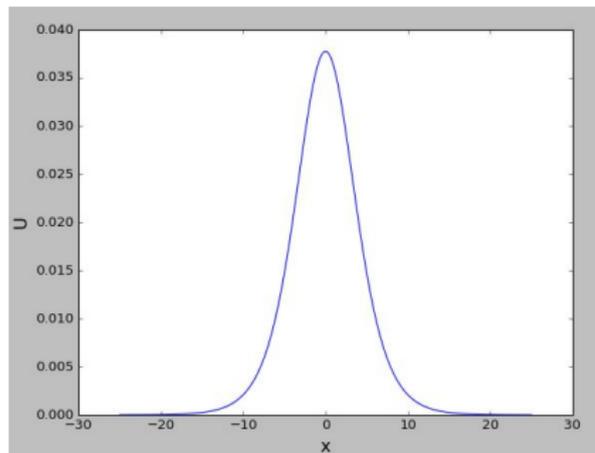


Figure – $\beta = 0.9$

Implémentation Python II

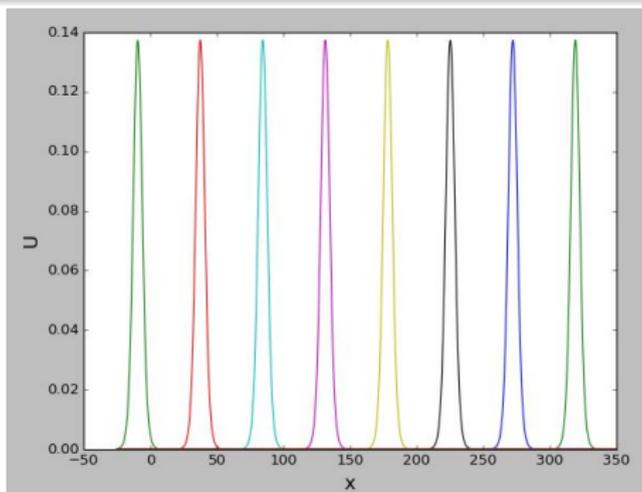


Figure – Propagation sans déformation du soliton ; courbes obtenues avec $\beta = 0.7$

Conclusion

- Mettre en évidence la propagation du message nerveux dans l'axone grâce au modèle de Hodgkin et Huxley.
- Comprendre comment le potentiel se propage sans atténuation dans l'axone grâce au modèle du soliton.

Annexe 1

```
def diffusion(Nx, Nt) :
    dt = 1
    dx = 1 * 10 ** - 7
    GNa = 0.12
    GK = 0.036
    ENa = 0.12
    EK = -0.012
    n0 = 0.32
    m0 = 0.06
    h0 = 0.56
    T = np.arange(0, Nt, 1)
    V = np.zeros((Nx, len(T)))
    for t in range(len(T)) :
        V[Nx-1, t] = 0
        V[0, t] = 0
    for k in range(len(Nx)) :
        V[k, 0] = 0
    V[Nx//2, 0] = 1
    coeff1 = 0.01
    coeff = 0.5
    for t in range(len(T)-1) :
        for k in range(1, Nx-1) :
            alphan = 0.01*(10 - V[k, t])/(np.exp(1 - V[k, t]/10) - 1)
            betan = 0.125*np.exp(- V[k, t]/80)
            alpham = 0.1*(25 - V[k, t])/(np.exp((25 - V[k, t])/10) - 1)
            betam = 4*np.exp(- V[k, t]/18)
```

```

    alphah = 0.7*(np.exp(- V[k,t]/20)+1)
    betah = 1/np.exp((30 - V[k,t])/10)
    ninfini = alphan/(alphan + betan)
    taun = 1/(alphan + betan)
    minfini = alpham/(alpham + betam)
    taum = 1/(alpham + betam)
    hinfini = alphah/(alphah + betah)
    tauh = 1/(alphah + betah)
    m = m0 - ((m0 - minfini)*(1 - np.exp(- t/taum)))
    h = h0 - ((h0 - hinfini)*(1 - np.exp(- t/tauh)))
    n = n0 - ((n0 - ninfini)*(1 - np.exp(- t/taun)))
    V[k,t+1] = V[k,t] + coeff*( V[k+1,t] + V[k-1,t] - 2*V[k,t] ) - coeff1*((V[k,t] -
ENa) - n**4*GK*(V[k,t] - EK))
    return V

```

```

Nx = 1000
Nt = 1000
V = diffusion(Nx,Nt)
X = np.linspace(0,1,Nx)
plt.plot(X,V[ :, : :Nt/10])
plt.xlabel('x')
plt.ylabel('potentiel')
plt.legend(loc='best')
plt.show()

```

Annexe 2

```
def soliton(beta) :
    c = 3 * 10 ** 8
    B1 = -16.6
    B2 = 79.5
    beta0 = 0.649851
    aplus = - (B1/B2)*(1 + np.sqrt((beta**2 - beta0**2)/(1 - beta0**2)))
    amoins = - (B1/B2)*(1 - np.sqrt((beta**2 - beta0**2)/(1 - beta0**2)))
    X = np.linspace(-25, 350, 1000)
    T = np.linspace(0, 1 * 10 ** - 6, 1000)
    U = np.zeros((len(T),len(X)))
    for t in range(len(T)) :
        for x in range(len(X)) :
            U[t,x] = 2*aplus*amoins/((aplus + amoins) + (aplus - amoins)*np.cosh((X[x] -
c*T[t])*np.sqrt(1 - beta**2)))
    plt.plot(X,U[:, : :len(T)//10])
    plt.xlabel('x',fontSize=16)
    plt.ylabel('U',fontSize=16)
    plt.show()
```