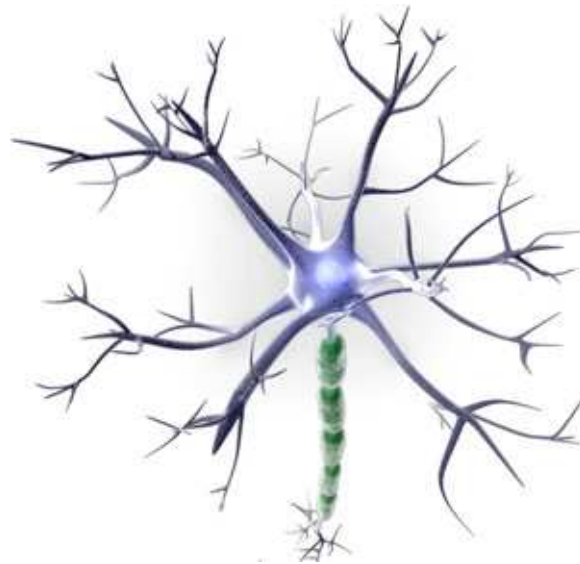


Etude mathématique et simulation numérique de la dynamique neuronale



Préparé par :

KÉVIN POLISANO
kevin.polisano@ensimag.imag.fr

Encadré par :

ARNAUD TONNELIER
arnaud.tonnelier@inria.fr

6 mars 2012

Résumé

La modélisation est devenue un outil essentiel en neurosciences notamment pour explorer la nature du « code neuronal », question qui reste très largement ouverte encore aujourd'hui. Différents projets internationaux ont pour objectifs la modélisation et la simulation de réseaux de neurones de grande taille et cherchent à reproduire, à analyser et à comprendre le fonctionnement de structures neuronales. Il convient alors d'utiliser des modèles de neurones réalistes (biologiquement plausibles) et dont la simulation soit peu coûteuse. Les modèles neuronaux de type intègre-et-tire (integrate-and-fire neural models) constituent une classe de modèles très largement utilisés en neuroscience computationnelle. Ces modèles qui s'écrivent comme des systèmes différentiels non-réguliers possèdent une dynamique très riche (points fixes, excitabilité, oscillations, chaos). Un élément essentiel des modèles de type intègre-et-tire est l'équation contrôlant le seuil d'un potentiel d'action. Dans ce travail nous proposons une généralisation de cette équation basée sur la réduction dimensionnelle de modèles détaillés.

Table des matières

1	Introduction	4
1.1	Environnement de travail	4
1.2	Déroulement du projet	4
1.3	Présentation du plan	4
2	Le modèle de Hodgkin-Huxley	5
2.1	Un peu d'histoire	5
2.2	Contexte biologique	5
2.3	Modélisation physique	6
3	Le modèle de Morris-Lecar	9
3.1	Réduction et diagramme de phase	9
3.2	Etude de stabilité du point fixe	13
3.3	Interprétation des phénomènes	13
3.4	Bifurcation de Hopf	17
4	Le modèle intègre-et-tire	17
4.1	Formulation mathématique	17
4.2	Introduction d'un nouveau type de seuil	19
4.3	Approximation du modèle de Hodgkin-Huxley	21
4.4	Discussion : application à des données biologiques	25
5	Conclusion	27
5.1	Perspectives futures	27
5.2	Bilan	27
5.3	Remerciements	27
6	Bibliographie	28
7	Annexes	29
7.1	Hodgkin.m	29
7.2	mainHH.m	30
7.3	Morris.m	31
7.4	MorrisExe.m	33
7.5	fi.m	34
7.6	gi.m	34
7.7	IF.m	34
7.8	mainIFizh.m	35
7.9	HHreduit.m	35
7.10	mainHHred.m	36
7.11	mapping.m	37
7.12	minimisationf.m	39
7.13	minimisationg.m	40
7.14	spike_train.m	41
7.15	init_conductance.m	42
7.16	input.m	42
7.17	mainIF.m	42
7.18	mainHHseuil.m	45

Table des figures

1	Echanges ioniques à travers les canaux la bicouche lipidique	5
2	Flux des principaux ions transitants par la membrane	6
3	Modélisation de la membrane d'un neurone par un circuit électrique	6
4	De haut en bas et de gauche à droite : potentiel au repos V de la membrane, et des variables n , m et h	8
5	Simulation du modèle de Hodgkin-Huxley : réponse de chaque variable du système pour un courant constant de $4\mu A/cm^2$ appliqué pour $t \geq t_{simul} = 10 ms$	8
6	Simulation du modèle de Hodgkin-Huxley : réponse de chaque variable du système pour un courant constant de $10\mu A/cm^2$ appliqué pour $t \geq t_{simul} = 10 ms$	9
7	Simulation de Morris-Lecar pour $I_{app} = 60 mA$, réponses temporelles V_2 et V_3 à partir de 2 conditions initiales distinctes : $V_2(0) = -20 mV$ et $V_3(0) = -10 mV$	10
8	Simulation de Morris-Lecar pour $I_{app} = 100 mA$, réponses temporelles V_2 et V_3 à partir de 2 conditions initiales distinctes : $V_2(0) = -20 mV$ et $V_3(0) = -10 mV$	11
9	Simulation de Morris-Lecar pour $I_{app} = 60 mA$ et représentation dans le plan de phase (V, n) des réponses V_2 et V_3 ainsi que des nullclines $V' = 0$ et $n' = 0$	12
10	Simulation de Morris-Lecar pour $I_{app} = 100 mA$ et représentation dans le plan de phase (V, n) des réponses V_2 et V_3 ainsi que des nullclines $V' = 0$ et $n' = 0$	12
11	Points d'équilibres pour $\phi \simeq 0$	14
12	Simulation de Morris-Lecar pour $I_{app} = 60 mA$ et $\phi = 0.004$, trajectoires dans le plan de phase	15
13	Simulation de Morris-Lecar pour $I_{app} = 90 mA$, la trajectoire en bleue s'enroule autour du point fixe tandis que la trajectoire en rouge s'approche du cycle limite	16
14	Interprétation des paramètres d'un modèle intègre-et-tire	18
15	Pattern Regular Spiking obtenu par le modèle d'Izhikevich avec $a = 0.02$, $b = 0.2$, $v_{reset} = -65 mV$ et $d = 8$	18
16	Pattern Chattering obtenu par le modèle d'Izhikevich avec $a = 0.02$, $b = 0.2$, $v_{reset} = -50 mV$ et $d = 2$	19
17	Coloration du plan de phase : pour chaque condition initiale $(V(0), n(0))$ on affecte au point correspondant une couleur suivant la valeur du potentiel maximal atteint par la trajectoire	20
18	On considère pour droite seuil la tangente à la V-nullcline au point d'inflexion	21
19	En haut la représentation de $g_{exc}(t)$ et en bas de $g_{inh}(t)$	22
20	Syncronisation des modèles HH avec et sans seuil	22
21	Minimisation de la V-nullcline par la méthode des moindres carrés	23
22	Approximation des surfaces f et g par des surfaces quadratiques	24
23	Problème des nullclines résultantes	25
24	Réponse d'un neurone biologique soumis à un courant donné pendant 40s	26
25	Principe de simulation et de prédiction de spikes	26

1 Introduction

Avant de rentrer dans le vif du sujet j'explique en quelques lignes comment et dans quel contexte s'est articulé mon travail de recherche.

1.1 Environnement de travail

Les travaux présentés dans ce document s'inscrivent dans le cadre du module Initiation à la Recherche en Laboratoire, proposé aux étudiants de deuxième année de l'Ensimag. C'est une occasion pour nous de découvrir le monde de la recherche en travaillant une demi-journée par semaine au sein d'une équipe de recherche en laboratoire. Chaque mercredi je me rendais à l'INRIA de Montbonnot pour travailler sur la dynamique neuronale au sein de l'équipe BIPOP, qui s'intéresse à l'étude de systèmes dynamiques non réguliers et à l'optimisation non différentiable. Je fus encadré par Arnaud Tonnelier, qui travaille plus spécifiquement sur la modélisation mathématiques en biologie et toxicologie.

1.2 Déroulement du projet

Dans un premier temps j'ai dû me familiariser avec le contexte biologique pour connaître l'état de l'art. Autrement dit un travail bibliographique préliminaire s'imposait. Cela consistait principalement à comprendre comment les premiers modèles décrivant le neurone ont été établis.

Dans un deuxième temps j'ai mis les mains à la pâte en étudiant mathématiquement ces modèles pour exhiber les caractéristiques de ces systèmes dynamiques riches (étude de stabilité, bifurcations, etc) tout en les simulant sous Matlab pour illustrer ces phénomènes.

Enfin, une fois les concepts de bases assimilés, j'étais en mesure de comprendre les problèmes qui se posent aux chercheurs. En effet les modèles initiaux décrivent très bien le comportement du neurone, mais ils présentent le gros défaut d'être lourd à simuler en raison de la complexité des équations qui interviennent, ce qui rend impossible la simulation de tout un réseau de neurones. C'est pourquoi une autre classe de modèles appelés « intègre-et-tire » sont utilisés. Ils font intervenir un seuil (que nous définirons plus loin dans le document) et des équations (de préférence polynomiales), qui se prêtent mieux à la simulation informatique. La dernière partie de mon travail consistait à rechercher un nouveau type de seuil, qui soit à la fois plus réaliste et qui ouvre de nouvelles perspectives sur le codage neuronal et le pouvoir prédictif des modèles de type intègre-et-tire.

1.3 Présentation du plan

Nous présenterons donc en trois parties la démarche suivie pour en arriver au problème posé et à sa résolution partielle. Dans la section 2 nous présentons le modèle d'Hodgkin-Huxley, puis dans la section 3 nous en donnons une réduction dimensionnelle. Cette réduction nous permet d'aboutir ensuite à un modèle de type intègre-et-tire avec une équation sur le seuil de potentiel d'action (appelé aussi *spike*).

2 Le modèle de Hodgkin-Huxley

2.1 Un peu d'histoire

Au début des années 1950 deux scientifiques, Alan Hodgkin et Andrew Huxley ont effectué des expériences sur un axone géant de calamar. Cet axone est remarquable de par son diamètre extraordinairement large (de l'ordre du millimètre), ce qui leur permis de réaliser des manipulations techniquement infaisables sur des neurones traditionnels trop petits. La contraction d'une tentacule est due à un courant électrique généré par les neurones du calamar, lorsqu'il repère un prédateur par exemple. Hodgkin et Huxley ont interprété ces potentiels d'actions en terme de migration d'ions sodium Na^+ et potassium K^+ à travers la membrane du neurone. Basé sur des mesures d'électrodes ils ont proposé un modèle mathématique connu sous le nom de modèle de Hodgkin-Huxley (HH). La popularité de leur modèle provient du formalisme biologique qu'il expose et des résultats réalistes qu'il fournit, puisqu'il est capable de reproduire toutes les propriétés d'un potentiel d'action. Hodgkin et Huxley ont été récompensés pour ces travaux par le prix Nobel de médecine et de physique en 1963.

2.2 Contexte biologique

Toutes les cellules vivantes possèdent une tension électrique (une différence de potentiel entre l'intérieur V_{in} et l'extérieur V_{out}) appelée potentiel membranaire, défini par :

$$V_M = V_{in} - V_{out}$$

Le potentiel de repos est le potentiel de la membrane quand la cellule est au repos. Typiquement le potentiel de repos d'un neurone se situe aux environs de -70 mV . Lorsqu'un flux d'ions positifs tels que Na^+ entre dans la cellule cela augmente le potentiel membranaire, on dit que la cellule est dépolarisée. Au contraire si un flux positif d'ions K^+ quitte la cellule ou un flux d'ions négatifs Cl^- entre alors la cellule devient hyperpolarisée. Ces trois types d'ions sont ceux que l'on trouve principalement de part et d'autre de la membrane. La membrane (et plus précisément la bicouche lipidique) est un mauvais conducteur, c'est en fait des canaux de protéines (dites transmembranaires) qui permettent aux ions de la traverser.

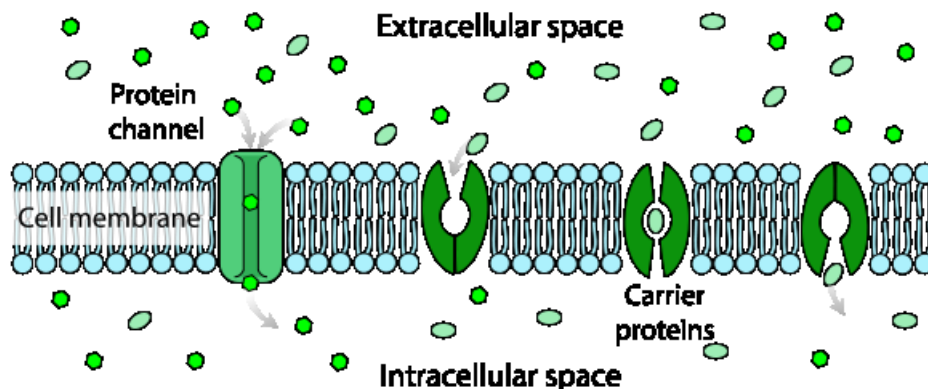


FIGURE 1 – Echanges ioniques à travers les canaux la bicouche lipidique

Il existe 2 types de canaux, ceux qui sont toujours ouverts et d'autres qui s'ouvrent et se ferment avec une certaine probabilité dépendant du potentiel membranaire. La plupart de ces derniers sont fermés quand la cellule est au repos. Supposons que la membrane soit perméable aux ions

potassium K^+ et aux ions Cl^- , le gradient de concentration de K^+ induit un flux sortant de la cellule provoquant un excès de charges positives en dehors et négatives à l'intérieur (par la présence d'anions A^-). Inversement pour les ions Cl^- , un équilibre s'installe au repos. Nous verrons plus loin que ce sont les ions Na^+ qui sont à l'origine d'un potentiel d'action (une décharge du neurone).

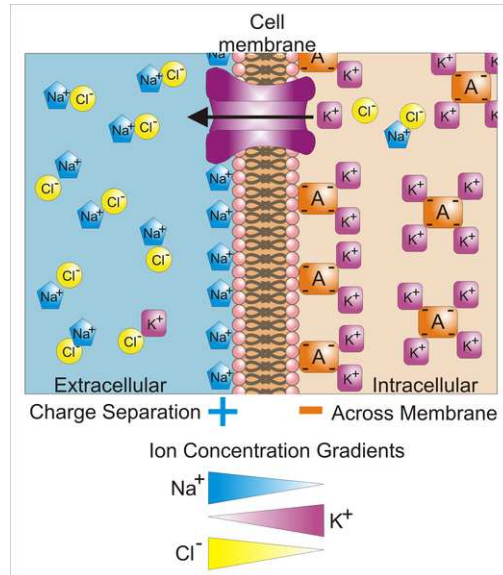


FIGURE 2 – Flux des principaux ions transistants par la membrane

2.3 Modélisation physique

On peut traduire les précédentes propriétés électriques par un circuit électrique équivalent :

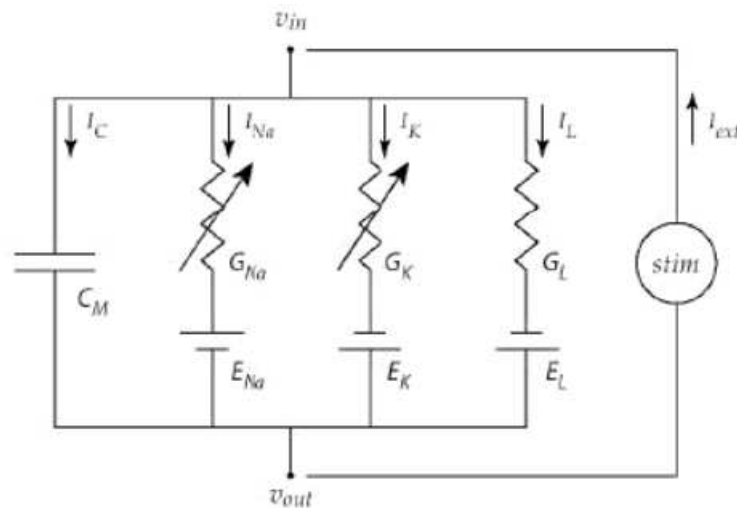


FIGURE 3 – Modélisation de la membrane d'un neurone par un circuit électrique

où le condensateur représente la capacité (*i.e* la charge électrique stockée) de la membrane, les résistances caractérisent la nature des canaux ioniques et les forces électromotrices pour représenter les potentiels électrochimiques.

Voici la façon dont on aboutit aux équations de Hodgkin-Huxley : la somme des courants étant nulle on a : $I_C + I_{Na} + I_K + I_L = 0$, avec $I_c = dq/dt$ (nombre de charges traversant la membrane par unité de temps) et $q = C_m V$ d'où $I_C = C_m \frac{dV}{dt}$. De plus pour chaque ion i le courant ionique vaut (d'après la loi d'ohm) $I_i = g_i(V - E_i)$ où g_i est la conductance et E_i le potentiel de Nerst.

$$C_M \frac{dV}{dt} = -g_{Na}(V - E_{Na}) - g_K(V - E_K) - g_L(V - E_L)$$

Les conductances ne sont pas constantes mais dépendent du potentiel membranaire V . Les canaux ioniques contiennent une ou plusieurs « portes », qui régulent le flux d'ions. On peut donc pour chacune d'elle associer une probabilité pour qu'elle soit ouverte ou fermée : n , m et h en fonction du type de portes. Pour qu'un flux d'ions passe à travers le canal il faut que toutes les portes soient ouvertes. Par exemple le canal potassium en contient 4 identiques de type n donc si on note \bar{g}_k la conductance maximale (constante cette fois) alors $g_K = \bar{g}_k n^4$. De même on a $g_{Na} = \bar{g}_{Na} m^3 h$.

En fait, comme il existe beaucoup de canaux ioniques, les variables n , m et h s'interprètent comme la fraction de portes ouvertes dans cette population. A un instant t donné il y a $p_i(t)$ ($p_i = n, m, h$) portes ouvertes et $1 - p_i(t)$ fermées. A l'instant $t + dt$ il se peut qu'un nombre de portes précédemment ouvertes se ferment et réciproquement, ceci dépendant du potentiel membranaire. On note $\alpha_i(V)$ le taux de portes s'ouvrant et $\beta_i(V)$ le taux de portes se fermant. Ainsi les variables n, m, h sont régies par une équation de cinétique chimique :

$$\frac{dp_i}{dt} = \alpha_i(V)(1 - p_i(t)) - \beta_i(V)p_i(t)$$

que l'on peut également mettre sous la forme :

$$\frac{dp_i}{dt} = \frac{p_\infty(V) - p_i(t)}{\tau_i(V)}$$

avec

$$p_\infty(V) = \frac{\alpha_i(V)}{\alpha_i(V) + \beta_i(V)} \quad \text{et} \quad \frac{1}{\alpha_i(V) + \beta_i(V)}$$

Finalement on aboutit à un système de 4 équations différentielles : une équation pour le potentiel membranaire et trois pour les canaux. Dans le cas de l'axone du calamar on a :

$$\left\{ \begin{array}{l} C_M \frac{dV}{dt} = -\bar{g}_{Na} m^3 h (V - E_{Na}) - \bar{g}_K n^4 (V - E_K) - \bar{g}_L (V - E_L) \\ \frac{dn}{dt} = \phi [\alpha_n(V)(1 - n) - \beta_n(V)n] \\ \frac{dm}{dt} = \phi [\alpha_m(V)(1 - m) - \beta_m(V)m] \\ \frac{dh}{dt} = \phi [\alpha_h(V)(1 - h) - \beta_h(V)h] \end{array} \right.$$

avec ϕ le paramètre de température. En effet la température joue un rôle important dans les expériences biologiques, les canaux y sont sensibles. Ce paramètre dépend exponentiellement de la température :

$$\phi = Q_{10}^{(T - T_{base})/10}$$

Pour l'axone géant du calamar, $T_{base} = 6.3^\circ C$ et $Q_{10} = 3$. De même les fonctions α_i et β_i ont été déterminées empiriquement (voir code). J'ai réalisé la simulation sous Matlab en utilisant une méthode de Runge-Kutta (le code est disponible en annexe). Au repos les potentiels sont constants :

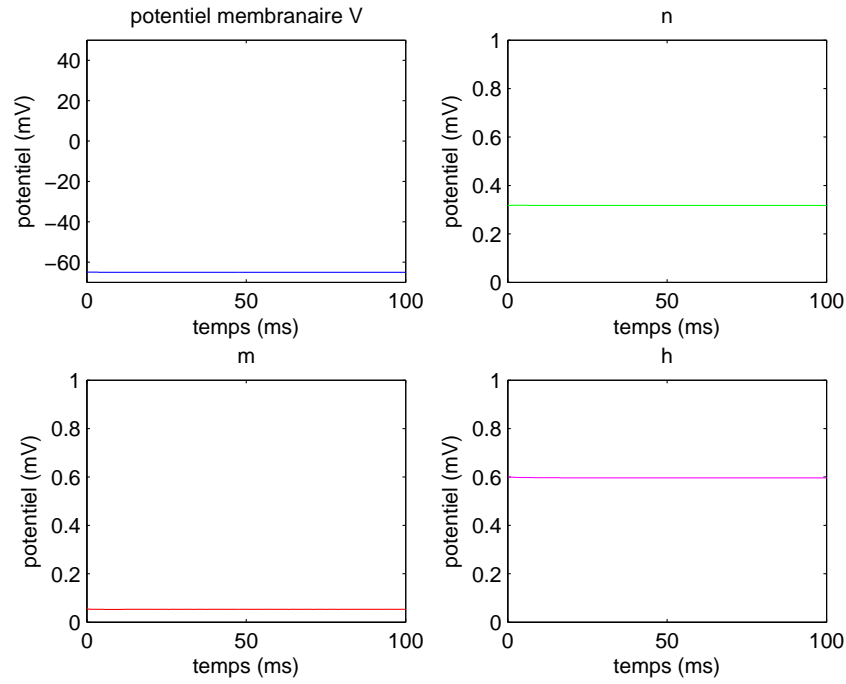


FIGURE 4 – De haut en bas et de gauche à droite : potentiel au repos V de la membrane, et des variables n , m et h

A partir de l'état de repos on a décidé d'injecter un courant constant $4\mu A$. On constate sur la figure 5 un potentiel d'action, puis la membrane revient rapidement à l'état de repos.

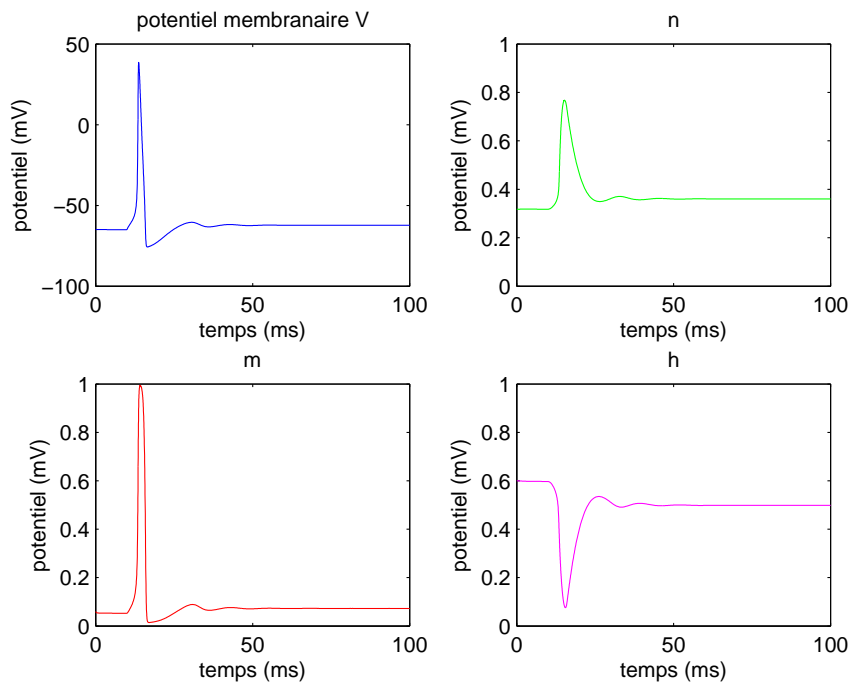


FIGURE 5 – Simulation du modèle de Hodgkin-Huxley : réponse de chaque variable du système pour un courant constant de $4\mu A/cm^2$ appliqué pour $t \geq t_{simul} = 10 ms$

En revanche si le courant est au dessus d'un certain seuil (environ $7\mu A/cm^2$) et appliqué suffisamment longtemps alors le modèle génère une réponse périodique, illustrée sur la figure 6. Nous expliquerons ces phénomènes dans la partie suivante.

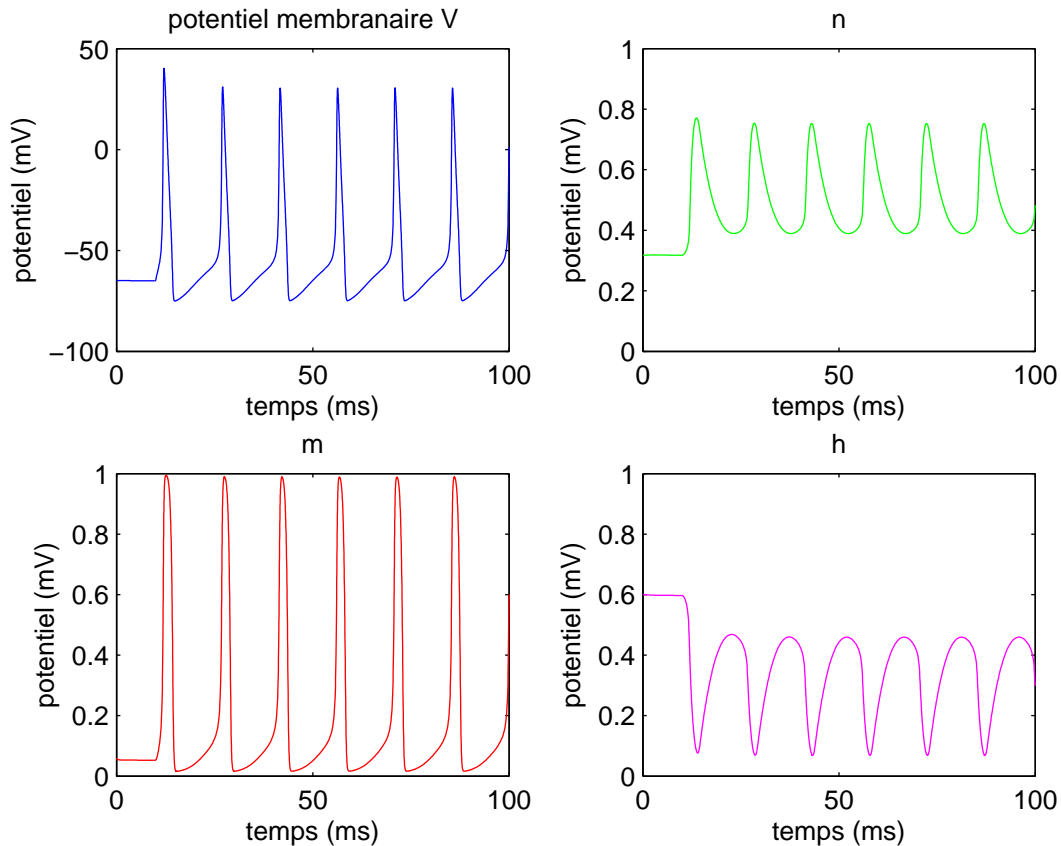


FIGURE 6 – Simulation du modèle de Hodgkin-Huxley : réponse de chaque variable du système pour un courant constant de $10\mu A/cm^2$ appliqué pour $t \geq t_{simul} = 10 ms$

3 Le modèle de Morris-Lecar

Histoire de changer un peu du calamar, je présente un autre modèle similaire à celui de HH, mais qui fait intervenir des ions Ca^+ chez le Balanus, un crustacé qui présente également un gros axone. La réduction que l'on présente ci-dessous est valable pour les deux modèles puisque le système est identique, au choix des constantes près (conductances, potentiels, etc dépendant du type de neurone).

3.1 Réduction et diagramme de phase

On peut constater sur les figures 5 et 6 que $n + h \simeq 0.8$. Par ailleurs si on examine

$$\frac{dm}{dt} = \frac{m_{\infty}(V) - m}{\tau_{\infty}(V)}$$

comme τ_{∞} est très petit, la dérivée augmente très rapidement et donc m atteint presque instantanément son régime stationnaire. Par conséquent, il est légitime d'effectuer l'approximation

$$m(t) \simeq m_{\infty}(V(t))$$

Ainsi on élimine deux variables et notre système différentiel initial se résume à un système de deux équations différentielles en V et n , appelé équations de Morris-Lecar :

$$\left\{ \begin{array}{l} C_M \frac{dV}{dt} = I_{app} - g_{Ca} m_{\infty} (V - E_{Ca}) - g_K n (V - E_K) - g_L (V - E_L) \equiv I_{app} - I_{ion}(V, n) \\ \frac{dn}{dt} = \frac{\phi(n_{\infty}(V) - n)}{\tau_n(V)} \end{array} \right.$$

On a vu précédemment qu'en fonction du courant appliqué I_{app} le comportement des réponses diffèrait. La théorie des systèmes dynamiques permet d'analyser celles-ci en fonction de ce paramètre. Pour cela on s'appuie sur le diagramme de phase qui se représente aisément puisque le système est de la forme :

$$\left\{ \begin{array}{l} \frac{dV}{dt} = f(V, n) \\ \frac{dn}{dt} = g(V, n) \end{array} \right.$$

J'ai simulé ce modèle à deux variables pour deux types de conditions initiales et deux courants. On s'aperçoit que ce modèle simplifié a le même comportement que celui de Hodgkin-Huxley, en effet pour un courant appliqué de 60 mA et une condition initiale $V_2(0) = -20 \text{ mV}$ légèrement supérieure au potentiel de repos $V_R \simeq -36 \text{ mV}$ la solution $V_2(t)$ retourne immédiatement à l'état de repos, tandis que pour $V_3(0) = -10 \text{ mV}$, le seuil d'excitabilité est dépassé et on observe un potentiel d'action avant le retour à l'état de repos.

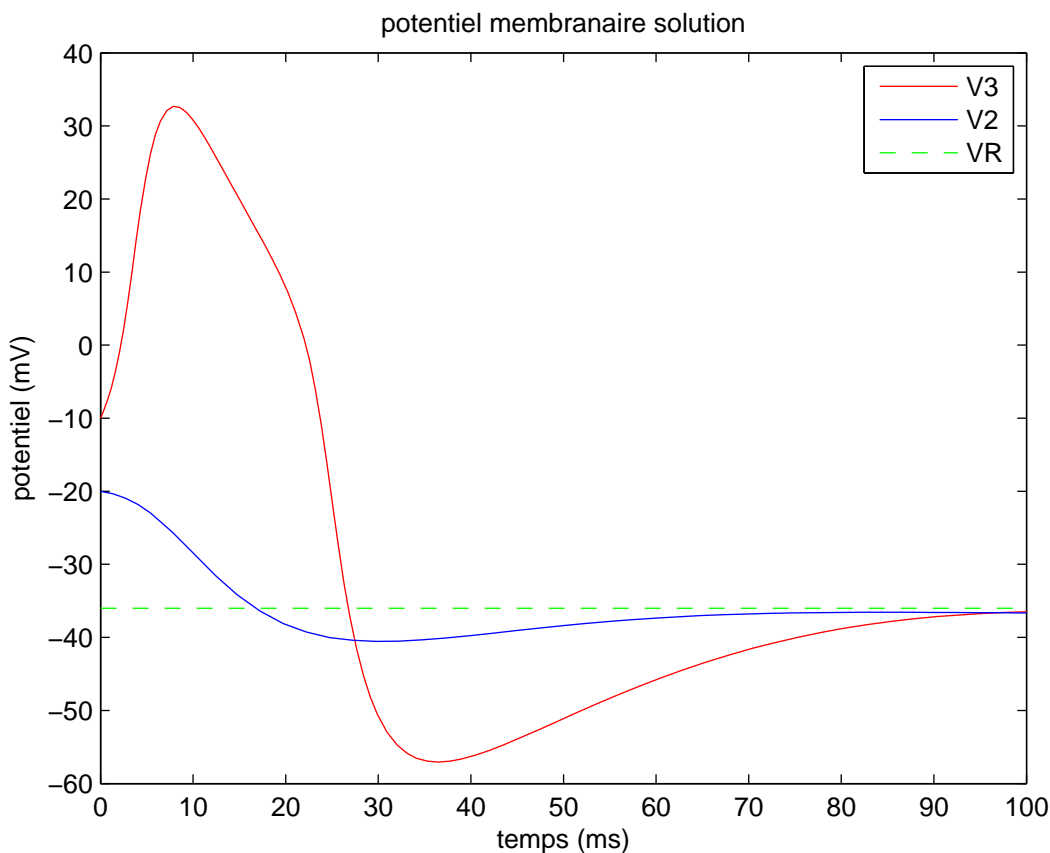


FIGURE 7 – Simulation de Morris-Lecar pour $I_{app} = 60 \text{ mA}$, réponses temporelles V_2 et V_3 à partir de 2 conditions initiales distinctes : $V_2(0) = -20 \text{ mV}$ et $V_3(0) = -10 \text{ mV}$

Si on augmente la valeur du courant appliqué à 100 mA alors comme pour le modèle de Hodgkin-Huxley on observe une réponse périodique pour les deux conditions initiales.

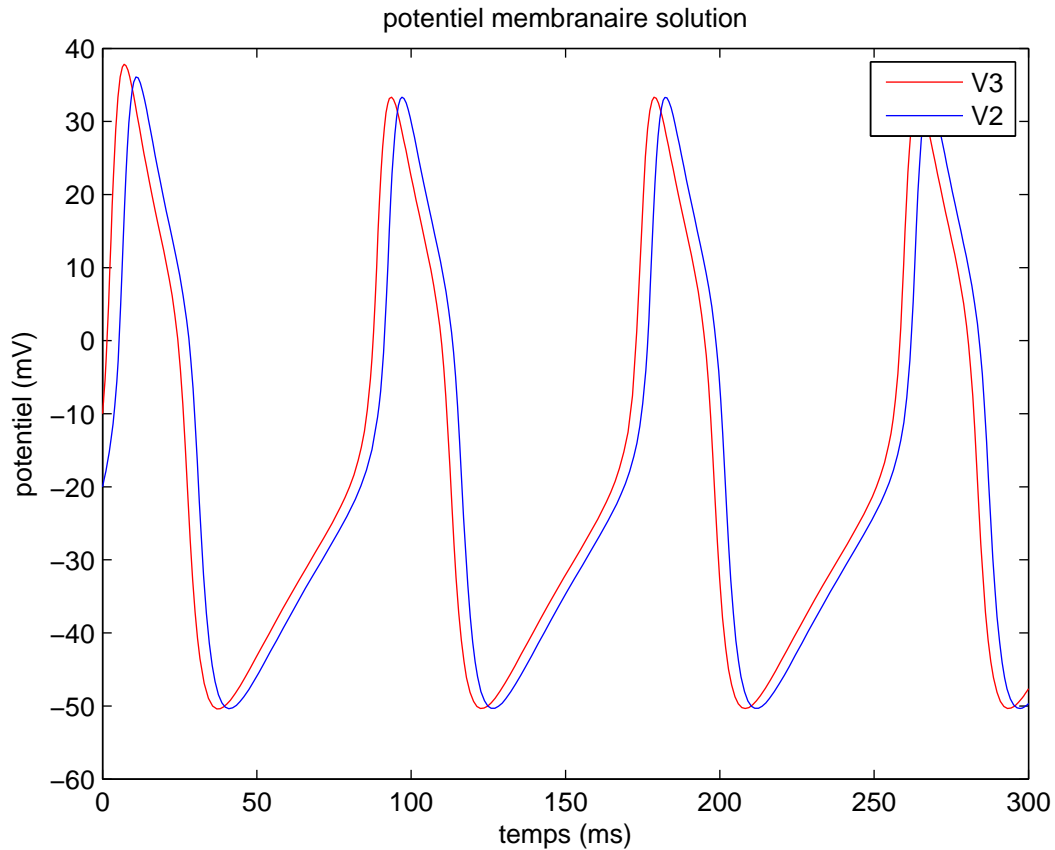


FIGURE 8 – Simulation de Morris-Lecar pour $I_{app} = 100 \text{ mA}$, réponses temporelles V_2 et V_3 à partir de 2 conditions initiales distinctes : $V_2(0) = -20 \text{ mV}$ et $V_3(0) = -10 \text{ mV}$

Mais alors on pourrait se demander quel est l'intérêt d'utiliser ce modèle simplifié si on observe les mêmes comportements? Et bien parce que ce modèle à deux degrés de liberté permet d'étudier mathématiquement ce qui se produit grâce à des arguments géométriques et l'étude dans le plan de phase.

Si $(V(t), n(t))$ est une solution du système, alors pour chaque instant t_0 , le couple $(V(t_0), n(t_0))$ définit un point dans le plan de phase. Ainsi la solution $(V(t), n(t))$ décrit une trajectoire dans ce plan. On fait apparaître également les courbes définies par $V' = f(V, n) = 0$ et $n' = g(V, n)$ que l'on appelle respectivement V-nullcline et n-nullcline. Leur intersection correspond au point fixe du système. On étudie dans la partie suivante la stabilité de ces points fixes.

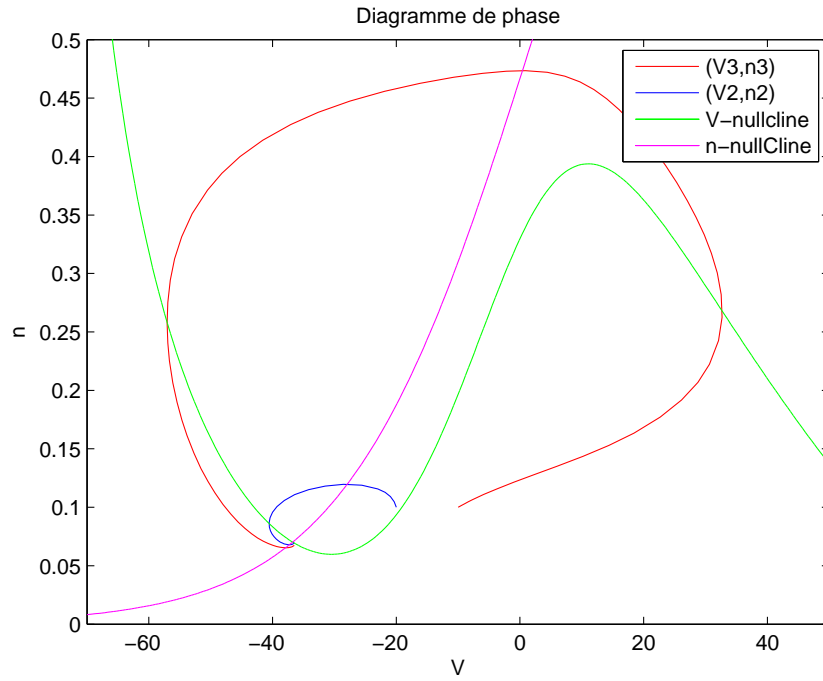


FIGURE 9 – Simulation de Morris-Lecar pour $I_{app} = 60 \text{ mA}$ et représentation dans le plan de phase (V, n) des réponses V_2 et V_3 ainsi que des nullclines $V' = 0$ et $n' = 0$

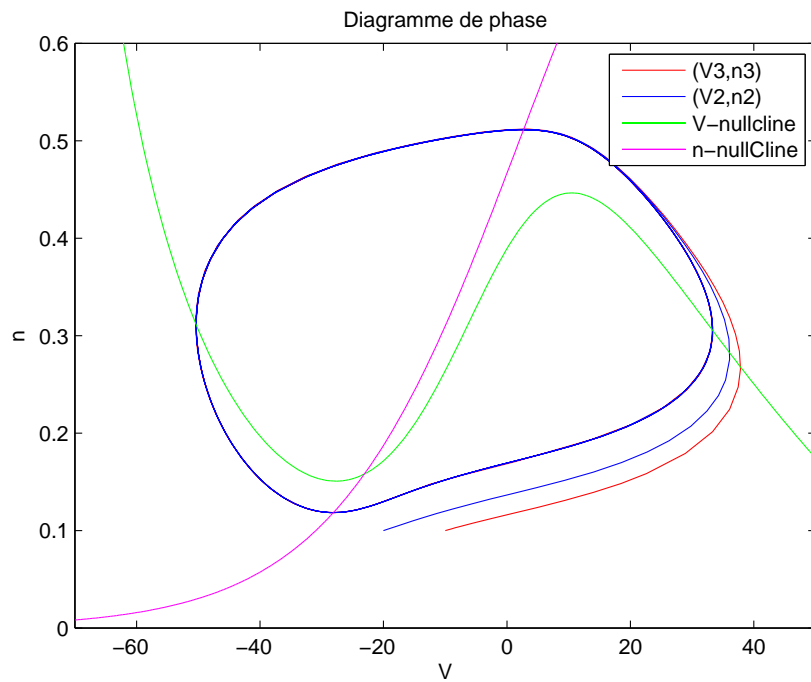


FIGURE 10 – Simulation de Morris-Lecar pour $I_{app} = 100 \text{ mA}$ et représentation dans le plan de phase (V, n) des réponses V_2 et V_3 ainsi que des nullclines $V' = 0$ et $n' = 0$

3.2 Etude de stabilité du point fixe

L'outil permettant cette étude est la linéarisation du champ de vecteurs au point fixe. Ce qui nous amène à considérer la matrice jacobienne en ce point, où V_R est le potentiel de repos (abscisse du point fixe) :

$$M = \begin{bmatrix} \frac{\partial f}{\partial V}(V_R, n_R) & \frac{\partial f}{\partial n}(V_R, n_R) \\ \frac{\partial g}{\partial V}(V_R, n_R) & \frac{\partial g}{\partial n}(V_R, n_R) \end{bmatrix}$$

Ce qui nous donne

$$M = \begin{bmatrix} -\frac{\partial I_{ion}(V_R, n_R)}{\partial V}/C_M & -g_K(V_R - E_K)/C_M \\ \phi n'_\infty(V_R)/\tau_n(V_R) & -\phi/\tau_n(V_R) \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Il est clair que $c > 0$ et $d < 0$ car les variables intervenant sont positives et $b < 0$ car le potentiel de Nerst E_K est inférieur au potentiel de repos V_R . C'est donc le signe de a qui détermine celui des valeurs propres et donc la stabilité du point fixe.

- Si $a < 0$ alors $Tr(M) = a + d < 0$ et $det(M) = ad - bc > 0$ donc le produit des valeurs propres est positif donc elles sont de même signe et comme leur somme est négative on en déduit qu'elles sont toutes deux négatives et il s'en suit que le point fixe est stable. Cela s'interprète géométriquement par une pente négative de la V-nullcline au voisinage du point fixe V_R , puisque à ce voisinage :

$$\frac{d\tilde{V}}{dt} = a\tilde{V} + b\tilde{n} = 0 \Leftrightarrow \tilde{n} = -\frac{a}{b}\tilde{V}$$

la pente vaut $-a/b$ avec $a < 0$ et $b < 0$ d'où le résultat.

- Si $a > 0$ (pente positive) alors il y a deux cas à distinguer :

- ★ Si la pente de la n-nullcline (qui est toujours positive car $-c/d > 0$) est inférieure à la pente de la V-nullcline (i.e $-a/b > -c/d$ soit $ad - bc < 0$) alors le déterminant est négatif et donc l'une des valeurs propres est positive d'où un point fixe instable.

- ★ Au contraire si la pente de la n-nullcline est supérieure à la pente de la V-nullcline alors la stabilité du point fixe est déterminée par le signe de la trace : stable si $a + d < 0$ et instable si $a + d > 0$. Comme $a > 0$ et $d = -\phi/\tau_n(V_R)$, il s'en suit que le point fixe est instable si ϕ (qui gouverne la rapidité du flux potassium) est suffisamment petit.

3.3 Interprétation des phénomènes

On peut désormais expliquer les phénomènes précédents, l'apparition ou non d'un potentiel d'action en fonction de la condition initiale, le comportement de la solution en fonction du courant appliqué, etc. Quand on change la valeur du courant appliqué on déplace les nullclines et de surcroît le point fixe, c'est la position de celui-ci sur la V-nullcline qui altère le comportement des solutions.

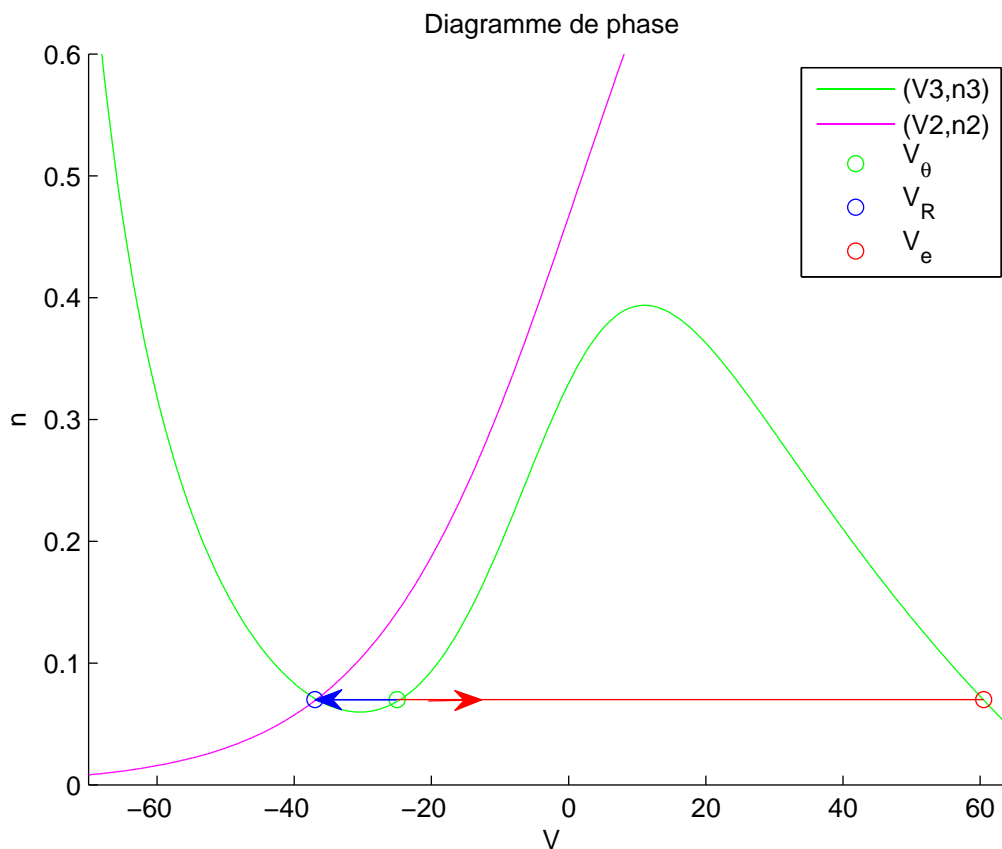
Cas où le courant appliqué vaut $I_{app} = 100 \text{ mA}$

On voit sur la figure 10 que le point fixe se trouve sur la pente positive de la V-nullcline, et ϕ n'est pas suffisamment petit donc V_R est instable, on obtient un cycle limite périodique.

Cas où le courant appliqué vaut $I_{app} = 60 \text{ mA}$

On voit sur la figure 9 que le point fixe se trouve sur la pente négative de la V-nullcline, donc d'après le premier point ($a < 0$) il est stable. Les trajectoires dans le plan de phase finissent fatalement au point fixe, retour à l'état de repos. Mais suivant la condition initiale (position du point $(V_i(t_0), n_i(t_0))$ dans le plan de phase) l'allure de la trajectoire n'est pas la même : en effet pour $(V_2(t_0), n_2(t_0)) = (-20, 0.1)$ situé à gauche de la branche centrale de la V-nullcline, le potentiel membranaire décroît immédiatement vers l'état de repos, tandis que pour $(V_3(t_0), n_3(t_0)) = (-10, 0.1)$ situé cette fois à droite on a initialement une augmentation du potentiel membranaire (correspondant au potentiel d'action) avant le retour à l'état de repos.

Afin d'expliquer cette différence on se place dans le cas extrême où ϕ est très petit, ainsi la dérivée n' varie très peu donc $n \simeq n_R$. Il y a 2 autres points d'équilibre intersection de la V-nullcline avec la droite $n = n_R$ que l'on note V_θ et V_e . Une étude de leur stabilité montre que le premier est instable tandis que le second est stable. Par conséquent selon que la condition initiale se trouve à gauche ou à droite de V_θ les trajectoires sont initialement attirées vers V_R ou V_e . Le potentiel V_θ est donc le seuil qui distingue l'apparition ou non d'un potentiel d'action.



Pour un ϕ petit mais pas trop (typiquement comme dans notre première simulation où $\phi = 0.04$), la dérivée n' n'est pas nulle mais exerce tout de même une force de rappel, la trajectoire $(V_3(t), n_3(t))$ se dirige bien initialement vers V_e mais elle est davantage attirée vers le point fixe de repos V_R ce qui explique qu'elle décrit une courbe dans le sens anti-horaire avant de s'écraser sur le point fixe.

remarque : si ϕ est suffisamment petit, on peut montrer que la courbe longe la V-nullcline comme ci-dessous, on obtient comme un phénomène d'hysteresis. On ne rentrera pas dans les détails, l'étude du système dynamique de cette forme :

$$\begin{cases} \frac{dV}{dt} = f(V, n) \\ \frac{dn}{dt} = \epsilon g(V, n) \end{cases}$$

est l'objet d'un chapitre de la référence [1].

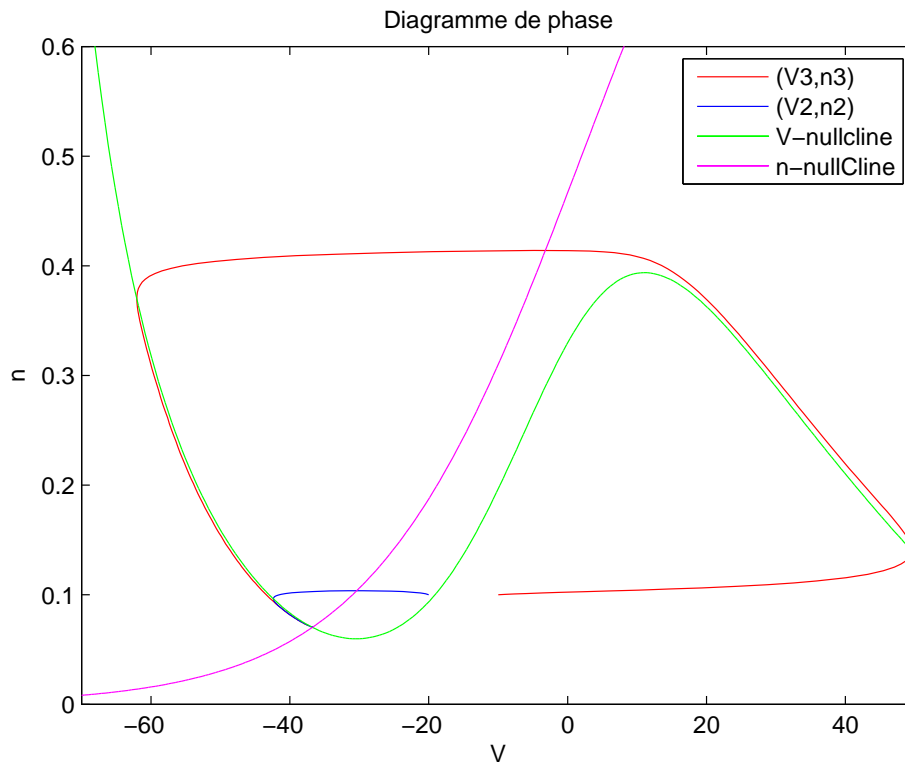


FIGURE 12 – Simulation de Morris-Lecar pour $I_{app} = 60 \text{ mA}$ et $\phi = 0.004$, trajectoires dans le plan de phase

remarque 2 : pour un ϕ quelconque il est légitime de se demander à quelle condition un spike est émis ou non, autrement dit déterminer une fonction seuil. Dans le cas précédemment où $\phi \simeq 0$ on a vu que le seuil était tout simplement le point V_θ . On a intuité également que c'est la branche du milieu de la V-nullcline qui jouait un rôle important, on y reviendra plus loin.

Cas où le courant appliqué vaut $I_{app} = 90 \text{ mA}$

Il reste ce dernier cas que nous n'avons pas encore illustré, on observe une bistabilité entre un point fixe et un cycle limite, c'est-à-dire que pour une condition initiale proche du point fixe

la trajectoire s'enroulera autour de celui-ci, sinon la réponse sera périodique et la trajectoire s'approche du cycle limite.

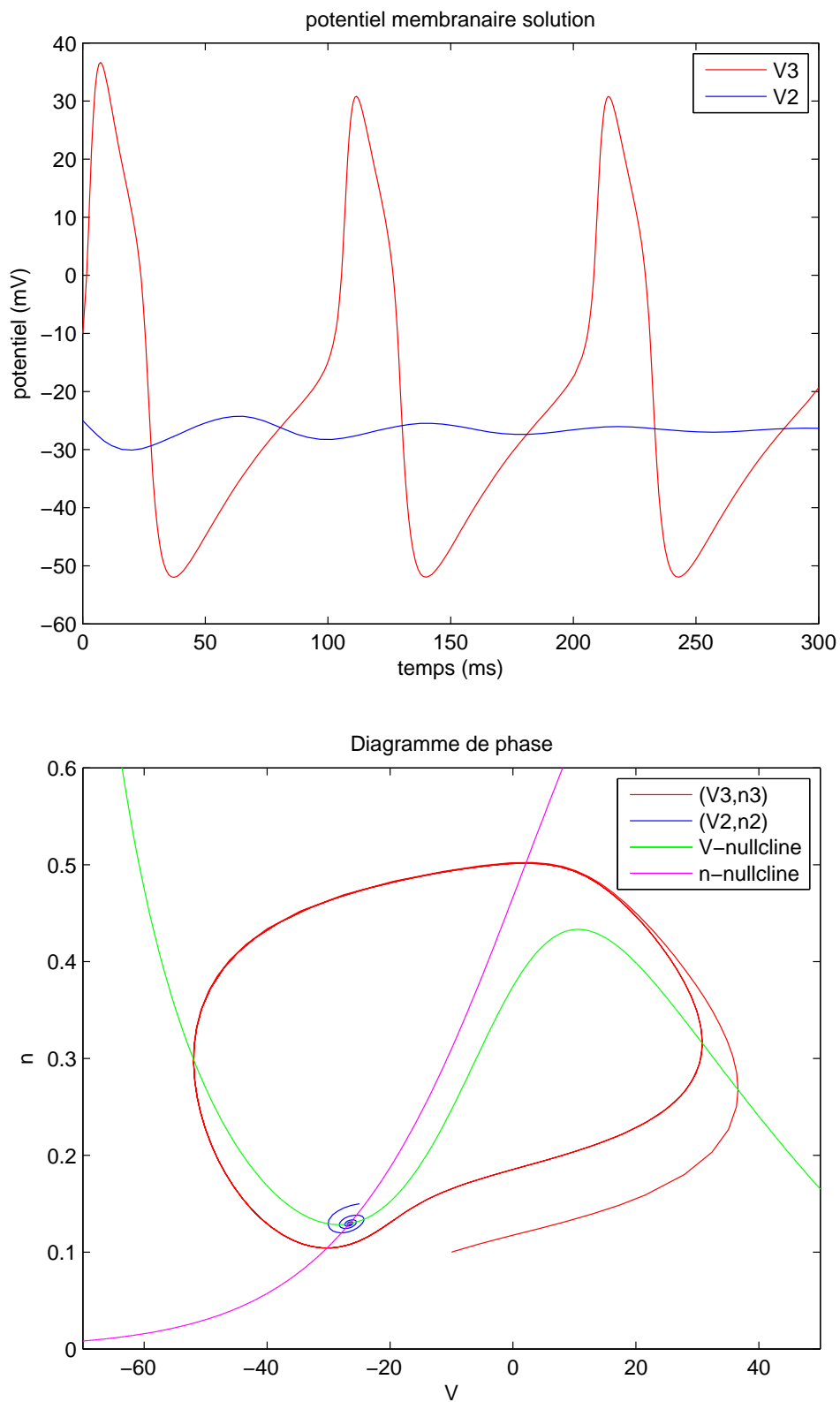


FIGURE 13 – Simulation de Morris-Lecar pour $I_{app} = 90 \text{ mA}$, la trajectoire en bleue s'enroule autour du point fixe tandis que la trajectoire en rouge s'approche du cycle limite

3.4 Bifurcation de Hopf

On a vu qu'en fonction du courant appliqué le comportement des solutions et le point fixe diffèrent. On peut facilement tracer pour chaque I_{app} le potentiel de repos correspondant et étudier sa stabilité. On constate alors que pour $I_{app} < 94$ et $I_{app} > 212$ il est stable, sinon il est instable. A ces frontières on observe ce que l'on appelle une bifurcation, le système dynamique est sensible à un paramètre (ici la valeur du courant injecté) et fournit des solutions radicalement différentes suivant celui-ci. C'est ce que nous avons démontré auparavant avec la jacobienne dans l'étude de stabilité, un point fixe étant stable si les valeurs propres de cette dernière ont toute une partie réelle négative, une bifurcation (ici un changement de stabilité) apparaît si au moins une valeur propre franchit l'axe imaginaire quand le paramètre I_{app} varie. Cette discussion est largement abordée dans les références [6], [8] et [9].

4 Le modèle intègre-et-tire

Nous avons pu constater que le modèle de Hodgkin-Huxley décrivait précisément le comportement biologique du neurone mais il se prête mal à la simulation de réseau de neurones en raison de sa complexité. En particulier le modèle décrit précisément la dynamique du spike, et durant ces 3 ms le potentiel croît et décroît brusquement, ce qui est très coûteux dans les résolutions numériques à pas adaptatif. En fin de compte on connaît l'allure d'un spike qui est sensiblement toujours de la même forme. Ce qui nous intéresse surtout c'est de savoir *quand* ils sont émis. C'est pourquoi d'autres modèles mathématiques plus simples ont été proposés de façon à décrire au mieux le comportement de certains types de neurones tout en gardant une complexité raisonnable. Nous allons introduire une famille de modèles appelés « intègre-et-tire » qui ont l'avantage d'être mathématiquement exploitables et capables de reproduire une grande variété de comportements du neurone.

4.1 Formulation mathématique

La forme générale de ces modèles est du type :

$$\begin{cases} \dot{v} = F(v) - w + I_{app} \\ \dot{w} = a(bv - w) \\ \text{si } v \geq v_{seuil} \\ \text{alors } v \leftarrow v_{reset} \text{ et } w \leftarrow w + d \end{cases}$$

Où v représente le potentiel membranaire du neurone et w représente la variable d'adaptation qui prend en compte l'activation des courants ioniques potassium et l'inactivation des courants ioniques sodium, et contribue à un feed-back sur v . Quand v dépasse un certain seuil (à fixer), on considère qu'il y a un spike émis et on réinitialise le potentiel membranaire à v_{reset} et on augmente la valeur de la variable d'adaptation de $d > 0$.

Le paramètre a représente une échelle de temps de la variable w , le paramètre b décrit la sensibilité de w aux fluctuations de v , et d permet la réinitialisation de la variable d'adaptation. Ces interprétations sont résumées sur la figure 14 ci-après :

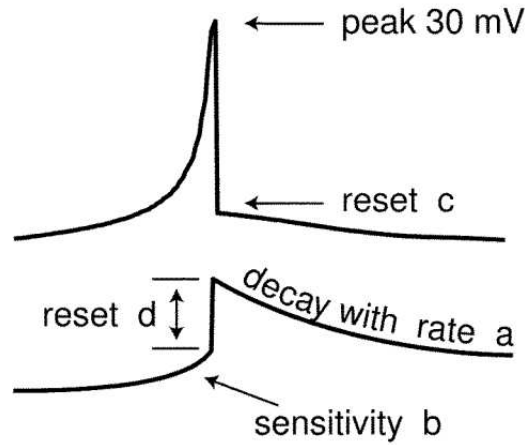


FIGURE 14 – Interprétation des paramètres d'un modèle intègre-et-tire

Par exemple, si on constate expérimentalement que lorsque le potentiel v est supérieur à 30 mV il y a automatiquement un spike émis alors on peut prendre cette valeur pour seuil, ainsi on ne simule que la courbe jusqu'à ce qu'elle atteigne le seuil, le reste étant ignoré. En stockant les instants t où le seuil a été dépassé on énumère ainsi tous les instants de spike. J'ai simulé sous Matlab certains modèles IF présentés dans la référence [2].

Par exemple le modèle d'Izhikevich considère :

$$\begin{cases} F(v) = 0.04v^2 + 5v + 140 \\ a = 0.02 \\ b = 0.2 \\ v_{seuil} = 30 \text{ mV} \\ I_{app} = 20 \mu A \end{cases}$$

- Pour $v_{reset} = -65 \text{ mV}$, $d = 8$ on obtient des spikes émis régulièrement :

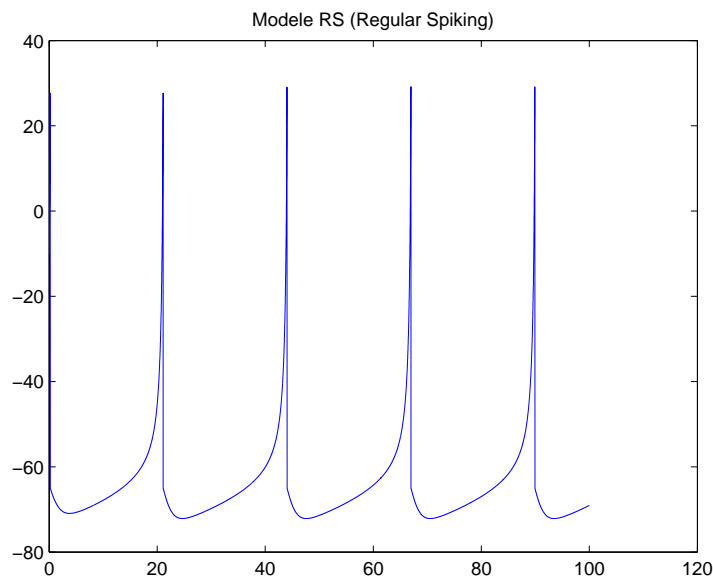


FIGURE 15 – Pattern Regular Spiking obtenu par le modèle d'Izhikevich avec $a = 0.02$, $b = 0.2$, $v_{reset} = -65 \text{ mV}$ et $d = 8$

- Pour $v_{reset} = -50 \text{ mV}$, $d = 2$ on obtient ce que l'on appelle des *burst* :

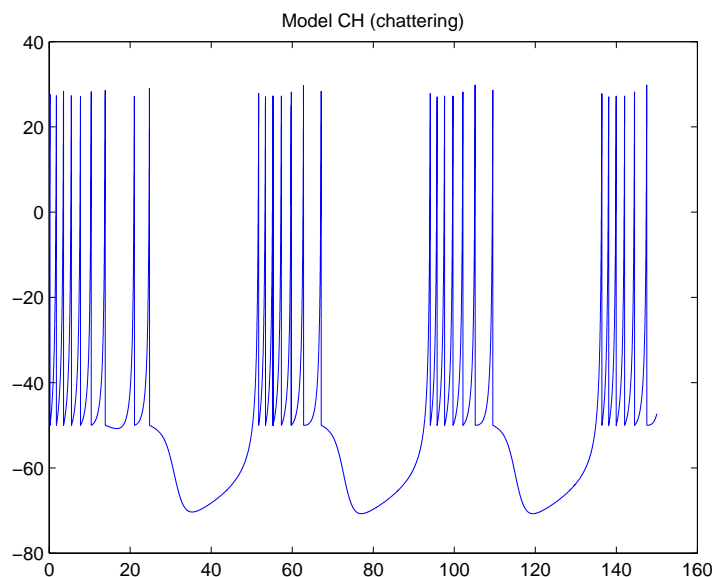


FIGURE 16 – Pattern Chattering obtenu par le modèle d'Izhikevich avec $a = 0.02$, $b = 0.2$, $v_{reset} = -50 \text{ mV}$ et $d = 2$

4.2 Introduction d'un nouveau type de seuil

Dans les formulations précédentes, le seuil était fixé tout au long de l'expérience, comme dans le modèle de Morris-Lecar où ϕ était très petit : c'est V_θ qui jouait le rôle du seuil. Un seuil fixe portant sur V se représente dans le plan de phase par une droite verticale passant par v_{seuil} , dès qu'un point du plan franchit cette délimitation on provoque le spike.

En 2009 l'INCF (the International Neuroinformatics Coordinating Facility) a lancé une compétition (référence [10]) visant à départager des modèles mathématiques devant reproduire au mieux des comportements de neurones biologiques, les auteurs de [4] ont utilisé un seuil variable au cours du temps, sous la forme d'une somme d'exponentielles de manière à introduire artificiellement une période réfractaire du neurone. Cependant aucun argument biologique ne justifie cette forme de seuil.

Afin de déterminer un seuil plus naturel nous sommes partis du modèle de Hodgkin-Huxley réduit, nous avons discrétisé le plan de phase (V, n) et pour chaque condition initiale $(V_i(0), n_i(0))$ nous avons calculé la trajectoire $(V_i(t), n(t))$ et $\max_t V_i(t)$. En fonction de cette dernière valeur nous avons coloré dans le plan le point $(V_i(0), n_i(0))$ suivant une échelle de couleurs (du bleu si le max est faible, autrement dit s'il n'y a pas de spike, vers le rouge sinon). On obtient la coloration suivante (ainsi que les nullclines en vert et jaune) :

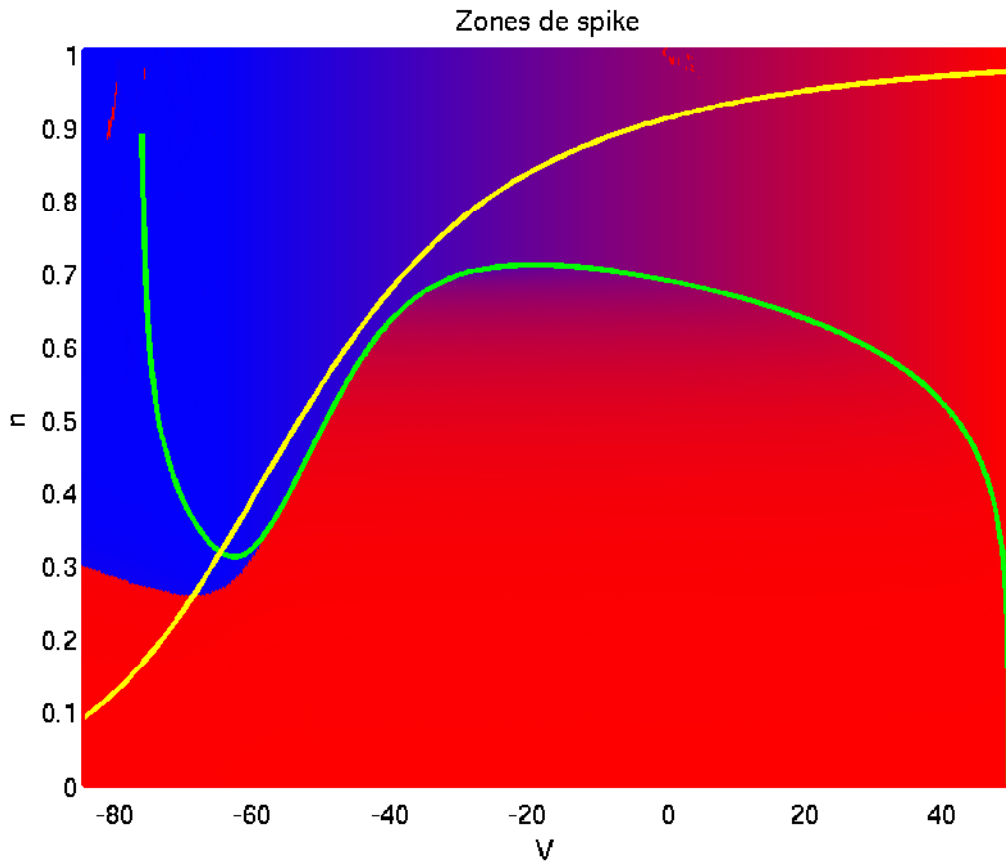


FIGURE 17 – Coloration du plan de phase : pour chaque condition initiale $(V(0), n(0))$ on affecte au point correspondant une couleur suivant la valeur du potentiel maximal atteint par la trajectoire

On voit clairement qu'un seuil du type $v = \text{constante}$ ne correspond pas à la réalité (si c'était le cas la zone de délimitation serait verticale), il semblerait que la fonction seuil possède à peu près le même comportement que la V -nullcline (sauf dans les potentiels faibles). On peut donc dans une première approche approximer la délimitation par une droite affine, par exemple au point d'inflexion de la courbe.

Pour cela j'ai utilisé Maple qui présente l'avantage d'effectuer du calcul formel. J'ai entré l'expression $f(V, n)$ puis j'ai calculé $f_v(V, n) = \frac{\partial f}{\partial V}(V, n)$ et $f_n(V, n) = \frac{\partial f}{\partial n}(V, n)$. J'ai ensuite étudié les variations de la tangente dont la pente p au potentiel V est donnée par :

$$p : V \mapsto -\frac{f_v(V, n)}{f_n(V, n)}$$

On trouve alors numériquement la droite $n = 0.0188V + 1.4322$ qui est bien tangente au point d'inflexion :

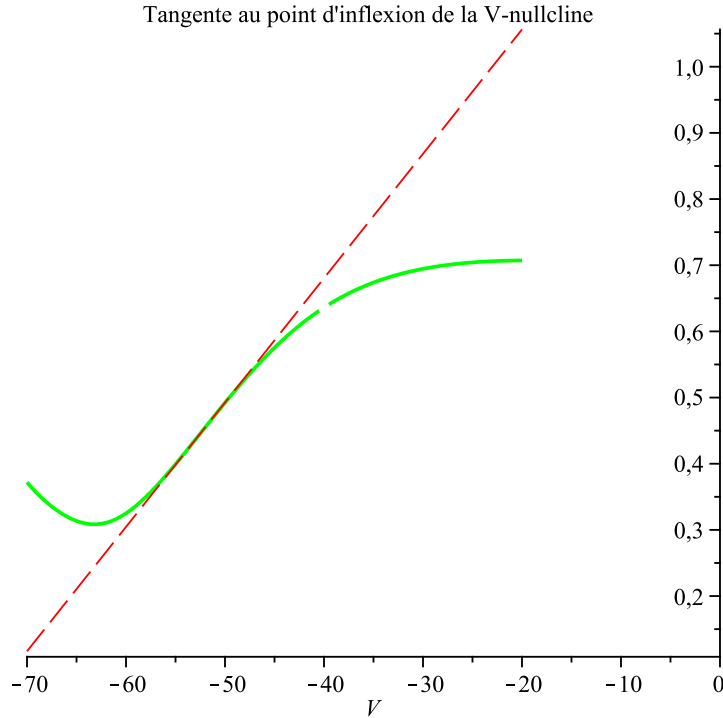


FIGURE 18 – On considère pour droite seuil la tangente à la V-nullcline au point d’inflexion

Doté de ce nouveau seuil notre modèle IF devient :

$$\begin{cases} \dot{v} = F(v) - w + I_{app} \\ \dot{w} = a(bv - w) \\ \text{si } \alpha v + \beta w = \gamma \\ \text{alors } v \leftarrow v_{reset} \text{ et } w \leftarrow w + d \end{cases}$$

remarque : en effectuant le changement de variables $v_2 = v$ et $w_2 = \frac{\alpha}{\beta}v + w$ on obtient le système suivant :

$$\begin{cases} \dot{v}_2 = F(v_2) - w_2 + I_{app} \\ \dot{w}_2 = a'(b'v_2 - w_2) \\ \text{si } w_2 = \gamma \\ \text{alors } v \leftarrow v_{reset} \text{ et } w \leftarrow w + d \end{cases}$$

Avec $a' = ab > 0$ et $b' = c/b\beta < 0$. On voit que la condition de tir porte sur la variable d’adaptation et que le signe des paramètres a changé.

4.3 Approximation du modèle de Hodgkin-Huxley

Nous avons identifié précédemment le seuil utilisé pour nos simulations. Nous pouvons d’ores et déjà resimuler le modèle de Hodgkin-Huxley réduit pour s’assurer que les temps de spike avec ce nouveau seuil coïncident avec le modèle d’origine. Pour cela nous avons soumis les deux modèles à un courant non plus constant mais aléatoire, qui simule la transmission d’information provenant de plusieurs neurones périphériques. Cela se traduit dans notre implémentation par l’injection d’une fonction `input(t,v)` dans la première équation du modèle de HH réduit. Etant donné qu’on a besoin de tester la condition de seuil au fur et à mesure de l’évolution du système, on ne peut plus avoir recours à la fonction `ode` prédéfinie de Matlab, j’ai donc effectué un schéma d’Euler.

remarque : ci-dessous on peut voir l'allure des conductances aléatoires $g_{exc}(t)$ et $g_{inh}(t)$ que nous avons générées par un processus de Poisson (voir fichier `init_conductance.m` pour les formules). Le courant aléatoire produit $y = \text{input}(t, v)$ est donc $y = g_{exc}(t)(E_{exc} - v) + g_{inh}(t)(E_{inh} - v)$ (avec $E_{exc} = -75 \text{ mV}$ et $E_{inh} = 0 \text{ mV}$) soit la contribution d'un courant excitateur et inhibiteur.

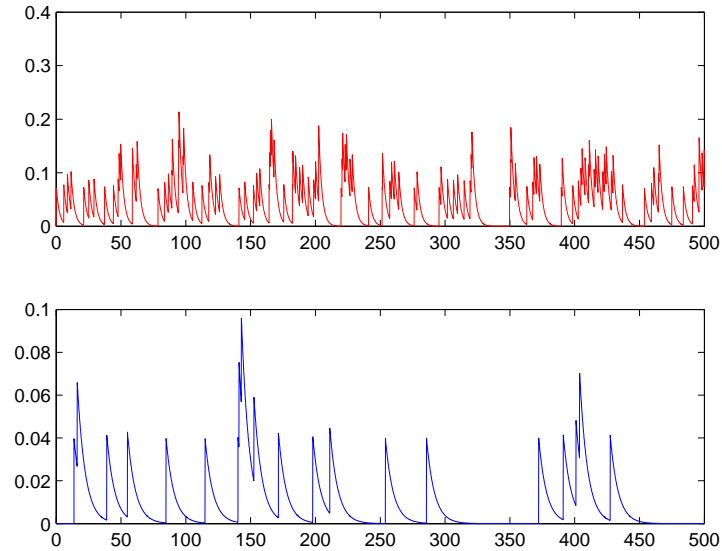


FIGURE 19 – En haut la représentation de $g_{exc}(t)$ et en bas de $g_{inh}(t)$

Soumis à ce courant, le modèle de Hodgkin-Huxley réduit émet des spikes au cours du temps (en bleu sur la figure 16). On s'assure que l'introduction du seuil ne modifie pas les temps de spike (pour cela nous avons introduit artificiellement une période réfractaire de 3 ms [voir code `maincurrent.m`]), et effectivement les potentiels tronqués (en rouge) sont bien synchronisés avec ceux d'origine :

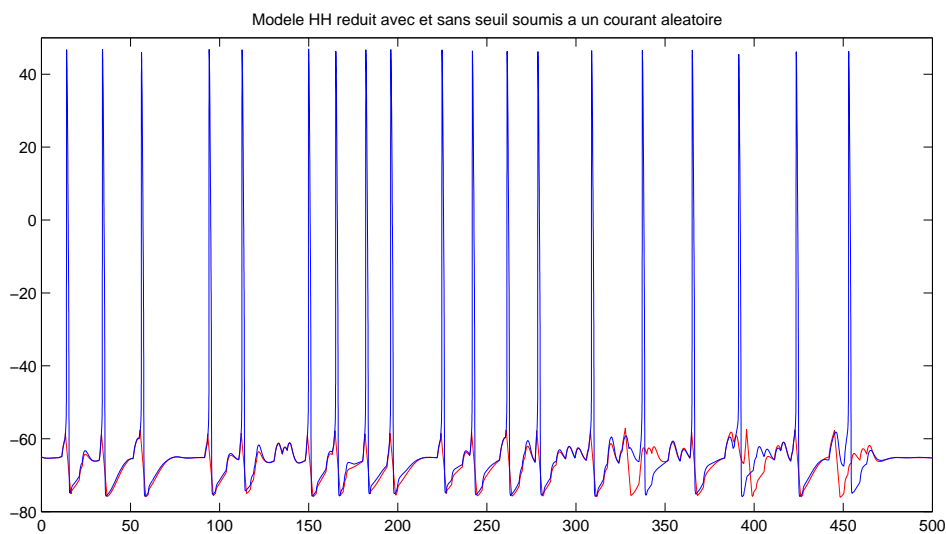


FIGURE 20 – Synchronisation des modèles HH avec et sans seuil

Ce qu'il serait encore plus intéressant de faire, serait de déterminer des expressions plus « simples » de $f(V, n)$ et $g(V, n)$, dans l'optique d'être plus performant informatiquement. J'ai donc tenté dans un premier temps d'approcher les nullclines par des polynômes. Pour cela j'ai discrétisé l'axe des abscisses $x_k = a + kh$ de manière non uniforme (grosso modo j'ai discrétisé plus finement sur les segments où la V-nullcline variait plus vite, de façon à garder suffisamment de points sur celle-ci) et j'ai calculé numériquement y_k tel que $f(x_k, y_k) = 0$. J'obtiens ainsi un nuage de points (x_k, y_k) sur la V-nullcline et je peux alors calculer les coefficients du polynôme de façon à ce qu'il approche au mieux le nuage au sens de l'erreur quadratique.

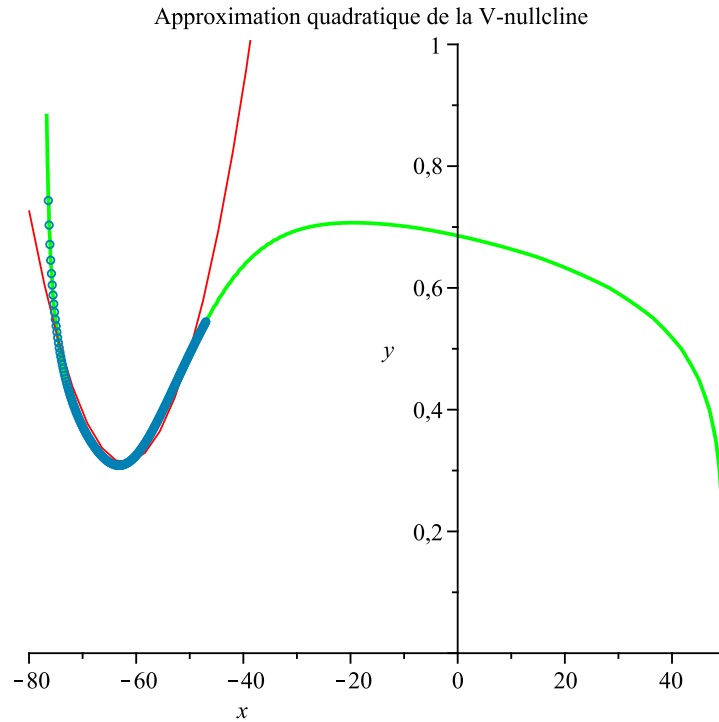


FIGURE 21 – Minimisation de la V-nullcline par la méthode des moindres carrés

Je me suis limité à la zone $[-80, -40]$ pour la simple et bonne raison que notre modèle ne simulera jamais au delà du seuil donc seule la partie convexe de la courbe nous intéresse, l'équation du polynôme quadratique l'approchant ci-dessus a pour équation :

$$P(V) = 1,27 \cdot 10^{-3} V^2 + 0,16V + 5,21$$

De même j'ai approché la n-nullcline par une droite, simplement la tangente au point d'inflexion, à savoir (calculs identiques à celui du seuil) :

$$n = 0,0159V + 1,343$$

Je me suis alors empressé de simuler le modèle IF correspondant en prenant $F(V) = P(V)$ et la droite ci-dessus pour l'expression de dw/dt . Ma déception fut grande quand je vis que la réponse n'avait rien de comparable avec HH, et pour cause : la seule approximation des nullclines ne peut rendre compte de la dynamique du système puisqu'elles caractérisent seulement les points où la dérivée s'annule!

Ce qu'il faudrait c'est non seulement approcher les nullclines mais aussi la surface $z = f(V, n)$ et $z = g(V, n)$ (les nullclines sont l'intersection de ces surfaces avec le plan $z = 0$). Pour cela j'ai utilisé Matlab (voir fichiers `minimization{f,g}.m`), le programme consiste à déterminer un quadruplet $(\alpha, \beta, \gamma, \mu)$ de sorte que la surface quadratique $z = \alpha V^2 + \beta V + \gamma + \mu n$ approche au mieux les surfaces $f(V, n)$ et $g(V, n)$ pour la norme de L^2 , c'est-à-dire déterminer :

$$\min_{(\alpha, \beta, \gamma, \mu) \in \mathbb{R}^4} \int_{V_{min}}^{V_{max}} \int_{n_{min}}^{n_{max}} [f(V, n) - (\alpha V^2 + \beta V + \gamma + \mu n)]^2 dn dV$$

J'ai effectué cette minimisation sur le pavé $[-60, -30] \times [0.1, 0.4]$, car des singularités semblaient poser problème dans le calcul de l'intégrale double (il n'y avait pas convergence de l'algorithme, même en diminuant la tolérance d'erreur). Cela dit c'est la zone qui nous intéresse principalement pour les mêmes raisons que précédemment (au delà du seuil on ne simule pas).

On détermine ainsi les surfaces approchantes :

$$z = f(V, n) \simeq_{L^2} z = 3V^2 + 374V + 10552 - 1170n$$

$$z = g(V, n) \simeq_{L^2} z = 4.10^4 V + 0,27 - 0,27n$$

Nous les représentons ci-dessous :

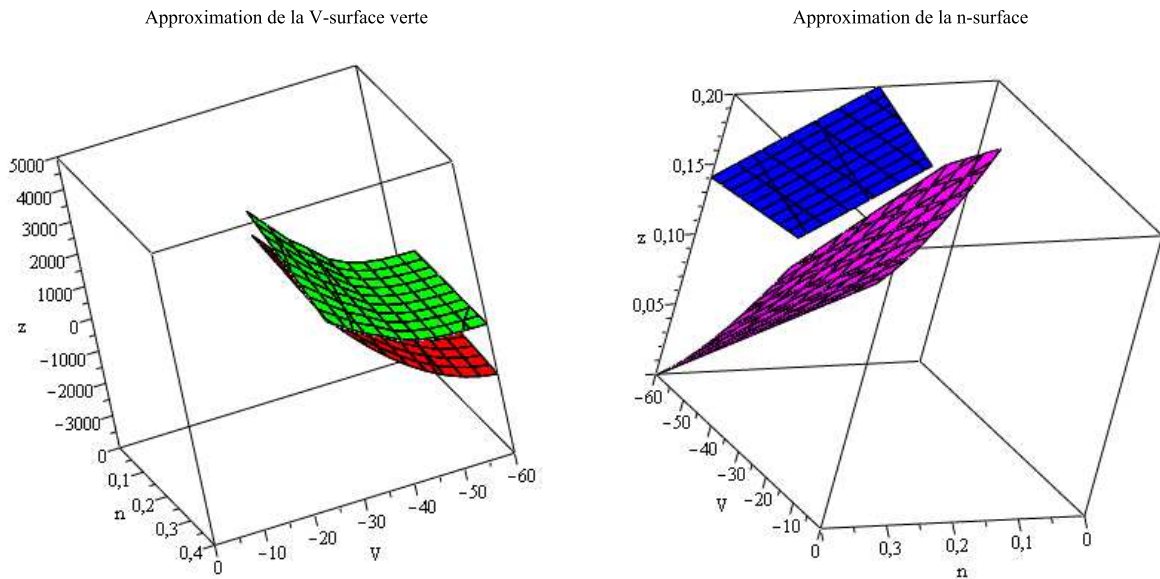


FIGURE 22 – Approximation des surfaces f et g par des surfaces quadratiques

L'approximation paraît correcte. Seulement si on coupe par le plan $z = 0$ on met en évidence un problème sur la figure 23 :

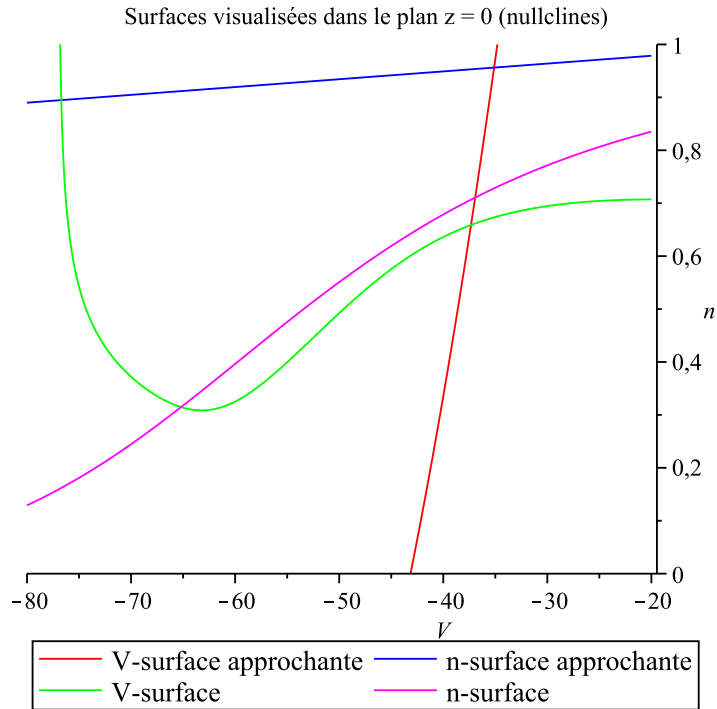


FIGURE 23 – Problème des nullclines résultantes

On a le problème inverse à celui qu'on exposait plus haut : les surfaces $z = f(V, n)$ et $z = g(V, n)$ sont maintenant bien approchées, mais les nullclines résultantes (obtenues en coupant par $z = 0$) n'ont plus du tout l'allure d'origine et elles se coupent vers $-40 mV$, le potentiel de repos ne correspond donc plus du tout. Et effectivement encore une fois la simulation de ces nouvelles équations dans le modèle IF n'a rien donné.

Ce qu'il faudrait idéalement faire, c'est effectuer une minimisation sous contrainte, c'est-à-dire approximer au mieux les surfaces tout en conservant des nullclines pertinentes. Je n'ai pas eu le temps d'exploiter cette idée.

4.4 Discussion : application à des données biologiques

Nous avons effectué différents nouveaux degrés de réduction du modèle de Hodgkin-Huxley dans le but de simuler les neurones avec des équations plus simples. La plupart des modèles intègre-et-tire sont déterminés empiriquement. Une fois ceux-ci établis il faut pouvoir vérifier qu'ils permettent effectivement de simuler aussi bien le comportement neuronal que le modèle complet HH, c'est-à-dire qu'ils permettent en jouant sur les paramètres de reproduire toute sorte de patterns (par exemple des *burst*, voir ref. [2] et [3]). Autrement dit la première étape est de vérifier que le nouveau modèle approche bien celui de HH, c'est ce que nous avons fait précédemment en le soumettant à un courant aléatoire que nous avons créé spécialement à ces fins.

Par ailleurs il faudrait vérifier également que c'est un modèle intéressant pour la prédiction. Par exemple lors du challenge mentionné plus haut, les participants disposaient de données biologiques : la réponse d'un vrai neurone soumis à un courant aléatoire sur les 40 premières secondes de l'expérience, que j'ai représenté ci-dessous avec Matlab (fichier `spike_train.m`) :

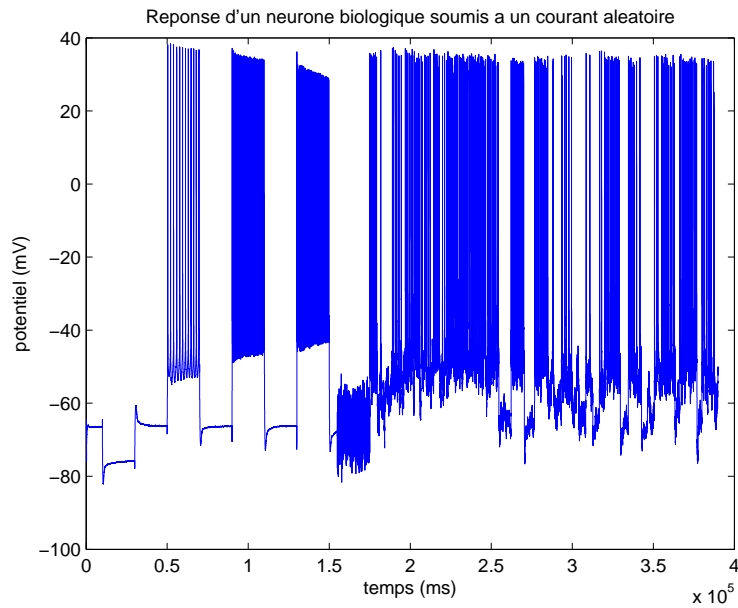


FIGURE 24 – Réponse d'un neurone biologique soumis à un courant donné pendant 40s

Connaissant alors les valeurs du courant sur toute la durée de l'expérience (à savoir 60 secondes), ils devaient alors soumettre leur modèle à ce courant pour d'une part ajuster leurs paramètres sur les 40 premières secondes de façon à coller au mieux aux spikes émis, puis fournir ce qu'ils obtenaient pour les 20 dernières secondes (prédiction de spikes). Les organisateurs qui avaient quant à eux la réponse du neurone sur toute la durée de l'expérience pouvaient alors comparer sur les 20 dernières secondes les résultats des divers participants, pour voir lequel prédisait au mieux les derniers spikes. La comparaison se fait grâce à des critères statistiques qui sont développés dans la référence [4].

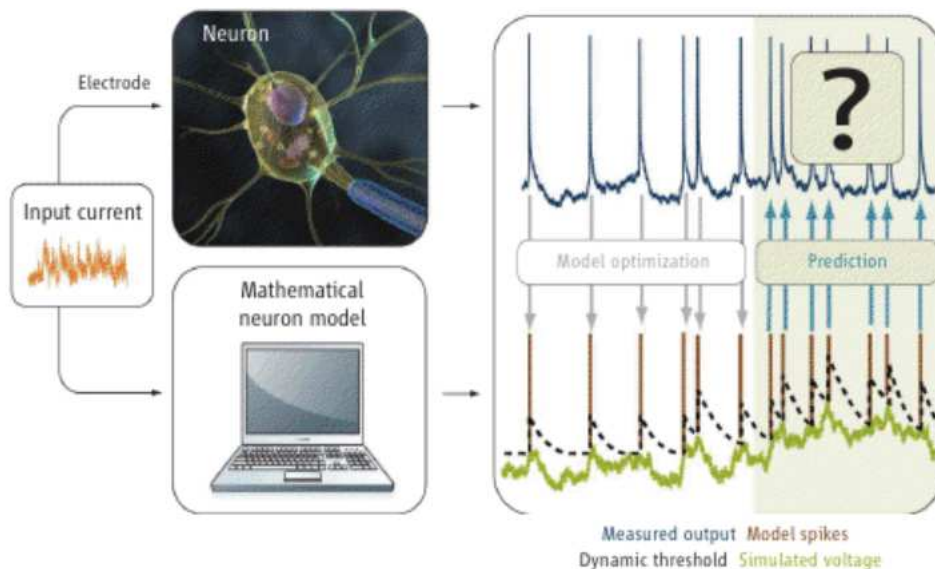


FIGURE 25 – Principe de simulation et de prédiction de spikes

5 Conclusion

5.1 Perspectives futures

L'idée nouvelle que l'on souhaitait développer était l'introduction d'un seuil variable, qui se justifie biologiquement (c'est-à-dire pas un simple seuil de coupure $V \geq V_{seuil}$ mais étudier les zones où le neurone tire afin de mettre en évidence une délimitation que l'on prendrait pour seuil); et qui de ce fait améliore les performances de simulation (en arrêtant la simulation plus tôt qu'avec le seuil fixe). Nous avons testé cela sur le modèle de Hodgkin-Huxley, puis nous espérons pouvoir en dégager un modèle intègre-et-tire satisfaisant. Afin de répondre à cette question il faudrait refaire les calculs de minimisations entre les surfaces en tenant compte des nullclines, puis resimuler le modèle obtenu en s'assurant qu'il se comporte comme HH. Puis on pourrait confronter notre modèle aux données biologiques de la compétition pour quantifier son pouvoir prédictif.

Indépendamment il serait intéressant d'introduire un seuil variable sur les modèles intègre-et-tire déjà existants, et mettre en évidence les bénéfices en temps de simulation. Et enfin mettre en application ce procédé pour la simulation d'un réseau de neurones.

5.2 Bilan

Au cours de ce module d'initiation à la recherche j'ai pu mettre les pieds dans les champs de modélisations en neurosciences, domaine que je souhaitais découvrir depuis longtemps. Ce fut très intéressant de pouvoir utiliser la boîte d'outils mathématiques que l'on remplit depuis la prépa, appliquée à des problèmes biologiques concrets, et d'en découvrir de nouveaux (par exemple la théorie des bifurcations) ainsi que les problèmes liés à la non-régularité des systèmes dynamiques. Etudier le comportement neuronal est un enjeu de taille, et j'ai pu entrevoir différents obstacles qui se posent actuellement aux chercheurs. Cela fut à la fois enrichissant scientifiquement mais également pour mon projet professionnel en ayant fait mes premiers pas dans le monde de la recherche.

5.3 Remerciements

Je tiens à remercier mon tuteur, Arnaud Tonnelier, pour l'aide et les connaissances qu'il m'a apporté, ainsi que le temps qu'il a consacré à répondre à mes questions et à relire mon rapport.

6 Bibliographie

- [1] Mathematical Foundations of Neuroscience, G.Bard Ermentrout & David H.Terman, 2010.
- [2] Simple Model of Spiking Neurons, Eugene M.Izhikevich, 2003.
- [3] Which Model to Use for Cortical Spiking Neurons, Eugene M. Izhikevich, 2004.
- [4] Made to order spiking neuron model equipped with a multi-timescale adaptive threshold, Ryota Kobayashi, Yasuhiro Tsubo & Shigeru Shinomoto, 2009.
- [5] A benchmark test for a quantitative assessment of simple neuron models, Renaud Jolivet, Ryota Kobayashi, Alexander Rauch, Richard Naud, Shigeru Shinomoto & Wulfram Gerstner, 2007.
- [6] Adaptive Exponential Integrate-and-Fire Model as an Effective Description of Neuronal Activity, Romain Brette & Wulfram Gerstner, 2005.
- [7] Importance of the Cutoff Value in the Quadratic Adaptive Integrate-and-Fire Model, Jonathan Touboul, 2009.
- [8] Bifurcation Analysis of general class of Nonlinear integrate-and-fire Neurons, Jonathan Touboul, 2007.
- [9] Spiking Dynamics of Bidimensional Integrate-and-Fire Neurons, Jonathan Touboul, 2009.
- [10] How Good Are Neuron Models?, Wulfram Gerstner & Richard Naud, 2009.

7 Annexes

7.1 Hodgkin.m

```
// On met le systeme sous forme differentiel y'=f(y)
// avec Y = (V,n,m,h) appartenant a R^4 et f : R^4 -> R^4
// Ici Hodgkin joue le role de f, elle calcule f(y)
5 function ypoint = Hodgkin(t,y)
// Constantes physiques
gNa=120;
gK=36;
gL=0.3;
10 eNa=50;
eK=-77;
eL=-54.4;
cM=1;
Q10=3;
15 T=6.3;
Tb=6.3;
phi=Q10^((T-Tb)/10);
current = 0; // 4 et 10 uA/cm^2
tstimul = 10;

20 // On injecte le courant a la date tstimul
if (t < tstimul)
    stimul = 0;
else
25     stimul = current;
end

// Image de y par f
ypoint(1) = 1/cM*(-gNa*y(3)^3*y(4)*(y(1)-eNa)-gK*y(2)^4*(y(1)-eK)
30     -gL*(y(1)-eL)+stimul);
ypoint(2) = phi*(alpha_n(y(1))*(1-y(2))-beta_n(y(1))*y(2));
ypoint(3) = phi*(alpha_m(y(1))*(1-y(3))-beta_m(y(1))*y(3));
ypoint(4) = phi*(alpha_h(y(1))*(1-y(4))-beta_h(y(1))*y(4));

35 ypoint = ypoint';

end

function res = alpha_n(V)
40     res = 0.01*(V+55)/(1-exp(-(V+55)/10));
end

function res = alpha_m(V)
     res = 0.1*(V+40)/(1-exp(-(V+40)/10));
45 end

function res = alpha_h(V)
     res = 0.07*exp(-(V+65)/20);
```

```

end
50 function res = beta_n(V)
    res = 0.125*exp(-(V+65)/80);
end

55 function res = beta_m(V)
    res = 4*exp(-(V+65)/18);
end

function res = beta_h(V)
60     res = 1/(1+exp(-(V+35)/10));
end

```

7.2 mainHH.m

```

function out = mainHH( )
// Temps de simulation (en ms)
tfinal=100;

5 // Conditions initiales (ici y0 est point fixe)
y0=[-65,0.3177,0.06,0.6];

// Resolution du systeme differentiel par methode de Runge-Kutta
display 'Resolution du systeme dynamique'
10 [t,y]=ode45('Hodgkin',[0,tfinal],y0);

// On recupere V, n, m et h
y1=y(:,1);
y2=y(:,2);
15 y3=y(:,3);
y4=y(:,4);

// On les trace

20 // Potentiel membranaire V
subplot(221)
plot(t,y1)
xlabel('temps (ms)');
ylabel('potentiel (mV)');
25 title('potentiel membranaire V');
axis([0 100 -70 50]);

// Variables de declenchement

30 // n
subplot(222)
plot(t,y2,'g')
xlabel('temps (ms)');
ylabel('potentiel (mV)');
35 title('n');
axis([0 100 0 1]);

```

```

// m
subplot(223)
40 plot(t,y3,'r')
   xlabel('temps (ms)');
   ylabel('potentiel (mV)');
   title('m');
   axis([0 100 0 1]);
45
// h
subplot(224)
plot(t,y4,'m')
xlabel('temps (ms)');
50 ylabel('potentiel (mV)');
   title('h');
   axis([0 100 0 1]);
end

```

7.3 Morris.m

```

function ypoint = Morris(t,y,Iapp)

// Image de y par f
5 ypoint(1) = f(y(1),y(2),Iapp);
  ypoint(2) = g(y(1),y(2));

  ypoint = ypoint';
10 end

function res = f(V,n,Iapp)
  gCa=4.4;
  gK=8;
15  gL=2;
  eCa=120;
  eK=-84;
  eL=-60;
  CM=20;
20 res = 1/CM*(-gCa*m_inf(V)*(V-eCa)-gK*n*(V-eK)-gL*(V-eL)+Iapp);
end

function res = m_inf(V)
  V1=-1.2;
25  V2=18;
  res = 0.5*(1+tanh((V-V1)/V2));
end

function res = g(V,n)
30 phi=0.0004;
  res = phi*(n_inf(V)-n)/tau_n(V);
end

```



```

function res = tau_n(V)
35     V3=2;
        V4=30;
        res = 1/cosh((V-V3)/(2*V4));
end

40 function res = n_inf(V)
        V3=2;
        V4=30;
        res = 0.5*(1+tanh((V-V3)/V4));
end

45 function out = main( )
// Temps de simulation (en ms)
tfinal=300;

50 // Conditions initiales
y03=[-10,0.1];
y02=[-20,0.1]
Iapp=100;

55 // Resolution du systeme differentiel par methode de Runge-Kutta
display 'Resolution du systeme dynamique'
[t3,y3]=ode45(@(t,y) Morris(t,y,Iapp),[0,tfinal],y03);
[t2,y2]=ode45(@(t,y) Morris(t,y,Iapp),[0,tfinal],y02);

60 // On recupere les solutions
V3=y3(:,1);
n3=y3(:,2);
V2=y2(:,1);
n2=y2(:,2);

65 // On les trace

//Potentiels membranaire V
figure(1);
70 plot(t3,V3,'r')
hold on
plot (t2,V2,'b')
line([0 100],[-36 -36],'LineStyle','--','Color','g')
xlabel('temps (ms)');
75 ylabel('potentiel (mV)');
title('potentiel membranaire solution');
leg2 = legend('V3','V2','VR');
hold off

80 // Diagramme de phase
figure(2);
plot(V3,n3,'r')
fi = @(V,n) f(V,n,Iapp);
hold on;

```

```

85 plot(V2,n2,'b');
   ez1=ezplot(fi,[-70,50]);
   set(ez1,'Color',[0 1 0]);
   ez2=ezplot('g',[-70,50]);
   set(ez2,'Color',[1 0 1]);
90 xlabel('V');
   ylabel('n');
   title('Diagramme de phase');
   axis([-70 50 0 0.6]);
   leg1 = legend('(V3,n3)', '(V2,n2)', 'V-nullcline', 'n-nullCline');
95 end

```

7.4 MorrisExe.m

```

function out = main( )
// Temps de simulation (en ms)
tfinal=300;
5
// Conditions initiales
y03=[-10,0.1];
y02=[-25,0.15]
Iapp=90;
10
// Resolution du systeme differentiel par methode de Runge-Kutta
display 'Resolution du systeme dynamique'
[t3,y3]=ode45(@(t,y) Morris(t,y,Iapp),[0,tfinal],y03);
[t2,y2]=ode45(@(t,y) Morris(t,y,Iapp),[0,tfinal],y02);
15
// On recupere les solutions
V3=y3(:,1);
n3=y3(:,2);
V2=y2(:,1);
20 n2=y2(:,2);

// On les trace

//Potentiels membranaire V
25 figure(1);
plot(t3,V3,'r')
hold on
plot (t2,V2,'b')
//line([0 700],[-36 -36],'LineStyle','--','Color','g')
30 xlabel('temps (ms)');
ylabel('potentiel (mV)');
title('potentiel membranaire solution');
leg2 = legend('V3', 'V2', 'VR');
hold off
35
// Diagramme de phase
figure(2);

```

```

    plot(V3,n3,'r')
    fi = @(V,n) f(V,n,Iapp);
40 hold on;
    plot(V2,n2,'b');
    ez1=ezplot(fi,[-70,50]);
    set(ez1,'Color',[0 1 0]);
    ez2=ezplot('g',[-70,50]);
45 set(ez2,'Color',[1 0 1]);
    // plot(-25,0.07,'o','Color','g');
    // plot(-37,0.07,'o','Color','b');
    // plot(60.5,0.07,'o','Color','r');
    // line([-37 -25],[0.07 0.07],'Color','b')
50 // line([60.5 -25],[0.07 0.07],'Color','r')
    xlabel('V');
    ylabel('n');
    title('Diagramme de phase');
    axis([-70 50 0 0.6]);
55 leg1 = legend('(V3,n3)', '(V2,n2)', 'V-nullcline', 'n-nullcline');

end

```

7.5 fi.m

```

function res = fi(v,w,Iapp)
    res = F(v,w)+Iapp;
end

5 function res = F(v,w)
    res = 0.04*v*v+5*v+140-w;
    //res = 0.00127627*v*v+0.158183*v+5.213845;
    //res = 3*v*v+374*v+10552-1170*w;
end

```

7.6 gi.m

```

function res = gi(v,w)
//a=0.0159; b=1.353;
a=0.02; b=0.2;
//res = 0.0004*v+0.2729-0.2707*w;
5 res = a*(b*v-w);
end

```

7.7 IF.m

```

function ysuiiv = IF(yn,tn,h,stimul)
// Constantes physiques

// Image de y par f
5 ysuiiv(1) = yn(1) + h*fi(yn(1),yn(2),stimul);
ysuiiv(2) = yn(2) + h*gi(yn(1),yn(2));

ysuiiv = ysuiiv';

```

10 **end**

7.8 mainIFizh.m

```
function out = mainIF()
tsimul = 150;
stimul = 20; // courant applique
step = 0.01; // pour la methode d'Euler
5 tn = 0;
yn(1) = -10;
yn(2) = 0.1;
vreset = -50;
seuilfixe = 30; // seuil au dela duquel on emet un spike
10 d = 2;
i = 1;
j = 1;

temps(i) = tn;
15 volt(i) = yn(1);

while (tn < tsimul)
    tn
    yn
20 yn = IF(yn,tn,step,stimul);
    tn = tn+step;

    if (yn(1) > seuilfixe)
        yn(1) = vreset;
25 yn(2) = yn(2)+d;
        spike(j) = tn;
        j = j+1
    end;
    i = i+1;
30 temps(i) = tn;
    volt(i) = yn(1);
end;

plot(temps,volt);
35 end
```

7.9 HHreduit.m

```
function ypoint = HHreduit(t,y)
// Constantes physiques
gNa=120;
gK=36;
5 gL=0.3;
eNa=50;
eK=-77;
eL=-54.4;
cM=1;
10 Q10=3;
T=6.3;
```

```

Tb=6.3;
phi=Q10^((T-Tb)/10);
//current = 10; // 4 et 10 uA/cm^2

```

15

```

//Image de y par f
ypoint(1) = fr(y(1),y(2));
ypoint(2) = gr(y(1),y(2));

```

20

```

ypoint = ypoint';

```

end

7.10 mainHHred.m

```

function out = main( )
// Temps de simulation (en ms)
tfinal=20;

```

```

5 a = -85; //Vmin
  b = 50; //Vmax
  nx = 24; //nb subdivisions des abscisses
  ny = 24;
  hx = (b-a)/(nx-1);
10 hy = 1/(ny-1);

```

```

// Conditions initiales (ici y0 est point fixe)

```

```

  y0=[-65,0.3177];
15 Iapp=0;

```

```

[t,y]=ode15s(@(t,y) HHreduit(t,y,Iapp),[0,tfinal],y0);

```

```

// On recupere V, n
20 y1=y(:,1);
  y2=y(:,2);
max(y1)

```

```

// if (a > -76.7)
25 //     V = [a : 0.1 : b];
  // else
  //     V = [-76.7 : 0.1 : b];
  // end
  // N = zeros(size(V,2),1);
30 // [N(1),fval] = fsolve(@(n) fr(V(1),n,Iapp),0.25);
  // N(1)
  // // V-nullcline
  // for i=2:size(V,2)
  //     i
35 //     [N(i),fval] = fsolve(@(n) fr(V(i),n,Iapp),N(i-1));
  //     N(i)
  // end;

```

```

//y(1:100)
40 plot(t,y1);
axis([0 tfinal -80 50]);
//plot(y1,y2,'b');
//fi = @(V,n) fr(V,n,Iapp);
hold on;
45 //ez1=ezplot(fi,[a,b]);
//set(ez1,'Color',[0 1 0]);
//ez1=plot(V,N,'Color',[0 1 0],'LineWidth',2);
//ez2=ezplot('gr',[a,b]);
//set(ez2,'Color',[1 0 1]);
50 xlabel('temps (ms)');
ylabel('potentiel (mV)');
title('n');
//axis([a b 0 1]);

55 end

```

7.11 mapping.m

```

function out = main( )
// Temps de simulation (en ms)
tfinal=20;

5 a = -85; //Vmin
b = 50; //Vmax
c = 1;
nx = 500; //nb subdivisions des abscisses
ny = 500;
10 hx = (b-a)/(nx-1);
hy = c/(ny-1);
cpt = 0;
k=1;

15 for i=1:nx
    for j=1:ny
// Resolution du systeme differentiel par methode de Runge-Kutta
i,j
// Conditions initiales
20 y0 = [a+(i-1)*hx,(j-1)*hy]

[t,y]=ode15s('HHreduit',[0,tfinal],y0);

// On recupere V
25 y1=y(:,1);
m=max(y1);
if m>50
    m=50;
    cpt=cpt+1;
30 end
M(i,j)=m;

```

```

X(k)=a+(i-1)*hx;
Y(k)=(j-1)*hy;
k=k+1;
35   end
end

cpt
X
40 Y

cmap = zeros(256,3);
cmap(1:128,3) = linspace(1, 0, 128); // bleu
cmap(129:end,1) = linspace(0, 1, 128); // rouge
45 cmap(129,:) = []; // efface les doublons [0,0,0]
cmap

Vmin=min(min(M))
Vmax=max(max(M))

50 size(X,2)
C = zeros(size(X,2),3);
k = 1;
for i=1:nx
55   for j=1:ny
       V=M(i,j);
       alpha=(V-Vmin)/(Vmax-Vmin);
       C(k,:)=(1-alpha)*[0 0 1]+alpha*[1 0 0];
       k = k+1;
60   end
end
C
X=X'
Y=Y'
65 Iapp=0;

if (a > -76.7)
    V = [a : 0.1 : b];
else
70   V = [-76.7 : 0.1 : b];
end

N = zeros(size(V,2),1);
[N(1),fval] = fsolve(@(n) fr(V(1),n),0.25);
75 N(1)
// V-nullcline
for i=2:size(V,2)
    i
    [N(i),fval] = fsolve(@(n) fr(V(i),n),N(i-1));
80   N(i)
end;

cpt

```

```

85 // On affiche la mapping ainsi que les nullclines
scatter(X,Y,10,C,'filled');
hold on
fi = @(V,n) fr(V,n);
ez1=plot(V,N,'Color',[0 1 0],'LineWidth',2);
90 ez2=ezplot('gr',[a,b]);
set(ez2,'Color',[1 1 0],'LineWidth',2);
xlabel('V');
ylabel('n');
axis([a b 0 c]);
95 title('Zones de spike');

```

end

7.12 minimisationf.m

```

function res = minimization( )
// condition initiale
x0 =[3,374,10552,-1170];
[x,resnorm] = fminunc(@fmin,x0);
5 x // solution de la minimisation
resnorm // erreur commise
end

// Distance L2 entre la surface f et la surface quadratique
10 function res = fmin(z)
alpha1=z(1)
beta1=z(2)
gamma1=z(3)
mu1=z(4)
15 xmin=-60
ymin=0.1
xmax=-30
ymax=0.4
ainteg = @(x,y)differ(x,y,alpha1,beta1,gamma1,mu1)
20 res = dblquad(ainteg,xmin,xmax,ymin,ymax,1.0e-3);
end

function res = differ(x,y,alpha1,beta1,gamma1,mu1)
res = (frm(x,y)-alpha1.*x.*x-beta1.*x-gamma1-mu1.*y).^2;
25 end

// surface f
function res = frm(x,y)
// Constantes physiques
30 gNa=120;
gK=36;
gL=0.3;
eNa=50;
eK=-77;

```



```

35 eL=-54.4;
    cM=1;
    res = 1/cM.*(-gNa.*m_infty(x).^3.*(0.8-y).*(x-eNa)
        -gK.*y.^4.*(x-eK)-gL.*(x-eL));
    end
40
    function res = m_infty(x)
        res = alpha_m(x)./(alpha_m(x)+beta_m(x));
    end

45 function res = alpha_m(x)
    if abs(x+40) < 0.001
        res = 1 + (x+40)./20;
    else
        res = 0.1.*(x+40)./(1-exp(-(x+40)./10));
50 end
    end

    function res = beta_m(x)
        res = 4.*exp(-(x+65)./18);
55 end

```

7.13 minimisationg.m

```

function res = minimization2( )
x0 =[0,0,0,0];
[x,resnorm] = fminunc(@gmin,x0);
x
5 resnorm
end

function res = gmin(z)
alpha1=z(1)
10 beta1=z(2)
    gamma1=z(3)
    mu1=z(4)
    xmin=-60
    ymin=0.1
15 xmax=-30
    ymax=0.4
    ainteg = @(x,y)differ(x,y,alpha1,beta1,gamma1,mu1)
    res = dblquad(ainteg,xmin,xmax,ymin,ymax,1.0e-3);
    end
20
    function res = differ(x,y,alpha1,beta1,gamma1,mu1)
    res = (grm(x,y)-alpha1.*x.*x-beta1.*x-gamma1-mu1.*y).^2;
    end

25 function res = grm(V,n)
    Q10=3;
    T=6.3;
    Tb=6.3;

```

```

phi=Q10^((T-Tb)/10);
30 res = phi*(alpha_n(V)*(1-n)-beta_n(V)*n);
end

function res = alpha_n(V)
if abs(V+55) < 0.001
35     res = 1 + (V+55)/20;
else
    res = 0.01*(V+55)/(1-exp(-(V+55)/10));
end
end

40 function res = beta_n(V)
    res = 0.125*exp(-(V+65)/80);
end

```

7.14 spike_train.m

```

/>\ Remplacer partout 'f' par 'pourcent f'
function out = spike_train(entree,sortie)
fid_entree = fopen(entree);
[tab_entree,count] = fscanf(fid_entree,'f',inf);
5 count
tab_entree = tab_entree';
fclose(fid_entree);
dim_entree = size(tab_entree,2)
//for i=1:dim_entree
10 //     disp(tab_entree(i));
//end

fid_sortie = fopen(sortie);
[tab_sortie,count] = fscanf(fid_sortie,
15     'f f f f f f f f f f f f f f f',[13 inf]);
count
tab_sortie = tab_sortie';
fclose(fid_sortie);
dim_sortie = size(tab_sortie)
20 //for i=1:1
//     for j=1:dim_sortie(2)
//         disp(tab_sortie(i,j));
//     end
25 //end

v = tab_sortie(:,5);
size(v)
/v3 = zeros(size(v),1);
30 for i=1:size(v)
    if (v(i)>=0)
        v3(i) = v(i);
    end
end

```

```

35 end

    size(v3)

    plot(1:size(v3),v3);
40 end

```

7.15 init_conductance.m

7.16 input.m

```

function y = input(t,v)
global Tinac

Eexc = 0. ;
5 Einh = -75. ;
if (t > Tinac)
    y = gexc(t)*(Eexc-v) + ginh(t)*(Einh-v) ;
else
    y = 0;
10 end;

```

end

```

function y = ginh(t)
15 global Tsim Taginh ;

n_bin = length(Taginh) ;
i = round(t/Tsim*(n_bin-1)) + 1 ;
20 y = Taginh(i) ;

```

end

```

function y = gexc(t)
25 global Tsim Tagexc ;

n_bin = length(Tagexc) ;
i = round(t/Tsim*(n_bin-1)) + 1 ;
30 y = Tagexc(i) ;

```

end

```

function y = heav(x)
35 if x>0
    y=1;
y=0;
end

```

7.17 mainIF.m

```

function out = mainIF()

init_conductance;

5  tsimul = Tsim;
   stimul = 0;
   step = 0.01;
   tn = 0;
   yn(1) = -65;
10  yn(2) = 0.3177;
   vreset = -75;
   nreset = 0.7;
   i = 1;
   j = 1;
15  refrac = 3;
   temps(i) = tn;
   volt(i) = yn(1);
   potn(i) = yn(2)

20  while (tn < tsimul)
       tn
       yn
       yn = IFc(yn,tn,step,stimul);
       tn = tn+step;

25       if (yn(2) < seuil(yn(1)))
           yn(1) = vreset;
           yn(2) = nreset;
           spike(j) = tn;
30           tn = tn+refrac;
           j = j+1
       end;
       i = i+1;
       temps(i) = tn;
35       volt(i) = yn(1);
       potn(i) = yn(2);
end;

spike
40  j
   //v=[1,1,1,1,1,1,1,1,1]
   //scatter(spike,v)

45  //plot(temps,volt);
   a=-75
   b=30

   Iapp=0;
50  if (a > -76.7)
       V = [a : 0.1 : b];
   else

```

```

        V = [-76.7 : 0.1 : b];
    end
55 N = zeros(size(V,2),1);
    [N(1),fval] = fsolve(@(n) fr(V(1),n,Iapp),0.25);
    N(1)
    // V-nullcline
    for i=2:size(V,2)
60         i
            [N(i),fval] = fsolve(@(n) fr(V(i),n,Iapp),N(i-1));
            N(i)
    end;

65 plot(volt,potn);
    //plot(y1,y2,'b');
    //fi = @(V,n) fr(V,n,Iapp);
    hold on;
    //ez1=ezplot(fi,[a,b]);
70
    ez1=plot(V,N,'Color',[0 1 0],'LineWidth',2);
    set(ez1,'Color',[0 1 0]);
    ez2=ezplot('gr',[a,b]);
    set(ez2,'Color',[1 0 1]);
75 //xlabel('temps (ms)');
    //ylabel('potentiel (mV)');
    //title('n');
    axis([-75 30 0 1]);

80 end

function res = seuil(V)
    a = 0.01879363;
    b = 1.432178915;
85 res = a*V+b;
    end

function ysuiiv = IF(yn,tn,h,stimul)
    // Constantes physiques
90
    // Image de y par f
    ysuiiv(1) = yn(1) + h*frc(yn(1),yn(2),stimul,tn);
    ysuiiv(2) = yn(2) + h*gr(yn(1),yn(2));

95 ysuiiv = ysuiiv';

    end

function res = frc(V,n,Iapp,t)
100 // Constantes physiques
    gNa=120;
    gK=36;
    gL=0.3;
    eNa=50;

```

```

105 eK=-77;
    eL=-54.4;
    cM=1;
    res = 1/cM*(-gNa*m_infty(V)^3*(0.8-n)*(V-eNa)
           -gK*n^4*(V-eK)-gL*(V-eL)+input(t,V));
110 end

function res = m_infty(V)
    res = alpha_m(V)/(alpha_m(V)+beta_m(V));
end

115 function res = alpha_m(V)
if abs(V+40) < 0.001
    res = 1 + (V+40)/20;
else
120 res = 0.1*(V+40)/(1-exp(-(V+40)/10));
end
end

function res = beta_m(V)
125 res = 4*exp(-(V+65)/18);
end

function res = g(V,n)
Q10=3;
130 T=6.3;
    Tb=6.3;
    phi=Q10^((T-Tb)/10);
    res = phi*(alpha_n(V)*(1-n)-beta_n(V)*n);;
end

135 function res = alpha_n(V)
if abs(V+55) < 0.001
    res = 1 + (V+55)/20;
else
140 res = 0.01*(V+55)/(1-exp(-(V+55)/10));
end
end

function res = beta_n(V)
145 res = 0.125*exp(-(V+65)/80);
end

```

7.18 mainHHseuil.m

```

function out = main( )

a = -85; //Vmin
b = 50; //Vmax
5 nx = 24; //nb subdivisions des abscisses
ny = 24;
hx = (b-a)/(nx-1);

```

```

hy = 1/(ny-1);

10  init_conductance;
    tfinal=Tsim;

    // Resolution du systeme differentiel par methode de Runge-Kutta
    i = 5
15  j = 2
    // Conditions initiales (ici y0 est point fixe)
    y0=[-65,0.3177];

    Iapp=0;

20  [t,y]=ode15s(@(t,y) HHreduit(t,y,Iapp),[0,tfinal],y0);

    // On recupere V, n

25  y1=y(:,1);
    y2=y(:,2);
    max(y1)

    size(t)
30  size(y1)

    t00=t;

    tsimul = Tsim;
35  stimul = 0;
    step = 0.001;
    tn = 0;
    yn(1) = -65;
    yn(2) = 0.3177;
40  vreset = -75;
    nreset = 0.7;
    //d = 8;
    i = 1;
    j = 1;
45  refrac = 3;
    temps(i) = tn;
    volt(i) = yn(1);

50  while (tn < tsimul)

        yn = IFc(yn,tn,step,stimul);
        tn = tn+step;

55  if (yn(2) < seuil(yn(1)))
        yn(1) = vreset;
        yn(2) = nreset;
        spike(j) = tn;
        tn = tn+refrac;

```

```

60         j = j+1
           end;
           i = i+1;
           temps(i) = tn;
           volt(i) = yn(1);
65 end;

spike
j

70 // On affiche les spikes tronques
plot(temps,volt,'r');

hold on;
size(t)
75 size(y1)

// Et la reponse complete
plot(t00,y1);
axis([0 tfinal -80 50]);
80 end

```