Introduction aux méthodes numériques et projet : Modélisation de l'activité neuronale

Profs O. Brüls, Q. Louveaux, F. Nguyen

Février 2018

Introduction

Notre système nerveux est composé de milliards de neurones interconnectés entre eux afin de transmettre l'information d'une partie à l'autre de notre corps ; ils composent également l'essentiel de notre cerveau. Le neurone est une cellule excitable, c'est-à-dire qu'elle répond à un influx nerveux et a la capacité de transmettre ce signal à un neurone voisin par l'intermédiaire de synapses. Depuis toujours, la communauté scientifique a marqué un grand intérêt à comprendre le fonctionnement de notre système nerveux. Depuis quelques années, des scientifiques ont proposé d'étudier le comportement du neurone en en modélisant mathématiquement le fonctionnement. Dans ce projet, nous allons étudier le modèle de Fitzhugh-Nagumo qui modélise la réaction du neurone à une excitation.

Modèle mathématique

Dans ce projet, nous allons utiliser le modèle de l'activation du neurone à travers un système de deux équations différentielles à deux variables.

$$\frac{dV}{dt} = V - \frac{V^3}{3} - n^2 + I_{app} \tag{1}$$

$$\frac{dn}{dt} = \epsilon(n_\infty(V) + n_0 - n). \tag{2}$$

La variable V représente la différence de potentiel de la membrane du neurone alors que la variable n est une variable qui représente la perméabilité de la membrane au potassium, le potassium ayant une action inhibitrice sur V. Dans ce système d'équations différentielles, un certain nombre de concepts doivent encore être explicités. Tout d'abord, le paramètre I_{app} est le courant externe appliqué sur la membrane. C'est ce courant externe qui va déterminer si le neurone va être activé ou pas. ϵ est une constante, dépendant du neurone, que nous fixerons à 0.1. Nous fixons également pour commencer $n_0 = 0$.

Enfin, n_∞ est une fonction dite d'activation de Boltzmann qui peut être calculée de la façon suivante:

$$n_\infty(V) = \frac{2}{1+e^{-5V}}.$$

A noter que le modèle (1)-(2) est formulé pour être adimensionnel à l'exception du temps qui est exprimé en ms. Dans ce projet, nous allons modéliser le neurone en résolvant numériquement le système d'équations différentielles et tenter d'en comprendre son fonctionnement.

Question 1 : méthodes de la sécante et de la bissection

Dans la suite du projet, nous aurons besoin de résoudre des équations non linéaires. Comme cette routine est très importante, la première question du projet consiste à implémenter les deux méthodes principales de recherche de racine qui ont été vues dans le cours, à savoir la méthode de la bissection et la méthode de la sécante. Ces implémentations doivent être les plus génériques possibles et donc pouvoir fonctionner pour tout type de fonction et avec le moins d'hypothèses de départ possibles. On demande donc d'implémenter les deux fonctions

```
function x = secante(@f,x0,x1)
```

et

```
function x = bissection(@f,x0, x1)
```

qui permettent de rechercher la racine d'une fonction MATLAB `f(x)` à partir de deux valeurs initiales `x0` et `x1` selon la méthode de la sécante et de la bissection. N'oubliez pas dans votre implémentation de vérifier les hypothèses des méthodes et d'éventuellement y remédier dans le cas où ces hypothèses ne sont pas rencontrées. Ces deux fonctions seront testées lors du premier milestone.

Question 2: implémentation du système d'équations différentielles

Dans cette question, nous allons modéliser le comportement de l'activité neuronale.

1. Implémenter une fonction `nagumo.m` qui prend en entrée une valeur temps, des valeurs V et n et qui renvoie le membre de droite du système d'équations différentielles (1)-(2). Cette fonction doit pouvoir également prendre en compte le fait que la fonction I_{app} sera variable en fonction du temps dans les questions suivantes.

2. Implémenter la méthode d'Euler explicite pour résoudre le système d'équations différentielles sur l'intervalle de temps [0,50] millisecondes en utilisant $I_{app} = 0.1$ pour la simulation. Discuter du choix du pas d'intégration. Représenter graphiquement l'évolution de V et de n. On utilise une valeur initiale de tension de $V = -1.5$ et $n = 0.5$.

3. Utiliser la fonction `ode45` (ou un autre solveur de matlab) pour résoudre le système d'équations différentielles dans les mêmes conditions qu'à la question précédente.

4. Utiliser à présent l'un des deux solveurs pour résoudre le problème où $I_{app} = 1$. Représenter à nouveau le résultat graphiquement. Effectuer la simulation pendant 200 millisecondes.

Question 3 : Détermination de l'activation nécessaire pour obtenir un spike

Dans cette question, nous allons essayer de comprendre l'impulsion nécessaire pour voir apparaitre le fameux spike du neurone qui encode l'information à transmettre. Nous avons observé dans la question précédente que le neurone produit des spikes à une certaine fréquence lorsque le courant d'impulsion est dans certaines limites. Nous allons essayer de déterminer numériquement quelle est cette limite. Pour cette question, il n'est nécessaire que d'observer la valeur de tension V produite par le neurone.

1. Ecrire une fonction matlab qui prend en entrée une valeur de courant I_{app} à appliquer tout au long de la simulation et qui donne en sortie, soit 0 si le neurone ne produit pas de spikes de manière régulière, soit la valeur de fréquence en Hz des spikes produits par le neurone.

2. Ecrire une fonction matlab qui détermine la valeur de courant I_{app} à appliquer au neurone afin que celui-ci produise des spikes avec une fréquence donnée en entrée. Pour cette question, il est conseillé de se servir des fonctions écrites à la question 1. La fonction que vous avez écrite peut également servir pour déterminer le courant limite à partir duquel le neurone commence à produire des spikes.

Question 4 : Transmission de messages par le neurone

Dans cette question, nous allons voir comment le neurone propage de l'information et en particulier comment il peut transcrire un message analogique en message digital.

1. Le courant d'entrée du neurone n'est, à présent, plus constant. Ecrire un script ou une fonction qui reproduit la réaction du neurone à un courant

d'entrée constant par morceaux. Par exemple, on fournit en entrée un courant comme représenté à la Figure 1.

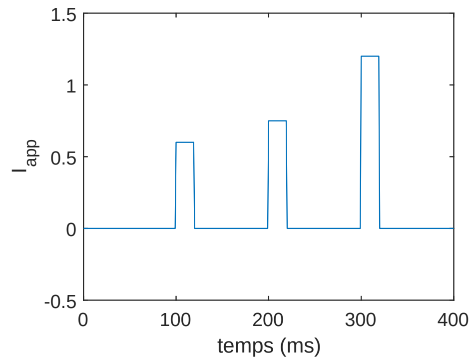

Figure 1: Courant d'entrée constant par morceaux

Le neurone ne produit qu'un léger spike (V en-dessous de 0) pour une valeur insuffisante de courant. Dans ce cas, on dit que le neurone ne produit rien, on peut en effet considérer que c'est imperceptible. Par contre, pour les autres valeurs de courant, le neurone produit un certain nombre de spikes que l'on peut considérer comme presque équivalents en amplitude et fréquence quelle que soit l'amplitude d'entrée au-delà d'un certain seuil.

2. Pour cette question, nous allons considérer le problème inverse, à savoir récupérer les informations produites par le neurone en tentant de récupérer les impulsions envoyées à celui-ci. On suppose donc, pour simplifier, qu'une impulsion de courant de 20 millisecondes est envoyée toutes les 100 millisecondes. Cette impulsion est, soit d'amplitude inférieur au seuil donnant lieu à un spike (ce qui encode un 0), soit supérieure au seuil (ce qui encode un 1). Ecrire une fonction matlab qui reçoit en entrée la tension mesurée sur la membrane du neurone et qui renvoie en sortie un message digital composé de 0 et de 1.

Consignes générales

- Le travail comporte un code de calcul MATLAB et un rapport d'une longueur de 10 pages maximum.
- Le code doit être correct et écrit par vous (ce que nous vérifierons à la présentation orale).
- Le code doit être soigné et commenté.
- Le code doit utiliser au maximum les possibilités vectorielles de MATLAB.

- Pour toute fonction, nous sommes susceptibles de vous demander de montrer un *profile* MATLAB et de l'interpréter.

Critères d'évaluation

La note finale n_f sera définie par la moyenne pondérée

$$n_f = 0.2 \times n_m \; + \; 0.35 \times n_r \; + \; 0.45 \times n_o$$

où n_m est la note de l'évaluation continue (milestones), n_r est la note du rapport et du code MATLAB et n_o est la note de la présentation orale.

Cependant, **une note inférieure ou égale à 9/20 pour l'oral** implique que c'est cette note qui est octroyée à l'étudiant.

Évaluation continue (poids $0,2$)

Deux “milestones” permettront de vérifier votre état d'avancement en cours de projet. Lors du milestone 1 organisé le 2 mars, votre groupe devra effectuer une démonstration du programme développé pour la **Question 1**. Lors du milestone 2 organisé le vendredi 16 mars, votre groupe devra effectuer une démonstration du programme développé pour la **Question 3 (point 1)**. Une cote sur 10 sera attribuée selon les critères suivants :

- le programme donne une réponse complète à la question : 10/10;
- le programme donne une réponse partielle à la question : entre 5 et 9/10;
- le programme est bien avancé mais il ne fonctionne pas : entre 3 et 4/10;
- le programme n'est pas bien avancé : entre 1 et 2/10.

Une absence non justifiée sera sanctionnée par une note individuelle de 0/10, indépendamment du résultat du groupe.

Rapport et code MATLAB (poids $0,35$)

Un fichier .zip par groupe comprenant un rapport au format PDF et accompagné des fichiers .m de votre programme doit être soumis via la plateforme eCampus au plus tard pour le vendredi 6 avril à 23h59. Le nom du fichier .zip et le nom du fichier .pdf doivent respecter le format suivant: “NumeroGroupe_NomA_NomB_NomC.xxx” (exemple : l'archive “27_Dupond_Beckers_Bastin.zip” doit inclure le fichier “27_Dupond_Beckers_Bastin.pdf”).

- La longueur du rapport ne peut dépasser 10 pages et ne doit pas comporter d'introduction.
- Pour chaque question, les résultats obtenus doivent être illustrés.

- La justification des choix numériques est très importante. Pensez à expliquer les choix qui vous ont semblé cruciaux.
- La forme du rapport est prise en compte. Il est recommandé de suivre les règles de bonne pratique pour la réalisation d'un rapport scientifique qui feront l'objet d'une présentation le mercredi 14 mars. Le nombre de pages étant limité, il est inutile de répéter l'énoncé. Allez donc à l'essentiel.
- La qualité du code (efficacité et soin) est également considérée dans l'évaluation.

En plus du rapport par groupe, nous vous demanderons de remplir un questionnaire individuel portant sur votre évaluation du fonctionnement du groupe de travail. Ce questionnaire individuel devra être obligatoirement remis via eCampus pour le dimanche 8 avril.

Présentation orale (poids $0,45$)

La présentation orale est **individuelle**. Vous devez faire une démonstration du programme de votre groupe et répondre à des questions supplémentaires. Les éléments suivants seront pris en considération :

- la maîtrise des éléments de base de matlab,
- la maîtrise du programme réalisé par votre groupe,
- les justifications et éclaircissements par rapport aux choix réalisés et aux résultats présentés dans le rapport,
- la maîtrise des notions théoriques vues au cours.

Deuxièmes sessions

Lors de la session de septembre, certains groupes peuvent être dispensés de remettre un nouveau code et un nouveau rapport. Dans ce cas, seul l'oral doit être représenté et compte pour 100% de la note finale. Pour les autres groupes, un nouveau code et un nouveau rapport doivent être remis 5 jours avant l'examen oral. La note finale est alors définie par la moyenne pondérée

$$n_f = 0.35 \times n_r \, + \, 0.65 \times n_o$$

où n_r est la note du rapport et du code MATLAB et n_o est la note de la présentation orale.

Consignes spécifiques pour les architectes

Les ingénieurs civils architectes suivent les mêmes consignes générales. Cependant, un nombre de consignes spécifiques s'appliquent pour eux.

- Seules les questions complètes 1 et 2 ainsi que la question 4 (point 1) doivent être résolues. La question 3 ainsi que la 4.2 ne doivent donc pas être résolues.
- Un seul milestone est organisé. Il aura lieu le jeudi 8 mars pour les groupes travaillant le jeudi, et le vendredi 9 mars pour les groupes travaillant le vendredi. Le milestone porte sur la question 1 comme pour les ingénieurs civils.
- Comme un seul milestone est organisé, celui-ci a un poids total de 20% de la note finale.