1 Description du problème que l'on souhaite traiter

- Décrire le problème physique ou pratique brièvement.

On souhaite mesurer le potentiel d'action (potentiel électrique) des cellules du myocyte ventriculaire chez l'homme, cellules contractiles composant le muscle cardiaque, à partir d'un modèle minimal fourni en début de projet, sur « Minimal model for human ventricular action potentiels in tissue », de Alfonso Bueno-Orovio, Elizabeth M. Cherry et Flavio H. Fenton.
Ils ont développé un modèle minimal mesurant le potentiel d'action (potentiel électrique) des cellules du myocyte ventriculaire chez l'homme, cellules contractiles composant le muscle cardiaque.

- Donner les équations.

$$\begin{aligned}
\partial_t u &= \nabla(\tilde{D}\nabla u) - (J_{fi} + J_{so} + J_{si}) \\
\partial_t v &= (1 - H(u-\theta_v))(v_\infty - v)/\tau_v^- - H(u-\theta_v)v/\tau_v^+ \\
\partial_t w &= (1 - H(u-\theta_w))(w_\infty - w)/\tau_w^- - H(u-\theta_w)w/\tau_w^+ \\
\partial_t s &= ((1 + \tanh(k_s(u-u_s)))/2 - s)/\tau_s,
\end{aligned}$$

$$\begin{aligned}
J_{fi} &= -vH(u-\theta_v)(u-\theta_v)(u_u - u)/\tau_{fi} \\
J_{so} &= (u-u_o)(1 - H(u-\theta_w))/\tau_o + H(u-\theta_w)/\tau_{so} \\
J_{si} &= -H(u-\theta_w)ws/\tau_{si}
\end{aligned}$$

$$\begin{aligned}
\tau_v^- &= (1 - H(u-\theta_v^-))\tau_{v1}^- + H(u-\theta_v^-)\tau_{v2}^- \\
\tau_w^- &= \tau_{w1}^- + (\tau_{w2}^- - \tau_{w1}^-)(1 + \tanh(k_w^-(u-u_w^-)))/2 \\
\tau_{so} &= \tau_{so1} + (\tau_{so2} - \tau_{so1})(1 + \tanh(k_{so}(u-u_{so})))/2 \\
\tau_s &= (1 - H(u-\theta_w))\tau_{s1} + H(u-\theta_w)\tau_{s2} \\
\tau_o &= (1 - H(u-\theta_o))\tau_{o1} + H(u-\theta_o)\tau_{o2}
\end{aligned}$$

$$v_\infty = \begin{cases} 1, & u < \theta_v^- \\ 0, & u \geq \theta_v^- \end{cases}$$

$$w_\infty = (1 - H(u-\theta_o))(1 - u/\tau_{w\infty}) + H(u-\theta_o)w_\infty^*.$$

- Expliquer ce que l'on souhaite calculer ou réaliser, donner un exemple.

Nous souhaitons mesurer l'intensité électrique une fois le potentiel d'action passé. On est alors apte à comparer l'intensité au début et à la fin du phénomène.

- Décrire brièvement le schéma de discrétisation qui sera utilisé.

Le schéma de discrétisation utilisé est la méthode d'Euler.

2 Programme à réaliser

2.1 Entrées et sorties souhaitées

- Quels sont les données qui seront nécessaires à l'exécution du programme ?

Ici, nous nous sommes servis des données recueillies dans la colonne ENDO. Ces valeurs représentent les caractéristiques des cellules endocardiaques, la couche de cellules musculaires la plus interne du cœur.

- Quels sont les données qui seront récupérées à la fin du programme pour une exploitation scientifique ou technique ?

La donnée récupérée à la fin est l'intensité électrique dans la membrane après le potentiel d'action passé, ce qui nous permettra de mesurer l'amplitude du potentiel électrique.

2.2 Description des algorithmes

D'abord on a commencé par définir toutes les constantes données ainsi que les fonctions caractérisant le système différentiel. Toute les fonctions caractérisant le problème seront regroupé dans une fonctions nommé G qui s'applique à une matrice qui a comme nombre de ligne le nombre de nos variable qui 4 et comme nombre de colonne le nombre de points de discrétisation:

- Après cela on va définir notre opérateur qui est le Laplacien pour cela on a appliqué la méthode de différence finie centrée à trois points. Ce qui nous donne le résultat suivant.

$$\frac{U_{i,j}^{n+1} - U_{i,j}^{n}}{\Delta t} = D\frac{U_{i+1,j}^{n} - 2U_{i,j}^{n} + U_{i-1,j}^{n}}{(\Delta x)^2} + D\frac{U_{i,j+1}^{n} - 2U_{i,j}^{n} + U_{i,j-1}^{n}}{(\Delta y)^2}$$

Pour la constante d on l'a pris égal à 1

Après on écrira cette équation sous forme matricielle pour obtenir pour pouvoir appliqué notre Laplacien à un vecteur y0[0, :]=u(t,x,y) qui est un vecteur à (n+1)^2 composante

- Maintenant qu'on a toute les fonctions caractérisant notre problème et une fonction qui nous calcule notre Laplacien il reste juste à appliquer la méthode d'Euler, , afin de faire tourner le système entier en fonction du temps pour le résoudre.

```
# Solution approchée par la méthode d'Euler
for n in np.arange(0,M+1):
  y1 = y0 + h*bocf.G(y0)
  y1[0,:] = y1[0,:] + h* (A*y0[0,:])
  y0 = y1
```

On récupérera finalement y[0,:] afin de modéliser spatialement son intensité au début et à la fin du programme.

2.3 Principe de l'implémentation

Comme expliqué plus haut, on a eu besoin de deux méthodes de DF. De plus, on a essayé au maximum de définir toutes nos variables sous forme de matrice. Afin d'en faciliter leur traitement, on s'est servi du module Numpy :

- Zéros (x, y) : crée une matrice de taille(x,y) remplie de zéros.
- Arange(a,b,N): stock les éléments de a à b espacés de N éléments

```
In [25]: A = np.arange(3)

In [26]: A
Out[26]: array([0, 1, 2])

In [27]: A = np.arange(2,10,3)

In [28]: A
Out[28]: array([2, 5, 8])

In [29]: A = np.arange(10,17)

In [30]: A
Out[30]: array([10, 11, 12, 13, 14, 15, 16])
```

- mgrid : afin de pouvoir discrétiser l'espace spatial. Exemple :

```
In [32]: np.mgrid[0:3,0:3]
Out[32]:
array([[[0, 0, 0],
        [1, 1, 1],
        [2, 2, 2]],

       [[0, 1, 2],
        [0, 1, 2],
        [0, 1, 2]]])

In [33]: np.mgrid[0:3]
Out[33]: array([0, 1, 2])
```

- Reshape : renvoie un vecteur de longueur x^2 en matrice de taille (x,x)
- Tanh : tangente hyperbolique (utilisé pour définir les équations du problème)
- Stack : joint différents tableaux ensemble pour en fait une matrice :

```
def G(y):
    u = y[0,:]
    v = y[1,:]
    w = y[2,:]
    s = y[3,:]
    return np.stack((I(u,v,w,s), g_v(u,v,w,s), g_w(u,v,w,s), g_s(u,v,w,s)))
```

- En gros, G(y) retourne : [y[0,:] , y[1,:] , y[2,:] , y[3,:]]]

On s'est aussi servi de du module Matplotlib (importé en tant que plt) afin de pouvoir représenter ce qu'on a obtenu, et le comparer à la situation initiale :

- Show : affiche ce qu'on a précédemment défini
- Contourf(X,Y,U,vmin=0,vmax=1) : X et Y sont les discrétisations spatiales des axes correspondants. U est la matrice à mapper sur le domaine spatial en 2 dimensions X x Y.
  Vmin et vmax réadaptent le domaine à l'échelle qu'ils définissent.

Le module Scipy.sparse (sous sp) a été nécessaire afin de pouvoir traiter la matrice creuse du Laplacien de la fonction u :

- spdiags(data,diags,a,b) : on a (a,b) la taille de la matrice creuse obtenue.
  Diags : quelles diagonales sont concernées ? C'est à dire quelles diagonales vont être affectées ?
  Data = Différents vecteurs classés par ordre d'apparence sur les diagonales.

```
In [107]: from scipy.sparse import spdiags as sp

In [108]: x = np.ones(4)

In [109]: diags = np.array([-3,-2,-1,0,1,2,3])

In [110]: data = np.array([-3*x,-2*x,-x,0*x,x,2*x,3*x])

In [111]: sp(data,diags,4,4).toarray()
Out[111]:
array([[ 0.,  1.,  2.,  3.],
       [-1.,  0.,  1.,  2.],
       [-2., -1.,  0.,  1.],
       [-3., -2., -1.,  0.]])
```

III/ ORGANISATION DU TRAVAIL

1/ Organisation collective :

Pendant les rencontres, chaque étudiant donne son point de vue sur la ou les questions à traiter. Par-là, on note une divergence ou une multitude sur certaines questions. Et on s'explique mutuellement afin d'avoir une solution satisfaisante.

2/ Réunion de travail pendant les séances de TD ou en dehors :

D'abord chacun se concentre sur l'objectif de la rencontre en essayant priori de traiter individuellement la ou les questions. Ensuite, on se regroupe pour donner une solution. Enfin, on s'explique ceux qui n'ont pas bien compris.

Le projet est réalisé par 4 étudiants durant tout le semestre.

3/ Outils de travail :

On a utilisé python pour écrire les algorithmes.

En dehors des rencontres, quand quelqu'un a vu préalablement une solution d'une question qu'on ne pouvait pas traiter pendant les rencontres, il envoie cette solution par mail aux autres étudiants.

4/ Difficultés rencontrées :

- Les difficultés d'ordre relationnel dans le groupe : on a des difficultés pour organiser des rencontres en dehors des séances de TP. Et cala ralentit énormément le travail.

- Les difficultés sur l'organisation pratique ???

- Les problèmes liés à l'utilisation d'ordinateurs : certains ont des difficultés pour l'utilisation de certains logiciels.

- Autres difficultés : certains ont rencontrés des problèmes avec python car c'est leur premier usage.

5/Pistes d'amélioration :

* Nous avons commis certaines erreurs comme l'organisation qui pouvait se faire autrement.

* Pour l'éviter, on pouvait séparer les taches pour chaque semaine et organiser une rencontre en dehors des séances de TP pour la mise au point. Nous discutons sur les solutions fournies en essayant de les améliorer ensemble et répartissons avec de nouvelles taches. Et chaque étudiant pouvait être noté par rapport à son travail hebdomadaire.

4 Qu'avez-vous retiré du projet

En réalisant ce projet on a su comment on peut passer d'un problème Physique à un problème purement mathématique.

Grâce au modèle de Feton-Karma on avait un système d'équation différentielle à 4 variables qui dépendaient de l'espace et du temps et à l'aide de méthode numérique on a pu avoir une approximation de la solution de ce système

Pour conclure :

La modélisation mathématique reste un outils indispensable pour résoudre les différents problèmes que ça soit au niveau médicale ou industriel.