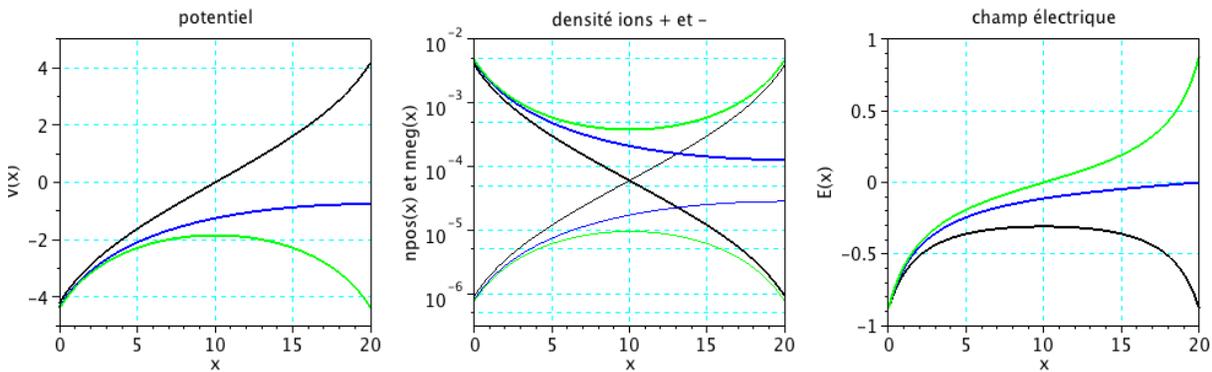


# TD4 - Résolution de l'équation de Poisson Boltzmann entre deux plaques chargées

## 1 Ce qu'on veut faire

Nous nous intéresserons à l'interaction électrostatique entre deux surfaces chargées dans une solution ionique de concentration  $n_0$  (sel monovalent) : ce système est en effet un modèle minimal pour l'interaction entre l'ADN et une protéine. Si on considère des surfaces planes (et infinies), le problème est unidimensionnel. Il s'agira donc de déterminer le potentiel électrostatique  $V(x)$  en fonction de la position  $x$  entre les deux plaques, qu'on considérera comme positionnées en  $x = 0$  et  $x = L$ . Le potentiel  $V(x)$  permet à son tour de déterminer les densités d'ions entre les plaques,  $n_+(x) = n_0 \exp(-\beta eV(x))$  pour les ions positifs et  $n_-(x) = n_0 \exp(-\beta(-e)V(x))$  pour les ions négatifs, et le champ électrique  $E(x) = -V'(x)$ .



## 2 Equation de Poisson Boltzmann

En introduisant le potentiel électrostatique adimensionnel  $\bar{V}(x) = \beta eV(x)$ , pour le cas de deux surfaces parallèles et en sel monovalent l'équation de Poisson entre les plaques s'écrit

$$\frac{d^2\bar{V}(x)}{dx^2} = -4\pi\ell_B(n_+(x) + n_-(x))$$

où  $\ell_B \equiv e^2/4\pi\epsilon k_B T \simeq 7 \text{ \AA}$  est la longueur de Bjerrum.

On peut en déduire alors l'équation de *Poisson-Boltzmann* (PB)

$$\frac{d^2\bar{V}(x)}{dx^2} = -4\pi\ell_B n_0 \left( e^{-\bar{V}(x)} + e^{+\bar{V}(x)} \right).$$

Si les deux plaques ont des charges surfaciques constantes  $\sigma_0$  (en  $x = 0$ ) et  $\sigma_L$  (en  $x = L$ ), les conditions au bord associées imposent la valeur du champ électrique, donc de la dérivée du potentiel, aux deux extrémités de l'intervalle :

$$\begin{aligned} \frac{d\bar{V}}{dx}(0) &= -4\pi\ell_B\sigma_0, \\ \frac{d\bar{V}}{dx}(L) &= 4\pi\ell_B\sigma_L. \end{aligned}$$

Nous allons résoudre numériquement l'équation de Poisson-Boltzmann avec ces conditions aux bord pour déterminer le potentiel (adimensionnel)  $\bar{V}(x)$ .

### 3 Résolution numérique avec `bvodeS`

SCILAB propose une fonction simplifiée pour la résolution d'équations différentielles avec conditions au bord qui s'appelle `bvodeS`. La syntaxe de cette fonction est un peu complexe : nous allons donc partir d'un premier exemple de code pour construire notre programme. Le fichier **exemple-bvodeS.sce** est à votre disposition sur le site de l'UE et est recopié ci-de suite :

```
// DE: y''(x)=-w*w*y(x)-k*y'(x)      equation differentielle
// BV: y(0)=0; y(1)=0                conditions au bord
// V=[y(x);y'(x)]                    vecteur V = [fonction, dérivée]

clf();
clear;

m=2;                                // ordre équa diff
n=1;                                // nombre équa diff

a=0;
b=10;                                // extrémités intervalle de resolution 0,L
zeta=[a b],                          // coordonnées des deux points pour les conditions "au bord"
N=101;                                // nombre de points dans l'intervalle 0,L
x=linspace(a,b,N); // vecteur intervalle 0,L

w=2*pi/3;                            // parametre (pulsation = 2*pi/periode)
k=0.5,                                // paramètre (viscosité)

function rhs=fsub(x,V,w)              // membre de droite de l'équation
    rhs=-w*w*V(1)-k*V(2);
endfunction

function g=gsub(i,V)                 // les conditions aux bords aux deux points dans zeta
    g=[V(1)-1    V(2)-2]             // sous la forme V(1)-A=0 pour la fonction,
                                       // ou V(2)-A=0 pour la dérivée
    g=g(i)                             // voir REM :
endfunction
// REM :
// on doit définir 2 conditions aux bords, aux 2 points donnés dans zeta.
// cette syntaxe permet de construire une fonction gsub qui soit une fonction
// de l'indice du point (1 pour le premier point de zeta, 2 pour le deuxième)
// et de V, où V(1) est la fonction et V(2) sa dérivée.

V=bvodeS(x,m,n,a,b,fsub,gsub,zeta); // integration de l'eq. differentielle

plot2d(x,V(1,:),style=5)              // plot de y(x)
plot2d(x,V(2,:),style=4)              // plot de y'(x)
xlabel('solution', 'x', 'y(x) et y''(x)');
hl=legend(['y(x)';'y''(x)']);
```

Le code précédent intègre l'équation différentielle d'ordre 2 ( $m=2$ ) donnée dans la première

ligne commentée. La grandeur vectorielle  $V$  contient deux éléments :  $V(1)$ , la fonction cherchée (à un  $x$  donné), et  $V(2)$ , sa dérivée. Ces deux grandeurs peuvent être (et sont, dans cet exemple) utilisées pour définir à la fois le second membre de l'équation différentielle et les conditions au bord. La fonction `fsub` définit le terme de droite de l'équation. Les conditions aux bords sont précisées par la fonction `gsub` (noter la syntaxe particulière, commentée dans le code).

On peut remarquer que s'agissant ici d'une équation différentielle de second ordre, elle peut toujours s'écrire sous la forme utilisée ici, à savoir  $y'' = f(y, y')$ .

## 4 Construire le code pour Poisson Boltzmann

1. La première partie de l'exercice de ce TD consiste à comprendre le fonctionnement du code donné dans **exemple-bvodeS.sce**, en le faisant tourner, en jouant sur les termes de l'équation ou les conditions au bord et en vérifiant que le comportement obtenu est bien le comportement attendu. Attention, une équation différentielle peut ne pas avoir de solution (ou avoir plusieurs solutions) pour des conditions au bord données.
2. Modifier ensuite le code pour obtenir la solution de l'équation de Poisson Boltzmann avec les conditions initiales données plus haut. On travaillera avec le potentiel adimensionnel  $\bar{V}(x)$  (et donc à  $k_B T = 1$ ). Voici un choix possible pour les paramètres (distances en Å, densités surfaciques en charges/Å<sup>3</sup> potentiel adimensionnel) :
  - $\ell_B = 7$
  - $\sigma_0 = 0.01$  (c'est la charge surfacique de l'ADN)
  - $\sigma_0 = r\sigma_0$  avec  $r = 1, 0, -1$  pour tester le cas de plaques chargées égales, une neutre, opposées ;
  - $n_0 = 6 \cdot 10^{-5}$  (concentration 0,1 M exprimée en ions/Å<sup>3</sup>)
  - $L = 20$  (ou autre, au choix).
3. Tracer le potentiel  $\bar{V}(x)$ , les concentrations d'ions positifs en négatifs (éventuellement en échelle semilog, utiliser alors de préférence `plot2d` avec l'option `'nl'`), et le champ électrique  $-\bar{V}'(x)$ .

## 5 Pour aller plus loin : la force entre les deux plans

On peut montrer, grâce à une dérivation de physique statistique qu'on ne peut pas transcrire ici, que la force  $f$  qui s'exerce entre les deux plans lorsqu'ils sont séparés d'une distance  $L$  est proportionnelle à

$$f \propto -\frac{1}{8\pi\ell_B} \left( \frac{d\bar{V}}{dx} \right)^2 + n_+ + n_- - 2n_0 = -\frac{1}{8\pi\ell_B} \left( \frac{d\bar{V}}{dx} \right)^2 + n_0 \left( e^{-\bar{V}(x)} + e^{+\bar{V}(x)} - 2 \right),$$

relation valable pour *n'importe quel choix du point  $x$  entre 0 et  $L$* .

Pour calculer comment cette force varie en fonction de la distance entre les plans  $L$  on propose la démarche suivante.

Dans un nouveau fichier, on veut créer une fonction `F=force(L)` qui accepte le paramètre distance entre les plans  $L$  en entrée et restitue une grandeur  $F$  proportionnel à la force en sortie, calculée selon par la relation précédente.

Cette fonction force devra donc contenir tout le calcul fait précédemment pour résoudre l'équation différentielle (à l'exception des plots), tout en fixant la seconde extrémité de l'intervalle à la valeur de  $L$  donnée en entrée (on prendra donc  $b=L$ ).

On pourra donc utiliser cette fonction `force(L)` pour calculer puis tracer la force en fonction de  $L$ , en faisant une boucle sur les valeurs de  $L$ .

Pour s'approcher du système ADN-protéine on choisira pour cette partie les paramètres suivants :

$$\sigma_0 = r\sigma_0 \text{ avec } r = 0.3$$

$L$  variant entre 1 et 40.

**Conclusion :** Qu'on observe-t-on ? Les deux plans se repoussent ou s'attirent, et dans quelles conditions ? Quel est donc le rôle joué par le sel dans ce système ?