

# Compte Rendu d'Analyse Numérique

Adrien A., Nicolas V.

May 11, 2020

## **TP Méthodes Numériques :** *Modélisation du nuage ionique autour d'un polymère chargé*

Nous avons décidé d'implémenter les différents algorithmes en Python. Le code et les figures que vous verrez dans ce document sont présentes dans le répertoire public :

<https://github.com/NicovincX2/debye-huckel-poisson-boltzmann>

Ainsi que des figures additionnelles non présentes dans ce pdf.

*Bonne lecture*

# 1 Introduction

## 1.1 Modèles et contenu du TP

La modélisation du comportement électrique des polymères en solution constitue un problème important, notamment en ce qui concerne les polymères biologiques qui interviennent dans le fonctionnement de nos cellules (mouvement cellulaire, signaux ioniques,...). Dans ce TP, nous allons étudier un modèle simple décrivant le potentiel électrique et la distribution de charges autour d'un polymère possédant des charges de surface, placé dans une solution ionique typique d'un milieu biologique. Un exemple est donné par un filament d'actine représenté dans la figure ???. Ce polymère biologique est chargé négativement en surface et entouré d'un nuage d'ions maintenus dans son voisinage par les interactions électrostatiques.

Un modèle classique décrivant le potentiel électrostatique  $\phi$  autour d'un polymère chargé dans une solution ionique est l'équation de Poisson-Boltzmann (PB) :

$$\frac{d^2\phi}{dr^2} + \frac{1}{r} \frac{d\phi}{dr} = \frac{2en_0}{\epsilon} \operatorname{sh}\left(\frac{e\phi}{k_B T}\right), \quad r > R, \quad (1)$$

avec les conditions aux limites :

$$\phi'(R) = -\frac{\sigma}{\epsilon}, \quad \lim_{r \rightarrow +\infty} \phi(r) = 0. \quad (2)$$

Le polymère est assimilé à un cylindre infini de rayon  $R$  et de densité de charge surfacique  $\sigma$ , autour duquel se trouve un solvant constitué d'ions monovalents (par exemple  $K^+$  et  $Cl^-$ ). La concentration de ces ions à l'infini est notée  $n_0$ ,  $e$  désigne la charge positive élémentaire,  $\epsilon$  la permittivité diélectrique du solvant,  $k_B$  la constante de Boltzmann,  $T$  la température.

A partir de calcul de  $\phi$ , on obtient le champ électrique  $E = -\nabla \phi$  autour du polymère et les concentrations de charges positives et négatives  $n_{\pm}$  dans le nuage ionique, données par la distribution de Boltzmann :

$$n_{\pm} = n_0 \exp\left(\mp \frac{e\phi}{k_B T}\right). \quad (3)$$

En posant  $u(x) = -\frac{e}{k_B T} \phi(Rx)$ , on se ramène à une version adimensionnée du modèle PB :

$$\frac{d^2u}{dx^2} + \frac{1}{x} \frac{du}{dx} = k^2 \operatorname{sh}(u), \quad x > 1, \quad (4)$$

$$u'(1) = -\mu, \quad \lim_{x \rightarrow +\infty} u(x) = 0, \quad (5)$$

où les paramètres sans dimension  $k$  et  $\mu$  sont fonction des constantes et paramètres physiques listés plus haut. On peut montrer que le problème (4)-(5) admet une solution  $u$  unique. Dans ce TP, on fixera  $k = 1$  et on calculera la solution  $u$  pour différentes valeurs de  $\mu$ , en utilisant un schéma aux différences finies et des méthodes itératives non-linéaires (méthode des approximations successives et méthode de Newton). La connaissance de  $u$  et donc de  $\phi$  fournit ensuite la distribution de charges ioniques autour du polymère par la formule (3).

On étudiera également le modèle de Debye-Hückel plus simple :

$$\frac{d^2u}{dx^2} + \frac{1}{x} \frac{du}{dx} = k^2 u, \quad x > 1, \quad (6)$$

qui correspond à l'approximation linéaire  $\text{sh}(u) \approx u$ . Pour cela, on utilisera un schéma aux différences finies et on résoudra le système linéaire correspondant par factorisation  $LU$ .

## 1.2 Schéma aux différences finies

Nous allons décrire un schéma aux différences finies du second ordre permettant d'approcher les solutions de (4)-(5) et de (5)-(6). Dans ce qui suit, on pose  $f(u) = \text{sh}(u)$  dans le cas de l'équation (4) et  $f(u) = u$  pour l'équation (6).

Pour les calculs numériques, on suppose  $x \in [1, 1 + \ell]$  avec  $k^{-1} \ll \ell$ . Les points de discrétisation sont  $x_i = 1 + ih$  avec un pas de discrétisation  $h = \frac{\ell}{n} \ll k^{-1}$  (dans les applications numériques, on fixera  $k = 1$  et  $\ell = 10$ ). On note  $u_i$  une approximation numérique de  $u(x_i)$ .

On remplace la condition aux limites  $\lim_{x \rightarrow +\infty} u(x) = 0$  par  $u(1 + \ell) = 0$ , correspondant à  $u_n = 0$ . Pour approcher la condition aux limites  $u'(1) = -\mu$ , on remarque que la solution vérifie

$$u(1 + h) = u(1) + h u'(1) + \frac{h^2}{2} (k^2 f(u(1)) - u'(1)) + O(h^3)$$

en effectuant un développement de Taylor et en évaluant  $u''(1)$  avec (4) ou (6). La condition  $u'(1) = -\mu$  équivaut donc à

$$-\mu(2 - h) = \frac{2}{h} (u(1 + h) - u(1)) - h k^2 f(u(1)) + O(h^2).$$

En négligeant le reste d'ordre 2, on obtient

$$2(u_1 - u_0) - h^2 k^2 f(u_0) = \mu h (h - 2). \quad (7)$$

Par ailleurs, en approchant les dérivées de l'équation (4) ou (6) par des différences finies centrées, on obtient pour  $i = 1, \dots, n - 1$  :

$$\frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1}))}{h^2} + \frac{1}{x_i} \frac{u(x_{i+1}) - u(x_{i-1}))}{2h} = k^2 f(u(x_i)) + O(h^2).$$

En négligeant le reste d'ordre 2, on obtient

$$u_{i+1} - 2u_i + u_{i-1} + \frac{h}{2x_i} (u_{i+1} - u_{i-1}) - k^2 h^2 f(u_i) = 0, \quad 1 \leq i \leq n - 1. \quad (8)$$

Le schéma aux différences finies pour l'équation (4) ou (6) correspond au système des  $n$  équations (7)-(8) à  $n$  inconnues  $u_0, u_1, \dots, u_{n-1}$  (on rappelle que  $u_n = 0$ ).

## 2 Questions

### 2.1 Factorisation $LU$ des matrices tridiagonales

1. On considère une matrice  $A \in M_n(\mathbb{R})$  tridiagonale admettant une factorisation  $A = LU$ . On note  $a \in \mathbb{R}^n$  la diagonale de  $A$ ,  $b \in \mathbb{R}^{n-1}$  la sous-diagonale et  $c \in \mathbb{R}^{n-1}$  la sur-diagonale. On rappelle que  $L$  et  $U$  sont bidiagonales, et on note  $l \in \mathbb{R}^{n-1}$  la sous-diagonale de  $L$  (de diagonale unité) et  $v \in \mathbb{R}^n$  la diagonale de  $U$  (de sur-diagonale  $c$ ). Rappeler les formules de récurrence qui déterminent

$l$  et  $v$  et écrire une fonction correspondante `[l,v]=lutri(a,b,c)`. Donner un exemple de votre choix pour tester cette fonction.

---

**Algorithm 1** Factorisation LU d'une matrice tridiagonale
 

---

**Entrées:**  $b \in \mathbb{R}^{n-1}$ ,  $a \in \mathbb{R}^n$ ,  $c \in \mathbb{R}^{n-1}$

**Sorties:**  $l \in \mathbb{R}^{n-1}$ ,  $v \in \mathbb{R}^n$

```

 $n \leftarrow \text{length}(a)$ 
 $v_0 \leftarrow a_0$ 
for  $k = 1, \dots, n-1$  do
   $l_{k-1} \leftarrow \frac{b_{k-1}}{v_{k-1}}$ 
   $v_k \leftarrow a_k - l_{k-1}c_{k-1}$ 
end for
```

---

Vérification de notre implémentation avec la matrice

$$A = \begin{pmatrix} 1 & 4 & 0 \\ 2 & 10 & 5 \\ 0 & 6 & 10 \end{pmatrix}$$

qui admet la décomposition  $LU$  :

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 3 & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 1 & 4 & 0 \\ 0 & 2 & 5 \\ 0 & 0 & -5 \end{pmatrix}$$

```

1 b, a, c = [2, 6], [1, 10, 10], [4, 5]
2 l, v = lutri(a, b, c)
3 print(l, v)
4 $ [2.0, 3.0] [1, 2.0, -5.0]
```

On décompose la résolution d'un système tridiagonal  $Ax = z \in \mathbb{R}^n$  en une étape de factorisation  $A = LU$ , suivie de la résolution de  $Ly = z$  (étape de descente) et  $Ux = y$  (étape de remontée).

**2.** Rappeler les formules de récurrence qui déterminent  $y$  et écrire une fonction correspondante `y=descente(l,z)`. Donner un exemple de votre choix pour tester cette fonction.

**Algorithm 2** Algorithme de descente**Entrées:**  $l \in \mathbb{R}^{n-1}$ ,  $z \in \mathbb{R}^n$ **Sorties:**  $y \in \mathbb{R}^n$  $n \leftarrow \text{length}(z)$  $y_0 \leftarrow z_0$ **for**  $k = 1, \dots, n-1$  **do** $y_k \leftarrow z_k - l_{k-1}y_{k-1}$ **end for**

Vérification de notre implémentation avec la matrice  $A$  de la question précédente. On essaie de résoudre le système  $Ax = g = \begin{pmatrix} 3 \\ 3 \\ 3 \end{pmatrix}$ .

La solution du système  $Ly = g$  est  $\begin{pmatrix} 3 \\ -3 \\ 12 \end{pmatrix}$ .

```

1 y = descente(1, [3, 3, 3])
2 print(y)
3 $ [3, -3.0, 12.0]
```

3. Rappeler les formules de récurrence qui déterminent  $x$  et écrire une fonction correspondante `x=remonte(v,c,y)`. Donner un exemple de votre choix pour tester cette fonction.

**Algorithm 3** Algorithme de remontée**Entrées:**  $v \in \mathbb{R}^n$ ,  $c \in \mathbb{R}^{n-1}$ ,  $y \in \mathbb{R}^n$ **Sorties:**  $x \in \mathbb{R}^n$  $n \leftarrow \text{length}(y)$  $x_n \leftarrow \frac{y_n}{v_n}$ **for**  $k = n-1, \dots, 0$  **do** $x_k \leftarrow \frac{y_k - c_{k+1}x_{k+1}}{v_k}$ **end for**

Vérification de notre implémentation en continuation de l'exemple précédent.

La solution du système  $Ax = g$  est  $\begin{pmatrix} -9 \\ 3 \\ -0.6 \end{pmatrix}$ .

```

1 x = remontee(v, c, [3, 3, 3])
2 print(x)
3 $ [-9.0, 3.0, -0.6]
```

## 2.2 Résolution de l'équation de Debye-Hückel

On considère le cas  $f(u) = u$ ,  $k = 1$  et  $h = 10/n$  dans (7)-(8), qui conduit au système linéaire :

$$-(2 + h^2) u_0 + 2 u_1 = \mu h (h - 2), \quad (9)$$

$$\left(1 - \frac{h}{2x_i}\right) u_{i-1} - (2 + h^2) u_i + \left(1 + \frac{h}{2x_i}\right) u_{i+1} = 0, \quad 1 \leq i \leq n-2, \quad (10)$$

$$\left(1 - \frac{h}{2x_{n-1}}\right) u_{n-2} - (2 + h^2) u_{n-1} = 0. \quad (11)$$

On note  $A \in M_n(\mathbb{R})$  la matrice tridiagonale du système linéaire (9)-(10)-(11).

4. Montrer que  $A$  est inversible et admet une factorisation  $A = LU$ .

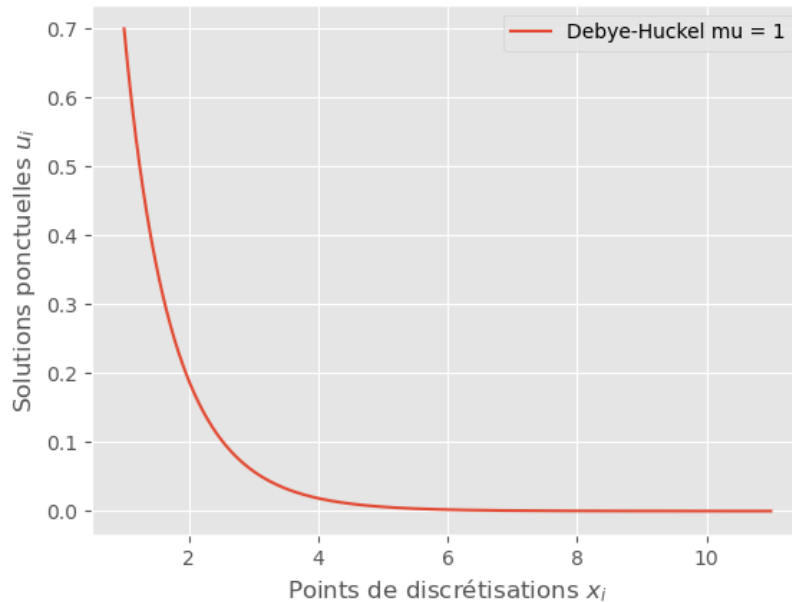
$A$  est inversible car à diagonale strictement dominante. En effet,

$$\begin{aligned} 2 + h^2 &> 2 \\ \forall i \in [1, n-2], \quad 2 + h^2 &> 2 = \left|1 - \frac{h}{2x_i}\right| + \left|1 + \frac{h}{2x_i}\right| \\ 2 + h^2 &> \left|1 - \frac{h}{2x_{n-1}}\right| \end{aligned}$$

La décomposition  $LU$  de  $A$  existe car toutes les sous-matrices principales d'ordre 1 à  $n-1$  sont inversibles (de par les calculs précédents qui montrent qu'elles sont à diagonale dominante).

5. On fixe  $\mu = 1$ . En utilisant les questions 1 à 3, calculer  $(u_0, u_1, \dots, u_{n-1})$  pour  $n = 1000$ , c'est à dire  $h = 10^{-2}$ . Tracer le graphe de  $u_i$  en fonction de  $x_i$ .

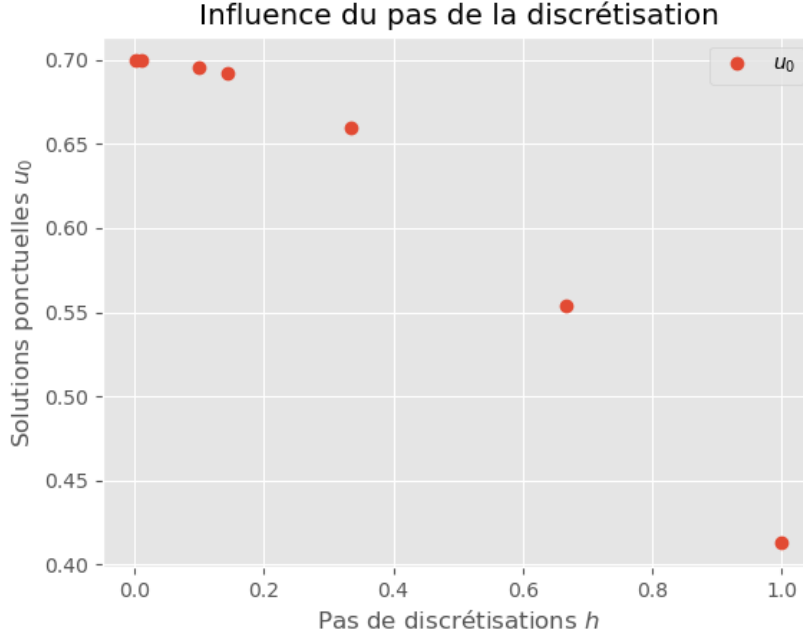
## Schéma aux différences finies pour l'équation de Debye-Hucke



On remarque que les conditions aux limites énoncées plus haut sont respectées.  $\lim_{x \rightarrow +\infty} u(x) = 0$  et  $u'(1) = -\mu$ .

**6.** Toujours pour  $\mu = 1$ , résoudre le système (9)-(10)-(11) pour différentes valeurs de  $n$  et tracer le graphe de  $u_0$  en fonction de  $h$ . Commenter les résultats.

On étudie ici l'influence du pas de discrétisation. On s'attend à observer qu'avec plus de points (ie.  $n$  grand donc  $h$  petit), on obtient une meilleure première approche de  $u_0$ . Cela revient à assurer que la solution discrète converge vers la solution exacte lorsque le pas tend vers 0.



Plus  $h$  est faible plus la première approximation de  $u$ ,  $u_0$ , est correcte (ie. proche de 0.70 pour  $\mu = 1$ ).

### 2.3 Méthode des approximations successives pour l'équation PB

On considère le cas  $f(u) = \text{sh}(u)$ ,  $k = 1$  et  $h = 10/n$  dans (7)-(8), qui conduit au système non linéaire :

$$-(2 + h^2)u_0 + 2u_1 = h^2 g(u_0) + \mu h(h - 2), \quad (12)$$

$$\left(1 - \frac{h}{2x_i}\right)u_{i-1} - (2 + h^2)u_i + \left(1 + \frac{h}{2x_i}\right)u_{i+1} = h^2 g(u_i), \quad (13)$$

$$1 \leq i \leq n - 2,$$

$$\left(1 - \frac{h}{2x_{n-1}}\right)u_{n-2} - (2 + h^2)u_{n-1} = h^2 g(u_{n-1}), \quad (14)$$

où on note  $g(u) = \text{sh}(u) - u$ .

Pour résoudre le système (12)-(13)-(14), on calcule la suite d'approximations successives  $u^{(k)} = (u_0^{(k)}, u_1^{(k)}, \dots, u_{n-1}^{(k)})$  définies par récurrence :

$$-(2 + h^2)u_0^{(k+1)} + 2u_1^{(k+1)} = h^2 g(u_0^{(k)}) + \mu h(h - 2), \quad (15)$$

$$\left(1 - \frac{h}{2x_i}\right)u_{i-1}^{(k+1)} - (2 + h^2)u_i^{(k+1)} + \left(1 + \frac{h}{2x_i}\right)u_{i+1}^{(k+1)} = h^2 g(u_i^{(k)}), \quad (16)$$

$$1 \leq i \leq n - 2,$$

$$\left(1 - \frac{h}{2x_{n-1}}\right)u_{n-2}^{(k+1)} - (2 + h^2)u_{n-1}^{(k+1)} = h^2 g(u_{n-1}^{(k)}), \quad (17)$$



avec  $u^{(0)} \in \mathbb{R}^n$  solution du système (9)-(10)-(11).

Le schéma itératif (15)-(16)-(17) s'écrit  $A u^{(k+1)} = G(u^{(k)})$ , où  $G : \mathbb{R}^n \rightarrow \mathbb{R}^n$  est l'application définie par le membre de droite de (15)-(16)-(17).

Pour les applications numériques, on fixera  $n = 1000$ , c'est à dire  $h = 10^{-2}$ .

7. Ecrire une fonction qui calcule  $u^{(k+1)}$  en fonction de  $u^{(k)}$  en utilisant les questions 1 à 3.

---

**Algorithm 4** Algorithme de calcul de  $u^{(k+1)}$

---

**Entrées:**  $u^{(k)} \in \mathbb{R}^n$ ,  $n \in \mathbb{N}$ ,  $\mu \in \mathbb{R}$

**Sorties:**  $u^{(k+1)} \in \mathbb{R}^n$ ,  $z \in \mathbb{R}^n$

```

 $h \leftarrow \frac{10}{n}$ 
 $a_0 \leftarrow -2 - h^2$ 
 $c_0 \leftarrow 2$ 
 $z_0 \leftarrow h^2(\sinh(u_0^{(k)}) - u_0^{(k)}) + \mu h(h - 2)$ 
 $x_0 \leftarrow 1$ 
for  $i = 1, \dots, n - 2$  do
     $a_i \leftarrow -2 - h^2$ 
     $x_i \leftarrow 1 + ih$ 
     $b_{i-1} \leftarrow 1 - \frac{h}{2x_i}$ 
     $c_i \leftarrow 1 + \frac{h}{2x_i}$ 
     $z_i \leftarrow h^2(\sinh(u_i^{(k)}) - u_i^{(k)})$ 
end for
 $a_{n-1} \leftarrow -2 - h^2$ 
 $x_{n-1} \leftarrow 1 + (n - 1)h$ 
 $b_{n-2} \leftarrow 1 - \frac{h}{2x_{n-1}}$ 
 $z_{n-1} \leftarrow h^2(\sinh(u_{n-1}^{(k)}) - u_{n-1}^{(k)})$ 

 $l, v \leftarrow \text{lutri}(a, b, c)$ 
 $y \leftarrow \text{descente}(l, z)$ 
 $u^{(k+1)} \leftarrow \text{remontee}(v, c, y)$ 

```

---

8. Ecrire un programme qui calcule les termes  $u^{(k)}$  pour  $k = 0, 1, \dots, k_0$ , jusqu'à ce qu'on ait soit  $k_0 > 200$  (nombre d'itérations trop grand, signe de non-convergence), soit (tolérance d'erreur atteinte, signe de convergence) :

$$\|A u^{(k_0)} - G(u^{(k_0)})\|_\infty < \eta_1 \text{ et } \|u^{(k_0)} - u^{(k_0-1)}\|_\infty < \eta_2$$

avec  $\eta_1 = 10^{-12}$ ,  $\eta_2 = 10^{-9}$ . Dans ce dernier cas,  $u^{(k_0)}$  est considéré comme la solution numérique de (12)-(13)-(14). Calculer cette solution pour  $\mu = 1$  et  $\mu = 4$ . On tracera dans chaque cas le graphe de  $u_i$  en fonction de  $x_i$  et on indiquera la valeur de  $k_0$ . Sur les mêmes graphes, comparer la solution de (12)-(13)-(14) à celle du modèle linéaire (9)-(10)-(11). Conclusion ?

On suppose qu'on dispose de  $u^{(0)}$ , solution de l'équation de Debye-Hückel.

---

**Algorithm 5** Algorithme de calcul de  $k_0$

---

**Entrées:**  $n \in \mathbb{N}$ ,  $\mu \in \mathbb{R}$

**Sorties:**  $k_0 \in \mathbb{N}$

$\eta_1, \eta_2 \leftarrow 10^{-12}, 10^{-9}$

$k \leftarrow 0$

$x \leftarrow Au^{(k)}$

$y \leftarrow u^{(k)}$

**while**  $k \leq 200$  et  $(\|x\|_\infty \geq \eta_1$  ou  $\|y\|_\infty \geq \eta_2)$  **do**

$u^{(k+1)}, z \leftarrow \text{calculer\_}u^{(k+1)}(u^k, n, \mu)$

$x \leftarrow Au^{(k+1)} - z$

$y \leftarrow u^{(k+1)} - u^{(k)}$

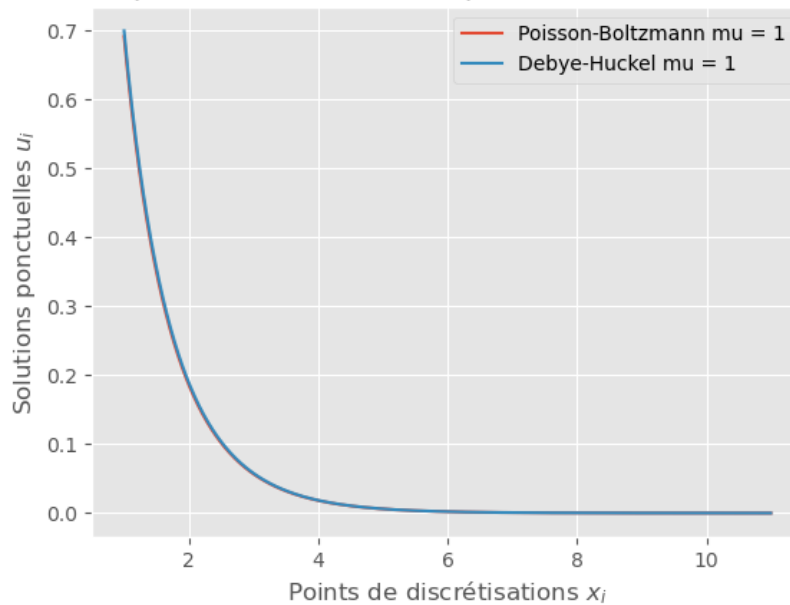
$k \leftarrow k + 1$

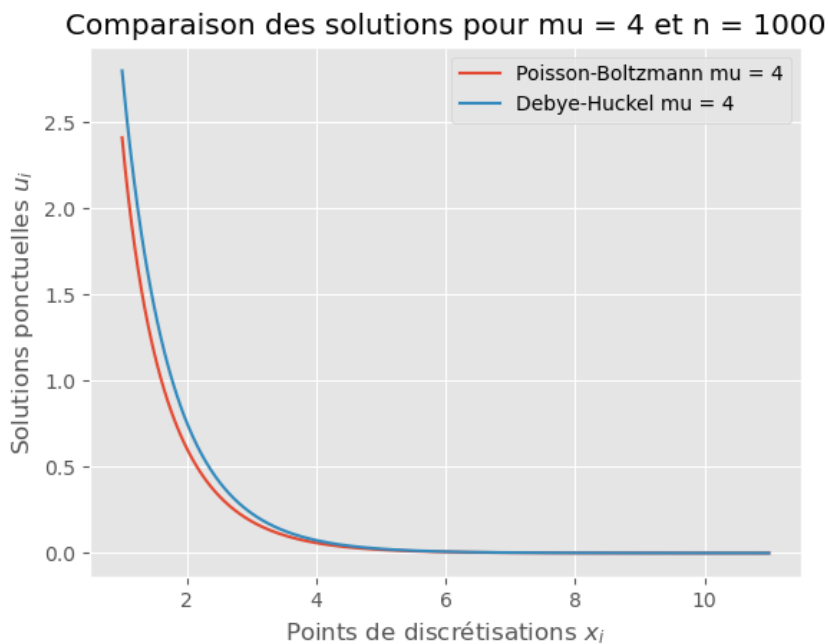
**end while**

$k_0 \leftarrow k$

---

Comparaison des solutions pour  $\mu = 1$  et  $n = 1000$





Pour  $\mu = 1$ , les solutions des deux modèles sont identiques alors que pour  $\mu = 4$ , la solution obtenue par le modèle Poisson-Boltzmann s'écarte de celle obtenue par le modèle de Debye-Huckel.

Cet écart est peut-être dû à la différence de consistance ou robustesse entre les schémas, ie. l'écart induit localement par les approximations inhérentes à l'opérateur discrétisé.

Le modèle de l'équation de Poisson-Boltzmann avec la méthode des différences finies n'est pas adapté pour des valeurs de  $\mu$  proches de  $\mu_{limite}$ , calculé question suivante.

9. Etudier numériquement si la méthode itérative converge ou diverge lorsque  $\mu$  varie dans l'intervalle  $[0, 7]$ .

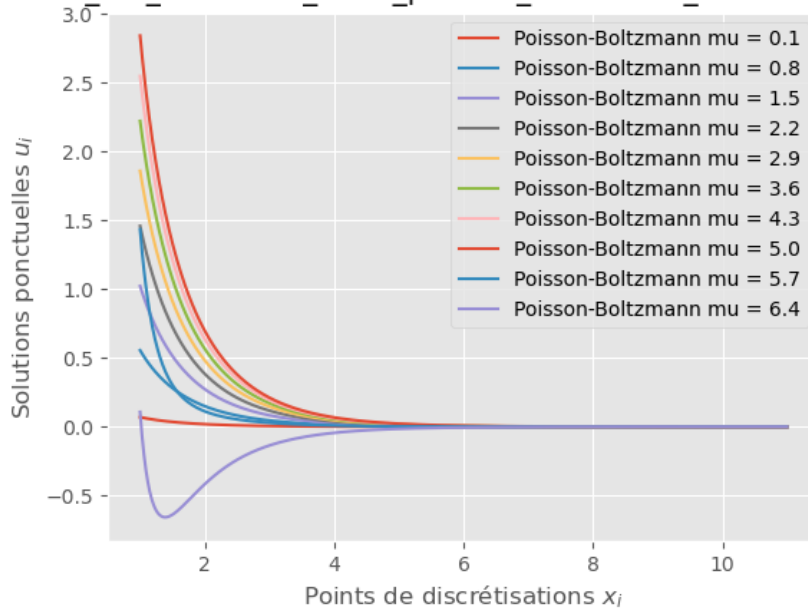
La convergence d'un schéma numérique assurant que l'écart (au sens ici de la norme infinie de convergence uniforme dans  $L^\infty$ ) entre la solution approchée et la solution exacte tend vers 0 lorsque le pas de discrétisation tend vers 0.

Nous sommes ici dans le cas d'un schéma explicite. Il faut tenir compte de la propagation des écarts qui, au cours des étapes du calcul, se combinent aux écarts précédents et peuvent progressivement s'amplifier. C'est la notion de stabilité du schéma.

On nous indique qu'il y a divergence si  $k_0 > 200$ . On réalise donc une dichotomie pour trouver la valeur limite de  $\mu$  à  $10^{-6}$  près pour  $\mu \in [0, 7]$ . Cela revient à vérifier si les valeurs des normes calculées tendent vers 0 lorsque  $h$  tend vers 0 (NB. : cette dernière remarque pourrait mériter un graphe).

On obtient  $\mu = 4.828635215759277$ . Au delà de cette valeur, le modèle numérique diverge.

poisson\_mu\_variations\_solve\_poisson\_boltzmann\_differences\_fin



Visuellement, on observe que la tendance initiale s'inverse pour les courbes obtenues avec  $\mu = 5.7$  et  $\mu = 6.4$ . En effet initialement, plus  $\mu$  augmente, plus  $u_0$  augmente. Et cela jusqu'à ce que l'on approche de  $\mu_{limite}$ .

## 2.4 Méthode de Newton pour l'équation PB

On considère le cas  $f(u) = \text{sh}(u)$ ,  $k = 1$  et  $h = 10/n$  dans (7)-(8). Le système non linéaire s'écrit sous la forme :

$$F_0(u_0, u_1) = 0, \quad (18)$$

$$F_i(u_{i-1}, u_i, u_{i+1}) = 0, \quad 1 \leq i \leq n-2, \quad (19)$$

$$F_{n-1}(u_{n-2}, u_{n-1}) = 0, \quad (20)$$

où les fonctions  $F_i$  sont définies par :

$$F_0(u_0, u_1) = -2u_0 - h^2 \text{sh}(u_0) + 2u_1 + \mu h(2-h), \quad (21)$$

$$F_i(u_{i-1}, u_i, u_{i+1}) = \left(1 - \frac{h}{2x_i}\right) u_{i-1} - 2u_i - h^2 \text{sh}(u_i) + \left(1 + \frac{h}{2x_i}\right) u_{i+1}, \quad (22)$$

$$1 \leq i \leq n-2,$$

$$F_{n-1}(u_{n-2}, u_{n-1}) = \left(1 - \frac{h}{2x_{n-1}}\right) u_{n-2} - 2u_{n-1} - h^2 \text{sh}(u_{n-1}). \quad (23)$$

Pour résoudre le système (18)-(19)-(20) par la méthode de Newton, on calcule une suite  $u^{(k)} = (u_0^{(k)}, u_1^{(k)}, \dots, u_{n-1}^{(k)})^T$  ( $(\cdot)^T$  désigne la transposition) définie par récurrence de la manière suivante :

$$J_k(u^{(k+1)} - u^{(k)}) = -F(u^{(k)}), \quad (24)$$

avec  $u^{(0)} \in \mathbb{R}^n$  solution du système (9)-(10)-(11). La fonction  $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$  est définie par  $F = (F_0, F_1, \dots, F_{n-1})^T$  et  $J_k = DF(u^{(k)}) \in M_n(\mathbb{R})$  désigne la matrice jacobienne de  $F$  (dans le cas présent tridiagonale), dont le coefficient  $(i, j)$  est  $\frac{\partial F_i}{\partial u_j}$  pour  $0 \leq i, j \leq n-1$ .

Pour les applications numériques, on fixera  $n = 1000$ , c'est à dire  $h = 10^{-2}$ .

**10.** Expliciter la matrice  $J_k$  en fonction de  $u^{(k)}$  et montrer que cette matrice est inversible.

La matrice  $J_k$  est tridiagonale.

$$J_k = \begin{bmatrix} -2 - h^2 ch(u_0^{(k)}) & 2 & 0 & \dots & 0 \\ 1 - \frac{h}{2x_1} & -2 - h^2 ch(u_1^{(k)}) & 1 + \frac{h}{2x_1} & 0 & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & 0 & 1 - \frac{h}{2x_{n-2}} & -2 - h^2 ch(u_{n-2}^{(k)}) & 1 + \frac{h}{2x_{n-2}} \\ 0 & \dots & 0 & 1 - \frac{h}{2x_{n-1}} & -2 - h^2 ch(u_{n-1}^{(k)}) \end{bmatrix}$$

$J_k$  est inversible car à diagonale dominante. En effet,

$$2 + h^2 ch(u_0^{(k)}) > 2$$

$$\forall i \in [1, n-2], \quad 2 + h^2 ch(u_i^{(k)}) > 2 = \left|1 - \frac{h}{2x_i}\right| + \left|1 + \frac{h}{2x_i}\right|$$

$$2 + h^2 ch(u_{n-1}^{(k)}) > \left|1 - \frac{h}{2x_{n-1}}\right|$$

**11.** Ecrire un programme qui calcule les termes  $u^{(k)}$  pour  $k = 0, 1, \dots, k_0$ , jusqu'à ce qu'on ait soit  $k_0 > 50$  (nombre d'itérations trop grand, signe de non-convergence), soit (tolérance d'erreur atteinte, signe de convergence) :

$$\|F(u^{(k_0)})\|_\infty < \eta_1 \text{ et } \|u^{(k_0)} - u^{(k_0-1)}\|_\infty < \eta_2$$

avec  $\eta_1 = 10^{-12}$ ,  $\eta_2 = 10^{-9}$ . Dans ce dernier cas,  $u^{(k_0)}$  est considéré comme la solution numérique de (18)-(19)-(20). Calculer cette solution pour  $\mu = 1$  et  $\mu = 4$ . On tracera dans chaque cas le graphe de  $u_i$  en fonction de  $x_i$  et on indiquera la valeur de  $k_0$ . Comparer les solutions obtenues par la méthode de Newton à celles obtenues précédemment par la méthode des approximations successives (question 8). Conclusion ?

On appelle  $J_k^{-1}$  l'inverse de la matrice  $J_k$  tridiagonale de diagonales  $b$ ,  $a$  et  $c$  (sous-diagonale, diagonale et sur-diagonale).

---

**Algorithm 6** Algorithme pour calculer  $u^{(k+1)}$

---

**Entrées:**  $u^{(k)} \in \mathbb{R}^n$ ,  $F(u^{(k)}) \in \mathbb{R}^n$ ,  $n \in \mathbb{N}$ ,  $\mu \in \mathbb{R}$

**Sorties:**  $u^{(k+1)} \in \mathbb{R}^n$ ,  $F(u^{(k+1)}) \in \mathbb{R}^n$

$h \leftarrow \frac{10}{n}$

$a_0 \leftarrow -2 - h^2 \cosh(u_0^{(k)})$

$c_0 \leftarrow 2$

$x_0 \leftarrow 1$

**for**  $i = 1, \dots, n-2$  **do**

$a_i \leftarrow -2 - h^2 \cosh(u_i^{(k)})$

$x_i \leftarrow 1 + ih$

$b_{i-1} \leftarrow 1 - \frac{h}{2x_i}$

$c_i \leftarrow 1 + \frac{h}{2x_i}$

**end for**

$a_{n-1} \leftarrow -2 - h^2 \cosh(u_{n-1}^{(k)})$

$x_{n-1} \leftarrow 1 + (n-1)h$

$b_{n-2} \leftarrow 1 - \frac{h}{2x_{n-1}}$

$u^{(k+1)} \leftarrow u^{(k)} - J_k^{-1} F(u^{(k)})$

$F_0(u^{(k+1)}) \leftarrow -2u_0^{(k+1)} - h^2 \sinh(u_0^{(k+1)}) + 2u_1^{(k+1)} + \mu h(2-h)$

**for**  $i = 1, \dots, n-2$  **do**

$F_i(u^{(k+1)}) \leftarrow b_i u_{i-1}^{(k+1)} - 2u_i^{(k+1)} - h^2 \sinh(u_i^{(k+1)}) + c_i u_{i+1}^{(k+1)}$

**end for**

$F_{n-1}(u^{(k+1)}) \leftarrow b_n u_{n-2}^{(k+1)} - 2u_{n-1}^{(k+1)} - h^2 \sinh(u_{n-1}^{(k+1)})$

---

On suppose qu'on dispose de  $u^{(0)}$ , solution de l'équation de Debye-Hückel.

---

**Algorithm 7** Algorithme pour calculer  $k_0$

---

**Entrées:**  $n \in \mathbb{N}$ ,  $\mu \in \mathbb{R}$

**Sorties:**  $k_0 \in \mathbb{N}$

$\eta_1, \eta_2 \leftarrow 10^{-12}, 10^{-9}$

$k \leftarrow 0$

$x \leftarrow Au^{(k)}$

$y \leftarrow u^{(k)}$

$F_0(u^{(k)}) \leftarrow -2u_0^{(k)} - h^2 \sinh(u_0^{(k)}) + 2u_1^{(k)} + \mu h(2-h)$

**for**  $i = 1, \dots, n-2$  **do**

$F_i(u^{(k)}) \leftarrow b_i u_{i-1}^{(k)} - 2u_i^{(k)} - h^2 \sinh(u_i^{(k)}) + c_i u_{i+1}^{(k)}$

**end for**

$F_{n-1}(u^{(k)}) \leftarrow b_n u_{n-2}^{(k)} - 2u_{n-1}^{(k)} - h^2 \sinh(u_{n-1}^{(k)})$

**while**  $k \leq 50$  et  $(\|x\|_\infty \geq \eta_1$  ou  $\|y\|_\infty \geq \eta_2)$  **do**

$u^{(k+1)}, F(u^{(k+1)}) \leftarrow \text{calculer\_}u^{(k+1)}(u^{(k)}, F(u^{(k)}), n, \mu)$

$x \leftarrow F(u^{(k+1)})$

$y \leftarrow u^{(k+1)} - u^{(k)}$

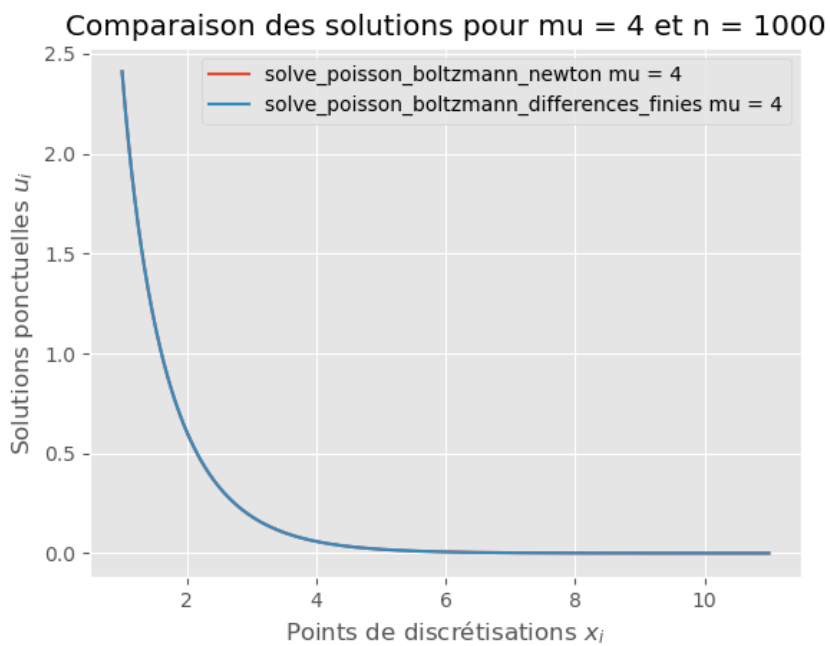
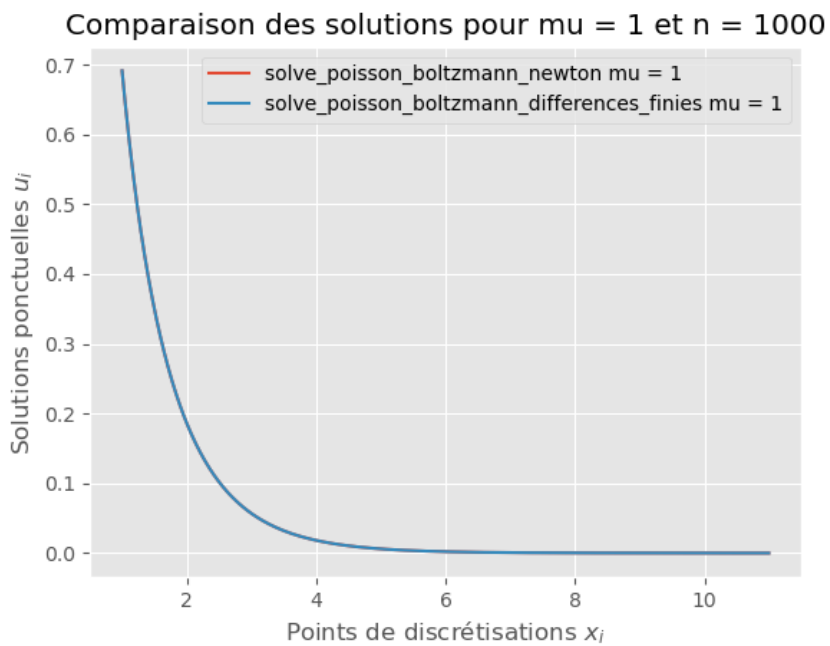
$F(u^{(k)}) \leftarrow F(u^{(k+1)})$

$k \leftarrow k + 1$

**end while**

$k_0 \leftarrow k$

---



Pour  $\mu = 1$  et  $\mu = 4$ , les solutions des deux modèles sont identiques.

On peut utiliser au choix l'un ou l'autre modèle numérique pour  $\mu$  dans l'intervalle  $[0, 7]$ .

12. Etudier numériquement si la méthode de Newton converge ou diverge

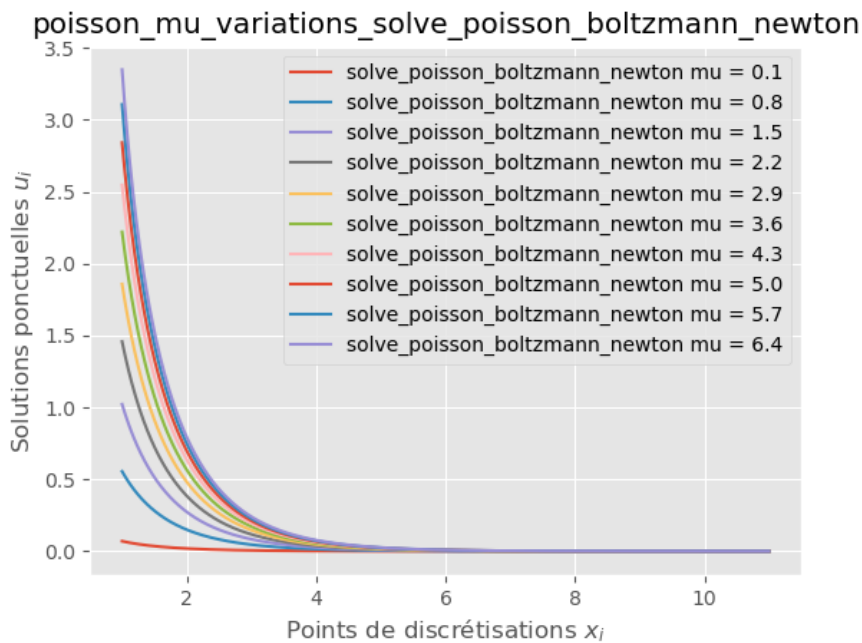


lorsque  $\mu$  varie dans l'intervalle  $[0, 7]$ .

On nous indique qu'il y a divergence si  $k_0 > 50$ . On réalise donc une dichotomie pour trouver la valeur limite de  $\mu$  à  $10^{-6}$  près pour  $\mu \in [0, 7]$ .

On obtient  $\mu = 6.998291015625$  très proche de 7. On a donc convergence sur l'intervalle  $[0, 7]$ .

On peut rechercher la valeur de  $\mu_{limite}$  hors de cet intervalle. On obtient  $\mu_{limite} = 77.13208198547363$ , valeur à partir de laquelle notre modèle diverge. Cette méthode suppose bien sûr que la valeur donnée de  $k_0$  est correcte.



Visuellement, n'observe pas le phénomène divergent observé à la question 9.