

Projet Méthodes Numériques

Résolution du courant dans une diode p-n par la méthode des différences finies

AMDDAH Mohamed

IQBALI Omar

FLISS Ayman

Ingénieure Physique pour la photonique et le microélectronique

Grenoble-INP PHELMA

30th March 2025

Contents

1	Introduction	3
2	Relations physiques	3
2.1	Potentiel et concentrations de porteurs:	3
2.2	Densités de courants:	4
2.3	Barrière de potentiel V_{bi}	4
3	Structure du code	4
4	Première implémentation de la boucle:	4
4.1	Initialisation de $\rho(x)$	4
4.2	Discretisation de l'équation de Poisson	4
4.3	Calcul de $n(x)$ et $p(x)$	6
4.4	Résultats	6
5	Deuxième Implémentation de Poisson avec un schéma Newton-Raphson	6
5.1	Poisson Amortie	6
5.2	Discretisation de Poisson amortie	6
5.3	Résultats	7
6	Résolution de courants dans la Jonction PN	8
6.1	Calcul de $n(x)$ et $p(x)$	10
6.2	Résolution de Poisson amortie sous polarisation	10
6.3	Détermination du courant	11
6.4	Résultats de la caractéristique	12
7	Conclusion	13

1 Introduction

Dans ce projet on s'est intéressé à simuler le courant dans une jonction PN en fonction de sa polarisation. Pour se faire on ne considère pas les hypothèses vu en cours et on ne considère que les relations entre le potentiel et la concentrations des porteurs de charges. La jonction est alors vu comme un système auto-consistant dans lequel le potentiel $V(x)$ affecte les concentrations $n(x)$ et $p(x)$ et par conséquent $\rho(x)$, et vice-versa. Ce qui sera implémenté par une boucle de calcul jusqu'à la convergence.

Ces équations seront ensuite résolues en utilisant la méthode des éléments finis.

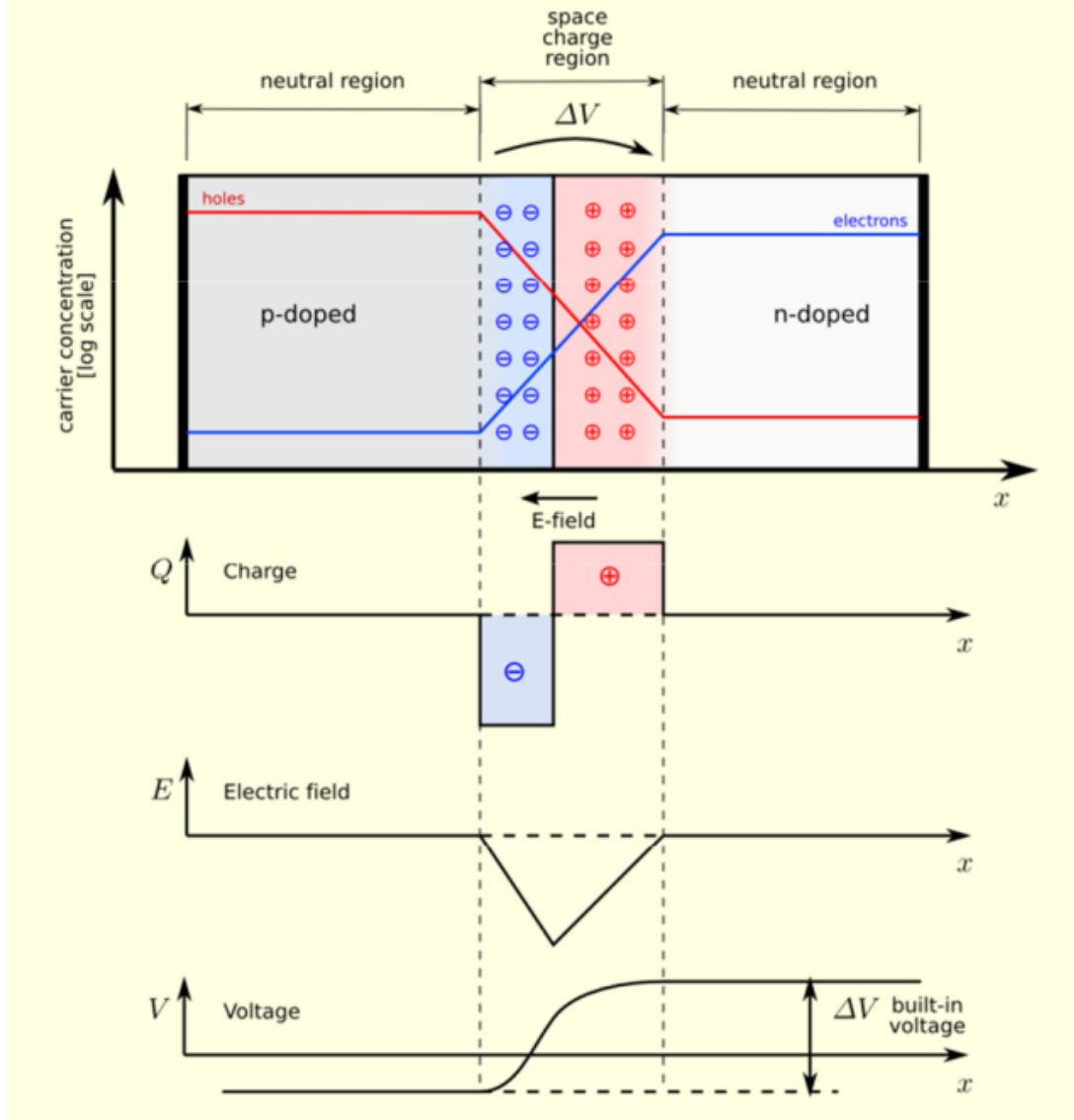


Figure 1: Modèle simplifiée vu en cours, considérant des distribution de charges constantes par morceaux.

2 Relations physiques

On donne ici les relations de la physique des semi-conducteurs qui seront utilisées dans la suite.

2.1 Potentiel et concentrations de porteurs:

D'autre part, le profil de charge est lié au potentiel par les relations :

$$\begin{cases} n(x) = n_0 \exp\left(\frac{qV(x)}{k_b T}\right) \\ p(x) = p_0 \exp\left(\frac{-qV(x)}{k_b T}\right) \end{cases} \quad (1)$$

On détermine la densité de charge dans un semi-conducteur à l'aide de l'équation

$$\rho(x) = p(x) + n(x) + N_D - N_A \quad (2)$$

N_D et N_A sont les densités des ions donneurs et accepteurs respectivement.

On donne finalement une équation du potentiel $V(x)$ en fonction des densités par l'équation de Poisson :

$$\Delta V = -\frac{\rho(x)}{\epsilon_s} \quad (3)$$

2.2 Densités de courants:

Les courants dérive-diffusion dans la jonction PN sont données par les relations:

$$\begin{cases} \vec{j}_n = q\mu_n(-n * \vec{\nabla}V(x) + \frac{k_bT}{q} * \vec{\nabla}n(x)) \\ \vec{j}_p = q\mu_p(-p * \vec{\nabla}V(x) + \frac{k_bT}{q} * \vec{\nabla}p(x)) \end{cases} \quad (4)$$

Dans le cadre de ce projet, on considère le courant dans un état stationnaire, ce qui réduit l'équation de continuité dans:

$$\text{div}(\vec{j}_n + \vec{j}_p) = 0 \quad (5)$$

2.3 Barrière de potentiel V_{bi}

On donne aussi la barrière de potentiel V_{bi} , qui apparaît dans la jonction PN suite a la diffusion dans la jonction après la mise en contact. On la calcule par :

$$V_{bi} = \frac{k_bT}{q} * \ln\left(\frac{N_D N_A}{n_i^2}\right) \quad (6)$$

Ici N_D et N_A sont les densités des impuretés introduites dans les cotes N et P de la jonction respectivement.

3 Structure du code

L'implémentation numérique de la simulation de la jonction pn est réalisée entièrement en MATLAB. Pour assurer une organisation claire et modulaire, le code est structuré en trois types de scripts distincts :

- **Scripts d'entrée et de configuration** : Ces fichiers contiennent les valeurs nécessaires aux simulations. Ils permettent d'ajuster facilement les simulations sans modifier directement les algorithmes sous-jacents.
- **Scripts de définition de fonctions** : Ces fichiers implémentent les différentes fonctions utilisées dans la résolution numérique, telles que le solveur de l'équation de Poisson, le calcul des densités de porteurs, etc.
- **Scripts d'exécution** : Ces scripts utilisent les deux précédents pour exécuter la simulation en fonction des résultats recherchés et de l'approche numérique adoptée.

Cette structure modulaire facilite la gestion du code, la réutilisation des fonctions et l'expérimentation avec différentes configurations sans altérer l'intégrité des algorithmes principaux.

4 Première implémentation de la boucle:

Dans un premier temps on implémente l'algorithme représenté dans la Figure 2, avec l'équation de Poisson habituelle dans (3). Ce code est implémenté dans le script `first_implementation.m`.

4.1 Initialisation de $\rho(x)$

On initialise la simulation avec une distribution similaire a celle présentée en Figure 1. Cette initialisation est implémente par la fonction `charge_initialisation.m` avec le syntaxe `rho=charge_initialisation(X,Na,Nd,Wp,Wn)` et donne une liste construite suivant le maillage X.

4.2 Discrétisation de l'équation de Poisson

Dans cet algorithme la résolution de l'équation de Poisson se fait par une discrétisation de l'équation de Poisson (3) sur un maillage uniforme avec un nombre de nœuds $N=100$, et un pas $dx = \frac{W_{tot}}{N}$, avec W_{tot} la longueur de la jonction, ce qui va nous permettre d'utiliser la méthodes des différences finies.

Cette discrétisation donne le système d'équations :

$$\frac{V_{i+1} - 2V_i + V_{i-1}}{dx^2} = -\frac{\rho_i}{\epsilon_s}, \quad \forall i \in \llbracket 2, N-1 \rrbracket \quad (7)$$

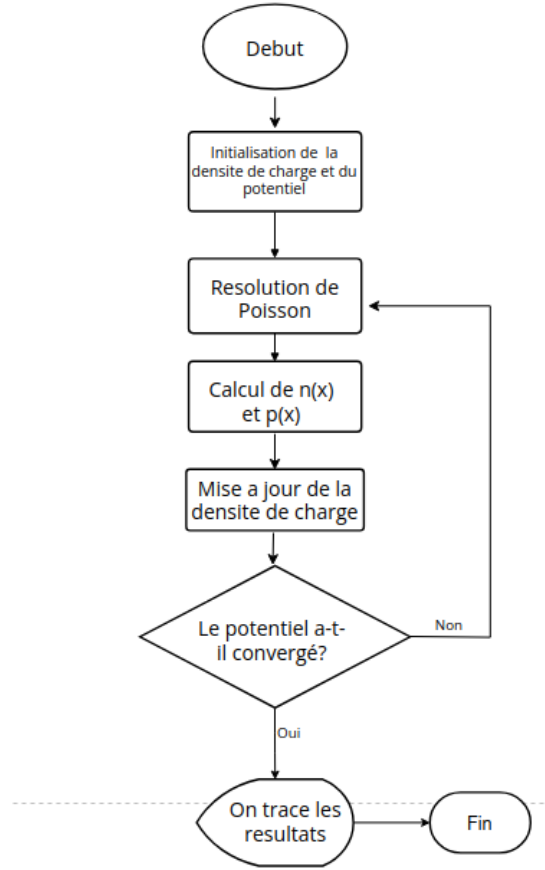


Figure 2: Algorithme pour la première implémentation de la boucle de détermination de $V(x)$, $\rho(x)$, $n(x)$ et $p(x)$.

Pour distinguer le cote P ($x < 0$) et le cote N ($x > 0$) on crée tout les listes suivant la liste du maillage X , e.g $V = \text{zeros}(\text{size}(X))$. Cette initialisation va nous permettre de faire une indexation logique des listes par x au lieu de i , e.g $\text{rho}(x < 0)$ pour définir ρ dans le cote P.

En outre, on considère des nœuds $i = 0$ et $i = N + 1$ en dehors de la jonction, qui seront normalement dans le contact métallique, ces nœuds seront utilisée pour imposer des conditions de limites de Dirichlet, a gauche (nœud $i=0$) et a droite (nœud $i=N+1$). Dans ce qui suit ces conditions seront notées $V_g = V_0$) et $V_d = V_{N+1}$. Ces nœuds

La résolution de ce système alors revient a trouver le vecteur V tel que

$$\Delta V = Y$$

tel que $\Delta_{i,j}$ vaut

$$\begin{cases} \frac{-2}{dx^2} & \text{si } i = j \\ \frac{1}{dx^2} & \text{si } i \pm 1 = j \\ 0 & \text{sinon.} \end{cases}$$

et pour Y_i

$$\begin{cases} Y_1 = -\frac{\rho_1}{\epsilon_s} - \frac{V_g}{dx^2} \\ Y_N = -\frac{\rho_N}{\epsilon_s} - \frac{V_d}{dx^2} \\ Y_i = -\frac{\rho_i}{\epsilon_s}, \forall i \in \llbracket 2, N-1 \rrbracket \end{cases}$$

Pour cette première implémentation, on prend la jonction a l'équilibre, i.e $V_g = 0$ et $V_d = V_{bi}$.

La construction de la matrice et la résolution de Poisson est implémentée dans la fonction `Poisson.m` avec le syntaxe $V = \text{Poisson}(X, \text{rho}, V_g, V_d, N_x)$ et donne le nouveau potentiel $V(x)$.

La verification de la convergence se fait a l'aide de la fonction `rms` sur matlab en calculant la valeur moyenne quadratique de la différence entre le nouveau et l'ancien potentiel, et en la comparant avec un erreur prédéfini qu'on prend ici égal a 10^{-8}

4.3 Calcul de n(x) et p(x)

On utilise les équations 1 pour déterminer n(x) et p(x) en utilisant le V(x) qu'on trouve dans l'étape précédente, i.e pour tout nœud i on prend:

$$\begin{cases} n_i = n_0 * \exp(\frac{qV_i}{k_b T}) \\ p_i = p_0 * \exp(\frac{-qV_i}{k_b T}) \end{cases} \quad (8)$$

Cette relation sera écrite différemment dans chaque cote de la jonction PN.

Dans le cote P (x<0):

$$p_0 = N_A, n_0 = \frac{ni^2}{N_A}$$

Dans le cote N (x>0):

$$n_0 = N_D, p_0 = \frac{ni^2}{N_D}$$

Le calcul de n et de p donne directement ρ selon l'équation 2.

Comme les étapes précédentes on utilise encore une fonction `charge_classiques.m` qui fait la mise a jour des listes [n,p,rho] selon le potentiel trouvé.

4.4 Résultats

On constate après l'exécution du programme que le potentiel ne converge pas et la simulation explose. Cela était prévu car les équation sont extrêmement non-linéaires, ce qui justifie l'approche dans la deuxième implémentation présentée dans la section suivante.

5 Deuxième Implémentation de Poisson avec un schéma Newton-Raphson

5.1 Poisson Amortie

On trouve une nouvelle implémentation avec un schéma de Newton-Raphson. On procède par considérer :

$$\begin{cases} \rho^{k+1} = \rho^k + d\rho \\ V^{k+1} = V^k + dV \end{cases}$$

Le k ici représente l'itération de la boucle présente dans l'algorithme dans la Fig 2, et on considère que la densité et le potentiel dans chaque itération diffère de la précédente par une quantité infinitésimal; ce qui représente un amortissement de l'équation de Poisson pour éviter la divergence.

On remplace $d\rho = \frac{d\rho}{dV} dV$, et on calcule la dérivée à l'aide des équations (2) et (1).

On trouve :

$$\frac{d\rho}{dV} = -\frac{q^2}{k_b T} (n + p)$$

Ce qui donne

$$\rho^{k+1} = \rho^k - \frac{q^2}{k_b T} (n + p) (V^{k+1} - V^k)$$

On utilise finalement Poisson pour remplacer ρ^{k+1} en fonction de V^{k+1} et on trouve finalement l'équation de Poisson amortie :

$$\Delta V^{k+1} - \frac{q^2}{k_b T} (n + p) V^{k+1} = -\frac{\rho^k}{\epsilon_s} - \frac{q^2}{k_b T} (n + p) \frac{V^k}{\epsilon_s} \quad (9)$$

5.2 Discretisation de Poisson amortie

Cette équation sera discrétisée comme on a fait pour l'équation de Poisson dans la partie 4.2, mais avec un différent Y et des différents éléments sur la diagonale de de la matrice, ce qui donne une équation matricielle :

$$LpNR * V = RHS_NR$$

On prend pour $LpNR_{i,j}$

$$\begin{cases} \frac{-2}{dx^2} - \frac{q^2}{k_b T} (n_i + p_i) & \text{si } i = j \\ \frac{1}{dx^2} & \text{si } i \pm 1 = j \\ 0 & \text{sinon.} \end{cases}$$

et pour RHS_NR_i

$$\begin{cases} RHS_NR_1 = -\frac{\rho_1}{\epsilon_s} - \frac{q^2}{k_n T} (n_1 + p_1) \frac{V_1}{\epsilon_s} - \frac{V_g}{dx^2} \\ RHS_NR_N = -\frac{\rho_N}{\epsilon_s} - \frac{q^2}{k_n T} (n_N + p_N) \frac{V_N}{\epsilon_s} - \frac{V_d}{dx^2} \\ RHS_NR_i = -\frac{\rho_i}{\epsilon_s} - \frac{q^2}{k_n T} (n_i + p_i) \frac{V_i}{\epsilon_s}, \forall i \in \llbracket 2, N-1 \rrbracket \end{cases}$$

Dans le code, la discrétisation est fait avec la fonction défini dans `Poisson_NR.m` ce qui donne `[LpNR, RHS_NR]` mais sans les conditions aux limites avec V_g et V_d .

ces conditions sont ajoutées avec la fonction dans `boundary_conditions.m`.

On intègre cette équation dans l'algorithme décrit dans Fig 2 pour la résolution du potentielle $V(x)$ et en gardant la même boucle, ce qui est implémente dans le script `second_implementation.m`.

5.3 Résultats

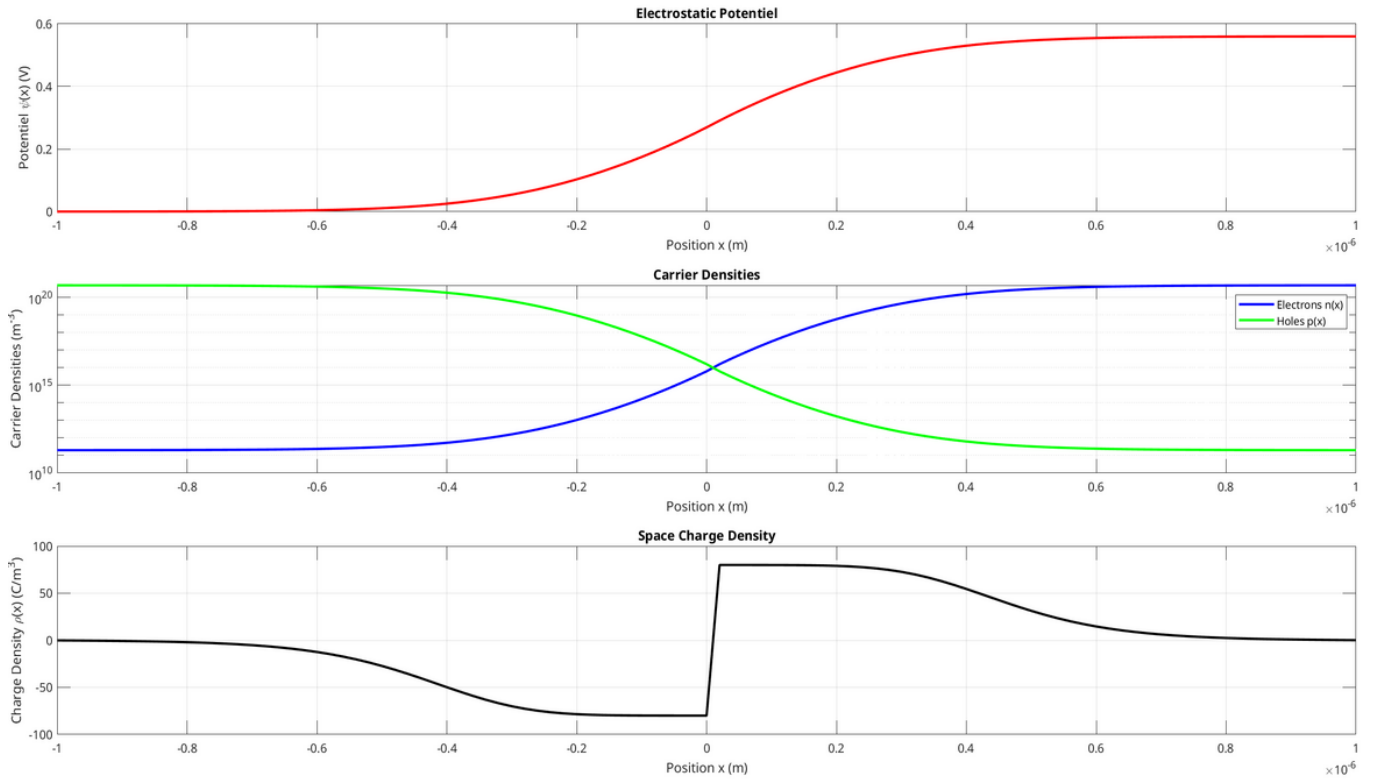


Figure 3: Résultats de l'implémentation de l'équation de Poisson amortie dans la boucle

On atteint la convergence en seulement 5 itérations de la boucle. On voit qu'on obtient un potentiel qui n'est pas trop loin de celui vu dans la figure 1 et obtenue à partir d'une densité de charge constante par morceaux. On met une comparaison en Fig 4. On voit notamment que la variation de potentiel est légèrement plus lente dans la résolution numérique dans les extrémités de la ZCE, cela est dû au fait qu'on a une variation de ρ qui est plus régulière comme on a vu et elle n'est plus constante par morceaux.

En revanche, on trouve une densité qui est plus régulière, qui représente des variations plus graduelles de la densité de charge ce qui semble plus naturel par rapport aux variations abruptes vu en cours.

Pour les densités de porteurs, ils sont donnés en échelle logarithmique vu leurs grands intervalles de variations, ce qui nous permet de voir les valeurs aux bords de la jonction, ce qui est bien consistant avec les valeurs de n_0 et p_0 données dans la section précédente.

On peut voir les variations plus clairement en utilisant une échelle linéaire comme dans Fig et qui montre plus clairement la zone de désertion

5.

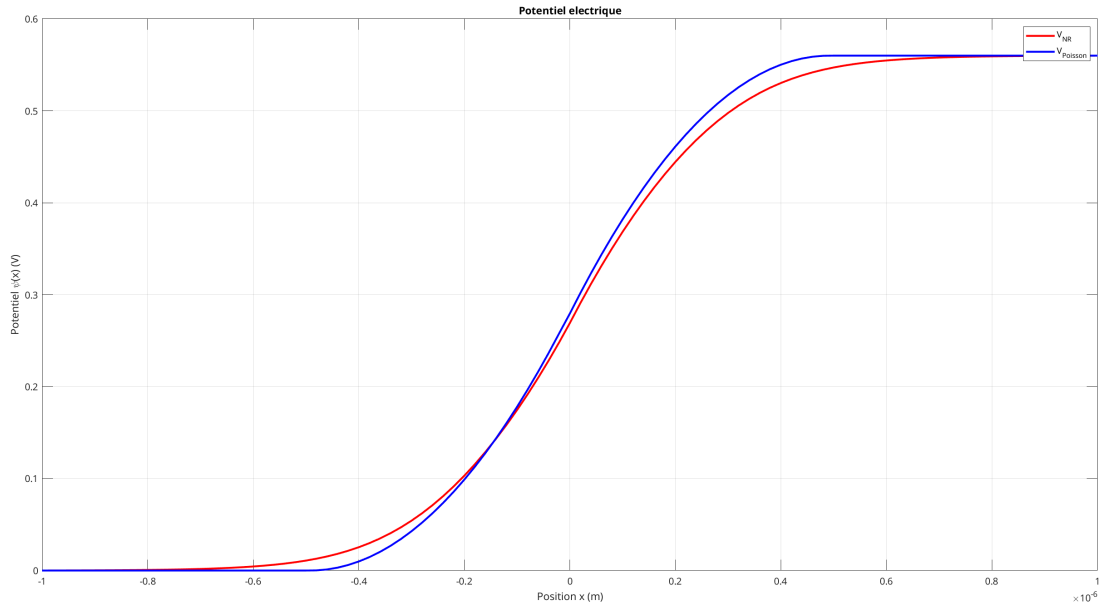


Figure 4: Comparaison entre le potentiel obtenu par cette simulation (en rouge) et celui vu en cours (en bleu).

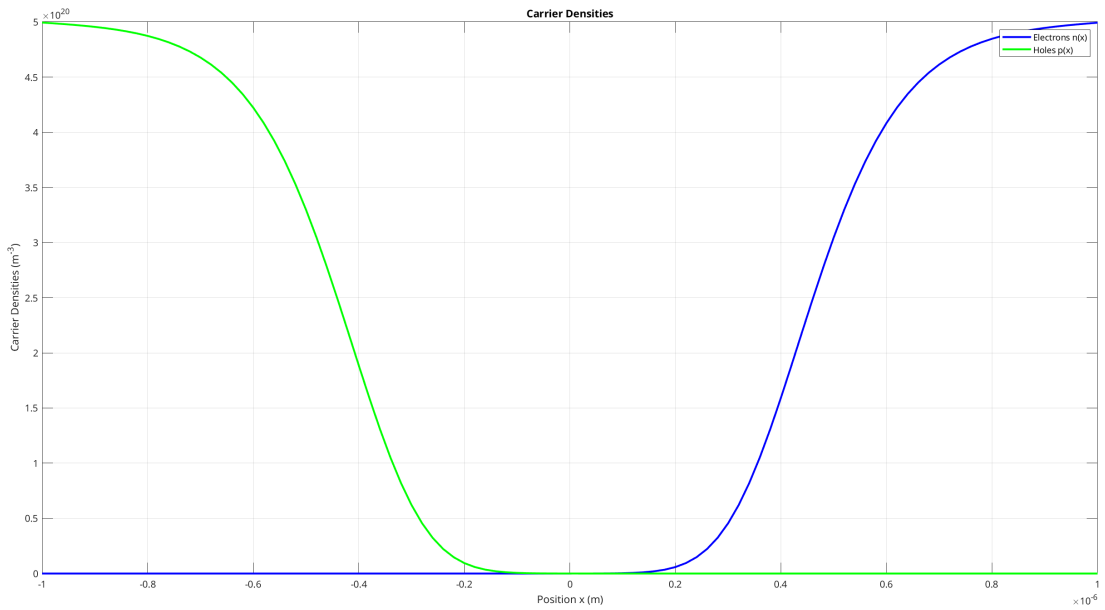


Figure 5: Concentration de porteurs obtenue par l'équation de Poisson amortie en échelle linéaire.

6 Résolution de courants dans la Jonction PN

On veut maintenant utiliser ces résultats pour déterminer la densité de courant dans la jonction PN en résolvant l'équation de continuité 5.

Néanmoins, une étude nous montre qu'une discrétisation de l'équation n'est pas suffisante car encore les équations sont fortement non-linéaire. L'idée sera d'imposer un amortissement sur les équations de calcul de $n(x)$ et $p(x)$ à partir de $V(x)$ dans les eqts 1.

Pour se faire, on crée un maillage intermédiaire dans les centres des éléments du maillage X utilisé pour la boucle (voir Fig 6).

On impose ensuite que le champ soit constant entre chaque deux nœuds i et $i+1$ de X. Cette condition donnera après développement de calcul une équation qui trouve les concentrations $n(x)$ et $p(x)$ à partir de $V(x)$ en prenant en

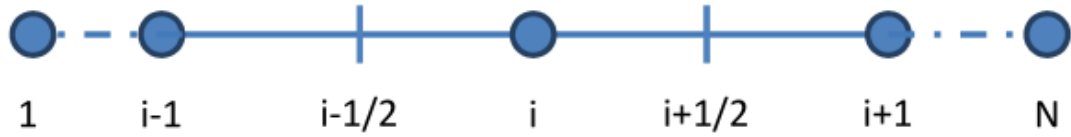


Figure 6: Maillage intermédiaire pour la résolution de l'équation de continuité.

compte l'effet des champs électriques forts qui peuvent causer le système a exploser.

Les calculs présentés dans le document du projet font apparaître des fonctions de Bernoulli $B(x) = \frac{x}{e^x - 1}$. Ces fonctions arrivent à réduire l'effet des grandes variations de potentiel (i.e. champ fort).

Ce modèle donne également une détermination du courant sur le maillage intermédiaire ce qui donnera en fin la caractéristique de la jonction.

La détermination du courant sera faite à l'équilibre et sous des polarisations directe (**Vapp_plus**) et indirectes (**Vapp_moins**). On utilisera la structure dans Fig 7.

L'initialisation des valeurs sera la même que pour la partie précédente avec **charge_classique** et **Poisson**.

En outre, On parcourt la liste **Vapp_plus** en ascendant en partant de **Vapp=0**, mais on parcourt **Vapp_moins** en descendant dans **deux boucles séparées**, on fait cela pour faire des petites variations de polarisation et commencer d'un potentiel proche de la vraie solution pour éviter la divergence du système. On calcule le courant pour chaque polarisation et on les stocke dans une liste **I** qui sera ordonnée pour suivre l'ordre croissant des polarisations **Vapp** qui n'est que l'union des deux intervalles au-dessus.

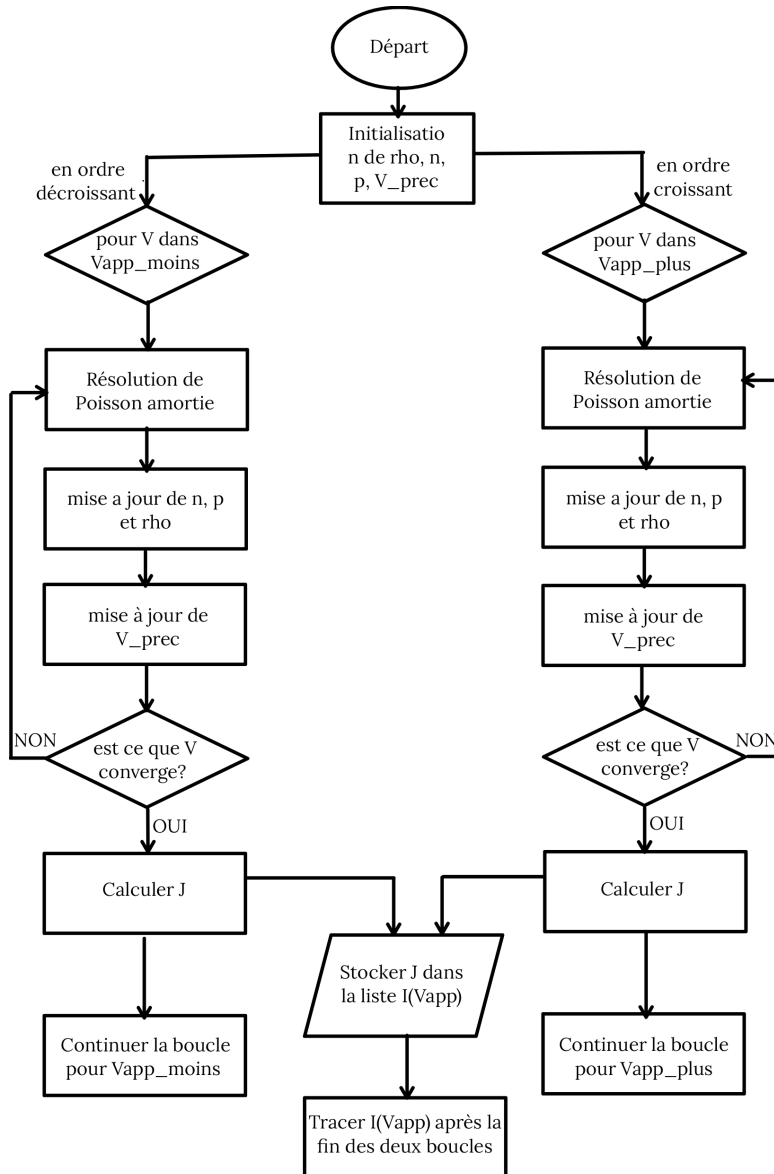


Figure 7: Algorithme pour la détermination de la caractéristique de la jonction.

6.1 Calcul de n(x) et p(x)

On trouve a l'aide du maillage intermediares deux systèmes a résoudre et qui fait intervenir les fonctions de Bernoulli, le premier systeme pour n :

$$\begin{aligned} & \frac{\mu_n^{i+1/2} k_b T}{(x_{i+1} - x_i) \left(\frac{x_{i+1} - x_{i-1}}{2} \right)} \left(n_{i+1} \cdot B \left(\frac{q(V_{i+1} - V_i)}{k_b T} \right) - n_i \cdot B \left(\frac{q(V_i - V_{i+1})}{k_b T} \right) \right) \\ & - \frac{\mu_n^{i-1/2} k_b T}{(x_i - x_{i-1}) \left(\frac{x_{i+1} - x_{i-1}}{2} \right)} \left(n_i \cdot B \left(\frac{q(V_i - V_{i-1})}{k_b T} \right) - n_{i-1} \cdot B \left(\frac{q(V_{i-1} - V_i)}{k_b T} \right) \right) = 0 \end{aligned}$$

Le deuxieme pour p :

$$\begin{aligned} & \frac{\mu_p^{i+1/2} k_b T}{(x_{i+1} - x_i) \left(\frac{x_{i+1} - x_{i-1}}{2} \right)} \left(p_{i+1} \cdot B \left(\frac{-q(V_{i+1} - V_i)}{k_b T} \right) - p_i \cdot B \left(\frac{-q(V_i - V_{i+1})}{k_b T} \right) \right) \\ & - \frac{\mu_p^{i-1/2} k_b T}{(x_i - x_{i-1}) \left(\frac{x_{i+1} - x_{i-1}}{2} \right)} \left(p_i \cdot B \left(\frac{-q(V_i - V_{i-1})}{k_b T} \right) - p_{i-1} \cdot B \left(\frac{-q(V_{i-1} - V_i)}{k_b T} \right) \right) = 0 \end{aligned}$$

On simplifie ces relations pour mettre les equations sous la forme matricielle :

$$\begin{cases} A_n * n = b_n \\ A_p * p = b_p \end{cases}$$

Pour $i = 2$ à $N - 1$, les matrices pour les électrons et les trous sont définies comme suit :

$$\begin{cases} A_n(i, i-1) = B \left(\frac{V_{i-1} - V_i}{V_t} \right) \\ A_n(i, i) = -B \left(\frac{V_i - V_{i-1}}{V_t} \right) - B \left(\frac{V_i - V_{i+1}}{V_t} \right) \\ A_n(i, i+1) = B \left(\frac{-V_i + V_{i+1}}{V_t} \right) \\ A_p(i, i-1) = B \left(\frac{-V_{i-1} + V_i}{V_t} \right) \\ A_p(i, i) = -B \left(\frac{-V_i + V_{i-1}}{V_t} \right) - B \left(\frac{-V_i + V_{i+1}}{V_t} \right) \\ A_p(i, i+1) = B \left(\frac{V_i - V_{i+1}}{V_t} \right) \end{cases}$$

On pose $V_t = \frac{k_b T}{q}$ pour alléger l'écriture.

Pour implementer les conditions aux limites de Dirichlet pour p et n on veut prend remplace la premiere et la derniere ligne pour chauqe matrice par des zeros sauf sur la diagonal ou on met des 1.

D'autre part, b_n et b_p sont nuls sauf aux bords, pour lesquels on pose :

$$\begin{cases} b_{p,0} = N_A \\ b_{n,0} = \frac{n_i^2}{N_A} \\ b_{p,N} = N_D \\ b_{n,N} = \frac{n_i^2}{N_D} \end{cases}$$

Ce qui assure l'implémentation des conditions aux limites.

Ce calcul est implémenté dans le code dans la fonction définie dans `charge_bernoulli.m`, qui assure la mise a jour de `[n,p,rho]`.

6.2 Résolution de Poisson amortie sous polarisation

On sait que le schéma Newton Raphson marche comme on a vu précédemment a l'équilibre, on va l'utiliser sous polarisation mais cette fois ci avec les concentrations de porteurs obtenue a l'aide des fonctions de Bernoulli. On trouve cette application dans le script `third_implementation.m`.

L'application d'une polarisation V_{app} changera les conditions aux limites, on met $V_d = V_{bi} - V_{app}$, mais la résolution reste la même que celle vu a l'équilibre.

Le changement des conditions aux limites est fait a l'aide de la fonction `boundary_conditions.m`, qui fait un mise a jour du système en changeant le second membre.

On obtient des courbes comme dans les Fig 8 et 9. Dans la première, on observe la diminution de la barrière de

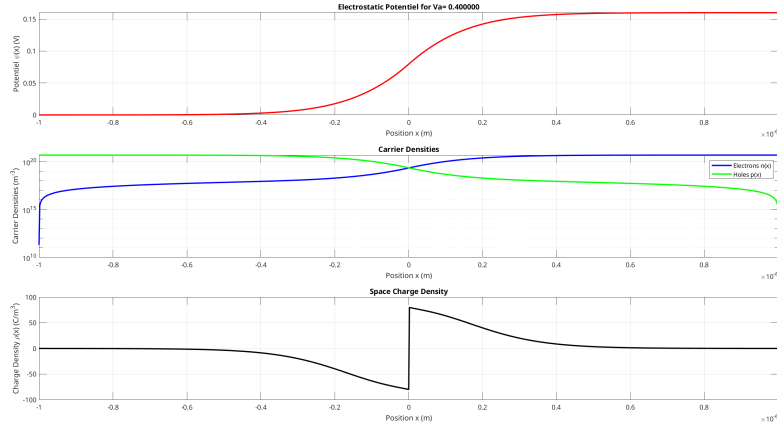


Figure 8: Polarisation direct avec $V_{app} = 0.4V$

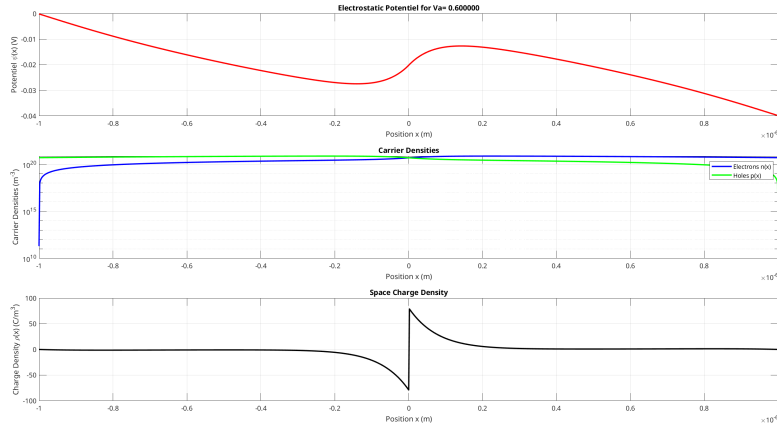


Figure 9: Polarisation direct avec $V_{app} = 0.6V$

potentiel au niveau de la jonction, comme prévu sous polarisation directe. De même, on constate la réduction de la zone de charge d'espace (ZCE). Cependant, dans la deuxième, lorsque la polarisation dépasse V_{bi} , on remarque que la courbe de potentiel perd sa forme caractéristique et tend à devenir quasi-linéaire, sauf au voisinage de l'interface. Autour de cette interface, on observe également une accumulation des charges d'espace, mais la ZCE ne disparaît pas complètement.

De même si on applique une polarisation indirecte on garde le même allure du potentielle mais la barrière devient plus grande et même chose pour la ZCE.

6.3 Détermination du courant

L'approche utilisée dans la section précédente donne aussi une relation de J_n et J_p en fonction des concentrations.

$$J_n^{i+1/2} = \frac{\mu_n^{i+1/2} k_b T}{x_{i+1} - x_i} \left(n_{i+1} \cdot B \left(\frac{V_{i+1} - V_i}{V_t} \right) - n_i \cdot B \left(\frac{V_i - V_{i+1}}{V_t} \right) \right)$$

$$J_p^{i+1/2} = -\frac{\mu_p^{i+1/2} k_b T}{x_{i+1} - x_i} \left(n_{i+1} \cdot B \left(\frac{-(V_{i+1} - V_i)}{V_t} \right) - n_i \cdot B \left(\frac{-(V_i - V_{i+1})}{V_t} \right) \right)$$

La résolution de ces équations donne les courants de chaque type de porteur sur un maillage XJ qui contient les points intermédiaire vu dans Fig 6. Les résultats fluctuent autour d'une valeur moyenne qui est plus ou moins constante sur tout le maillage comme present dans Fig 10.

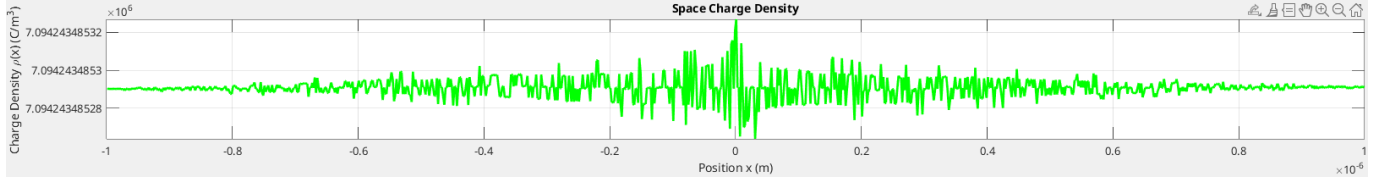


Figure 10: Fluctuation de la densité de courant $J(x)$

On prend alors la moyenne de chacun et on les somme pour avoir le courant totale a travers la jonction. Ces relations sont implémentées dans la fonction définie dans `Courants.m`. Il n'y a pas d'équation a résoudre, c'est juste un poste traitement de n et p pour trouver le J en traversant les indices de ces listes. En faisant cela pour chaque polarisation, on trouve finalement la caractéristique $I(V_{app})$ de la jonction a l'aide du script `Caractéristique.m`.

6.4 Résultats de la caractéristique

On obtient pour des polarisations entre -1V et 1V la caractéristique ci dessous. On voit dans la Fig 11 qu'elle possède bien l'allure exponentielle de la diode donnée par le modèle de Shockely. Une caractéristique sur un échelle semi-log (Fig 12) montrera plus clairement le comportement exponentielle. Elle permet de plus de vérifier les différent régimes de fonctionnement de la jonctions PN qui dicte la pente de la courbe.

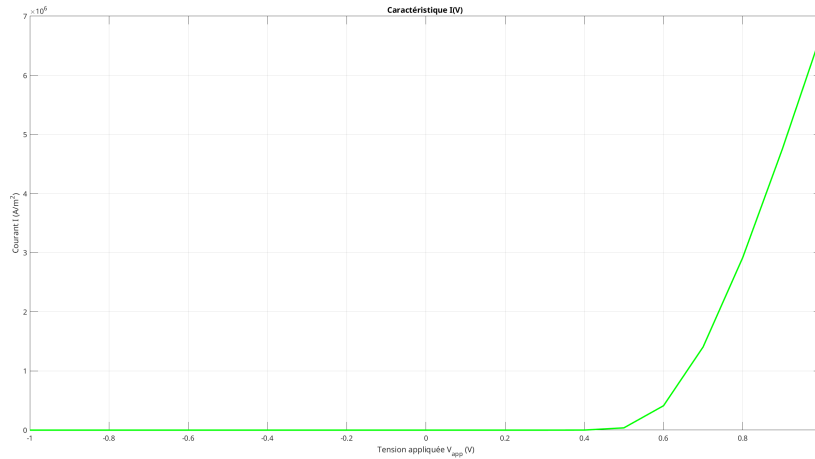


Figure 11: caractéristique de la jonction en échelle linéaire

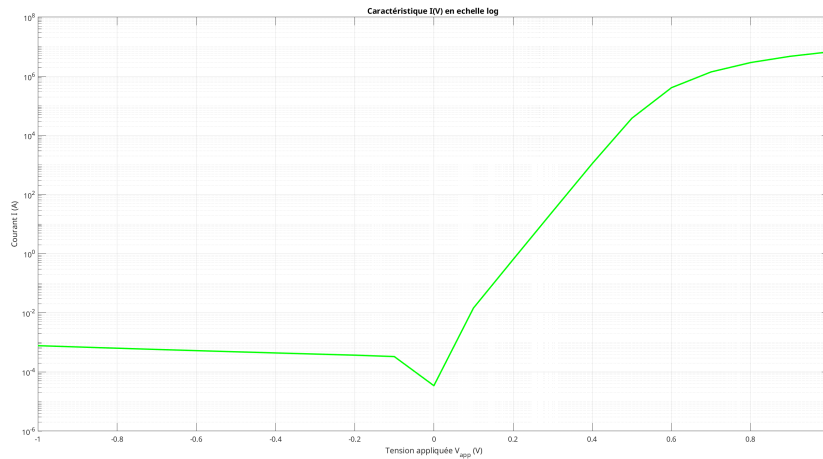


Figure 12: Caractéristique en échelle semi-log.

7 Conclusion

Conclusion

Ce projet nous a permis d'étudier le comportement électrique d'une jonction PN en régime stationnaire en utilisant la méthode des différences finies. Nous avons résolu les équations gouvernantes pour déterminer :

- La variation du potentiel électrique
- Le courant total
- Les concentration des porteurs de charge (électrons et trous) et la densité de charges.

Les résultats obtenus confirment les phénomènes physiques attendus:

- Sous **polarisation directe**, réduction de la barrière de potentiel et de la zone de charge d'espace (ZCE)
- Sous **polarisation inverse**, élargissement de la ZCE et augmentation de la barrière.
- On a bien obtenue la bonne caractéristique de la jonction PN.

On remarque que même si on met des polarisation indirecte importantes on ne voit pas l'effet avalanche qui fait que la jonction devient passante dans le sens inverse. Ce qui est normale car notre modèle ne prend pas l'ionisation. Une amélioration pour développer ce projet peut être de prendre en considération l'ionisation et d'autre type de génération et de recombinaison, vu que dans ce projet on a négliger ces phénomènes et on a considéré un régime stationnaire. Ce travail a renforcé notre compréhension des jonctions PN tout en illustrant la puissance des méthodes numériques pour la simulation de phénomènes physiques complexes en microélectronique.