

*Université Blida 1*



*Faculté Des Sciences*

*Département de Physique*

*En Vue de l'obtention d'un diplôme de master*

*Domaine : Science de la matière*

*Filière : Physique*

*Spécialité : Physique Des Matériaux*

*Thème :*

Simuler l'Aléatoire dans la Croissance d'une Monocouche Par EJM :  
Analyse Par La Méthode KMC

Cas : Adsorption-Diffusion en Surface (Basse Température)

*Présenté Par :*

*\*REZZOUG AZZEDDINE*

*Soutenu le 22 décembre 2021 devant Le jury Composé de :*

<i>Dr Y. Belaroussi</i>	<i>Maitre de recherche A</i>	<i>CDTA-Baba Hassen</i>	<i>Présidente</i>
<i>Dr S. Ouïr</i>	<i>Maitre de conférences B</i>	<i>U.Blida 1</i>	<i>Examinatrice</i>
<i>Dr A. Alï-Messaoud</i>	<i>Maitre de conférences A</i>	<i>U.Blida 1</i>	<i>Rapportrice</i>

*Blida , 2020/2021*

# Table des matières

<b>Résumé .....</b>	<b>ii</b>
<b>Table des figures .....</b>	<b>iv</b>
<b>Liste des tableaux .....</b>	<b>vi</b>
<b>Liste des abréviations.....</b>	<b>vi</b>
<b>Remerciements.....</b>	<b>vii</b>
<b>Introduction Générale .....</b>	<b>1</b>
<b>Chapitre I : Techniques Expérimentales</b>	
I.1. Généralités.....	3
I.2. Croissance par Epitaxie par Jet Moléculaires (EJM) .....	3
I.2.1. Epitaxie.....	3
I.2.2. Epitaxie par jets moléculaires.....	4
I.2.2.1. Définition.....	4
I.2.2.2. Principe de la technique EJM.....	5
I.3. Techniques de caractérisation.....	6
I.3.1. Caractérisations in situ.....	6
I.3.1.1. Caractérisation par diffraction des électrons de haute énergies RHEED.....	6
I.3.1.2. Spectroscopie des électrons Auger (AES) .....	7
I.3.1.2.1. Description du dispositif AES .....	7
I.4.1. Caractérisation <i>ex situ</i> .....	8
I.4.1.1. Microscope électronique en transmission TEM.....	8
I.4.1.1.1. Description et principe de fonctionnement.....	8
I.4.1.2. Diffraction des rayons X.....	10
I.4.1.2.1. Diffractomètre RX.....	10
I.4.1.2.2. Diffraction rasante.....	11
<b>Chapitre II : Généralités sur les couches minces</b>	
II.1. Introduction.....	12

II.2.	Généralités sur les couches minces.....	12
II.3.	Application des couches minces.....	13
II.4.	Mécanisme de formation d'une couche mince.....	15
II.4.1.	Nucléation.....	15
II.4.2.	Coalescence.....	16
II.4.3.	Croissance cristalline.....	16
II.5.	Méthodes de dépôt des couches minces.....	17
II.5.1.	Méthodes physiques .....	18
II.5.1.1.	Pulvérisation cathodique.....	18
II.5.1.2	Epitaxie par Jets Moléculaires.....	19
II.5.1.3	Ablation Laser.....	20
II.4.1.4	Evaporation sous vide.....	20

### Chapitre III : Tirage des nombres aléatoires

III.1.	Introduction .....	22
III. 2.	Définition d'un générateur de nombres pseudo-aléatoires .....	22
III.3.	Source de nombres pseudo-aléatoires.....	23
III.4.	Générateurs des nombres pseudo-aléatoires (GNPA) .....	23
A.	Générateur de Von Neumann (1946) .....	23
B.	Générateurs à congruence linéaire (1948) .....	24
III.5.	Méthodologie de tirage des nombres pseudo-aléatoires en C++.....	24
III.5.1.	Présentation de l'environnement Dev C++ version 5.11 .....	24
III.5.2.	Méthode d'utilisation de Dev C++ 5.11 pour l'écriture du GN-PA.....	25
III.5.3.	Application d'un code pour tirage des nombres aléatoires.....	26
III.6.	Tirage des nombre pseudo-aléatoire dans un intervalle $[0, N_{\max}-1]$ .....	30
III.7	Test sur le tirage des nombres pseudo-aléatoires .....	32
III.8.	Analyse des résultats du GN-PA .....	34

### Chapitre IV : Résultats et Discussion

IV.1.	Procédure de simulation de la croissance cristalline d'une couche mince....	35
IV.1.1.	Version 1 du simulateur : Simulation de l'Interaction Gaz/Surface (Adsorption) à basse température	35
IV.1.1.1.	Procédure de gestion des atomes arrivant à la surface du substrat.....	36
IV.1.1.2.	Procédure de programmation.....	37

IV.1.1.3.	Résultats et validation du simulateur version 1 : Adsorption à Basse Température	39
A.	Etude de l'Evolution de la croissance en fonction du temps de simulation ..	40
B.	Etude de la morphologie de la surface au cours de la croissance d'une monocouche	41
C.	Etude de l'Evolution du taux de couverture de la surface de la croissance ..	42
IV.1.2.	Version 2 du simulateur : cas de Diffusion en surface.....	43
A.	Processus élémentaires.....	44
B.	Architecture du simulateur -Version 2.....	45
C.	Procédure programmation du simulateur - version 2.....	47
D.	Conditions aux limites périodiques.....	48
VI.1.3.	Résultat de la simulation : Cas de diffusion en surface de croissance cristalline via la méthode KMC.	48
IV.1.3.1	Etude du taux de couverture de la surface de croissance.....	48
A.	Etude de l'évolution du taux de couverture de la surface de croissance .....	48
B.	Etude de la rugosité de surface et validation du résultat.....	53
IV.4	Conclusion.....	55
		56
	<b>Conclusion Générale</b> .....	
	<b>Perspectives</b> .....	57
	<b>Annexe A</b> .....	58
	<b>Annexe B</b> .....	59
	<b>Annexe C</b> .....	61
	<b>Bibliographie</b> .....	65

# Liste des Figures

<b>Figure.I.1.</b> Formation des couches épitaxie sur un substrat.	3
<b>Figure.I.2.</b> Image d'un microprocesseur.	4
<b>Figure.I.3.</b> Représentation d'une disposition de fer et D'oxygène sur un substrat	4
<b>Figure. I.4.</b> Schémas représente la rugosité d'une surface.	4
<b>Figure. I.5.</b> Image représente le bâti d'épitaxie par jet moléculaire.	5
<b>Figure.I.6.</b> Vue schématique d'un bâti d'épitaxie	5
<b>Figure. I.7.</b> Disposition des cellules à effusion et recouvrement des flux d'atomes près du substrat	5
<b>Figure. I.8.</b> Caractérisation d'une surface avec (RHEED).	6
<b>Figure. I.9.</b> Représentation des deux phénomènes [A] émission de Rayons X Caractéristique Et [B] émission d'électrons Auger.	7
<b>Figure. I.10.</b> Schéma représentatif de l'analyseur à miroirs cylindriques.	7
<b>Figure.I.11.</b> Image d'un microscope électronique en transmission.	8
<b>Figure.I.12.</b> Image de la matière obtenue par un microscope électronique en transmission.	9
<b>Figure.I.13.</b> Principe de fonctionnement d'un microscope électronique en mode microscope (a) et (b).en mode diffraction.	11
<b>Fig.I.14.</b> Schémas représentant le principe de la diffraction des rayons.	11
<b>Figure.I.15.</b> Schémas représentatif d'une anti cathode. X par les plans réticulaire d'indice (h,k,l) utilisé pour la production des rayons X.	10
<b>Figure.I.16.</b> Diffractomètre utilisé pour la diffraction des rayons X.	11
<b>Figure.II.1.</b> Dépôts d'une couche minces sur un substrat.	12
<b>Figure. II.2</b> Représentation du processus de fabrication des couches minces	13
<b>Figure.II.3.</b> Changement des propriétés microscopique magnétique entre un matériau massif et une couche minces.	15
<b>Figure II.4.</b> Représentation de l'étape de la nucléation des couches minces, (a) : l'arrivé des atomes sur un substrat, (b) la Morphologie du substrat.	16
<b>Figure II.5.</b> Les étapes ultérieures de la croissance.	16
<b>Figure II.6.</b> Les différentes étapes de croissance d'une couche mince.	17
<b>Figure II.7.</b> Diagramme des étapes du procédé de fabrication de couches minces.	18
<b>Figure II.8 :</b> Bâti de dépôt par pulvérisation cathodique. Les ions Argon, créés par L'excitation haute tension, arrachage de la cible les composés à déposer sur la surface des Substrats .	19
<b>Figure II.9.</b> Principe du dépôt de couches minces par ablation laser.	20
<b>Figure II.10.</b> Schéma conventionnel d'une évaporation thermique.	21
<b>Figure.III.1. (A-B-C-D-E-F-G-H).</b> Représentation les étapes de fonctionnement du programme Dev C++	24
<b>Figure III.2.</b> (A) Programme écrit en langage C++ (sous Windows) pour générer <b>un seul nombre pseudo-aléatoire</b> Avec la fonction {rand () } de la bibliothèque standard de ce langage, (B) la fenêtre de l'exécution du programme (A).	27
<b>Figure III.3.</b> (A) Programme écrit en langage C++ (sous Windows) pour générer <b>sept nombres pseudo-aléatoires</b> avec la fonction rand (), (B) la fenêtre de l'exécution du programme (A).	28
<b>Figure III.4.</b> (A) Programme en C++ (sous Windows) pour générer <b>un seul nombre pseudo-Aléatoire</b> avec la combinaison des fonctions srand et rand () de la bibliothèque standard de ce langage, (B) la fenêtre de l'exécution du programme (A).	29
<b>Figure III.5.</b> Programme en C++ (sous Windows) pour générer <b>un seul nombre pseudo-aléatoire</b> avec la combinaison des fonctions srand(), time(NULL) et rand () de la bibliothèque standard de ce langage.	29
<b>Figure III.6.</b> Trois consoles représentant chacune une exécution du programme contenant les trois fonctions (srand), (time) et (rand).	30
<b>Figure III.7.</b> Programme en C++ (sous Windows) pour générer <b>un seul nombre pseudo-aléatoire</b> dans l'intervalle [0,100] avec la combinaison des fonctions "srand ()," time(NULL)," rand () "et "modulo", de la bibliothèque standard de ce langage.	30

<b>Figure III.8.</b> Programme en C++ (sous Windows) pour générer <b>un seul nombre pseudo-aléatoire dans l'intervalle</b> [Min, Max] avec la combinaison des fonctions "srand ()", "time(NULL)", "rand ()" et "modulo", de la bibliothèque standard de ce langage.	
<b>Figure III.9.</b> Programme en C++ (sous Windows) pour générer <b>300 nombre pseudo-aléatoire dans l'intervalle</b> [Min, Max] avec la combinaison des fonctions "srand ()", "time(NULL)", "rand ()" et "modulo", de la bibliothèque standard de ce langage	
<b>Figure.III.10.</b> Uniformité du GN-PA pour 100 tirages dans l'intervalle [1-100].	31
<b>Figure.III.11.</b> Uniformité du GN-PA pour 300 tirages dans l'intervalle [1-100].	
<b>Figure.IV.1.</b> Type de configuration d'Adsorption en surface "	32
<b>Figure IV.2.</b> (A) Fichier « Mesdocument » représentant les Inputs du Simulateur. (B) Structure du calendrier des « Arrivés » noté Cal 1. La dimension de cal 1 est égale à la dimension du substrat (en nombre de sites).	33
<b>Figure IV.3.</b> Organigramme du simulateur de la croissance cristalline par la méthode KMC SPARCC– <b>Version 1 : Interaction Gaz/Surface</b> par « <b>Adsorption</b> » des atomes en surface à basse température	36
<b>Figure IV.4.</b> Tracé de l'évolution du nombre des évènements en fonction du temps de simulation. eve : est le nombre d'atomes incidents, et eveA : est le nombre d'atomes adsorbés pour une surface de 10x10 et Eve max=500.	40
<b>Figure IV.5.</b> Evolution de la morphologie de surface pour différentes valeurs du nombre d'évènement total. Les sites vacants sont représentés en rouges et les sites occupés en bleu. Les surfaces colorées en violet représentent la couverture en cours de croissance sur le substrat représenté avec la surface blanche.	41
<b>Figure IV.6.</b> Représentation du taux de couverture d'une surface 10X10 en fonction de Nbres d'évènements. (A) Tracé dans l'intervalle [0-3000] évènements (B) Tracé dans l'intervalle [0-80] évènements.	42
<b>Figure IV.7.</b> (A) modèle multicouche théorique et de simulation d'autres auteurs Madhukar [30] et S.V. Ghaisas [31] (croissance couche par couche, chaque courbe est une monocouche) (B) Isothermes d'adsorption de Langmuir : $T1 < T4$ [32]	43
<b>Figure IV.8 :</b> Schéma du processus élémentaire de la croissance cristalline en couche mince. (A) Modèle de boule, (B) Modèle de cube de Kossel (Solid On Solid)	44
<b>Figure IV.9 :</b> processus de diffusion d'un adatome vers le site de son voisin droit.	44
<b>Figure IV.10 :</b> (A) Les situations de diffusion d'un adatome vers un site vacant ayant une structure de coin ou de lacune. (B) Les quatres positions de proches voisins en surface d'un adatome en vu de sa diffusion.	45
<b>Figure IV.11.</b> Structure des calendriers :(A) « Arrivés » noté Cal 1, et (B) « Diffusion » noté Cal 2. La dimension de cal 1 est égale à la dimension du substrat (en nombre de sites). La dimension de Cal 2 dépend du nombre des adatomes.	45
<b>Figure IV.12.</b> Organigramme du simulateur de la croissance cristalline par la méthode KMC – Version 2 – Adsorption et diffusion des atomes adsorbés en surface, à basse température.	46
<b>Figure IV.13 :</b> Représentation schématique du découpage d'un l'échantillon réelle de façon périodique et application des conditions aux limites sur la partie simulée de l'échantillon	48
<b>Figure IV.14.</b> Représentation du taux de recouvrement d'une surface 20X20 en fonction de Nombre d'évènements. (A)Tracé globale (B) Analyse du Tracé	50
<b>Figure IV-15 :</b> Statistiques relevées durant la phase initiale de la croissance. (Haut) type d'évènements et type de sites en surface (Bas) Type de sites sur la surface de croissance en %	51
<b>Figure.IV.16. Evolution</b> de la morphologie de surface d'une surface=20X20 sites, pour certains Taux de couverture (A, B,C,D) et détermination du nombre de sites vacants en fonction du nombre d'Evènements (Arrive+diffusion) . Les sites vacants sont représentés en vert, les lacunes en bleus, leurs nombres sont mentionnés en bas de chaque configuration de surface obtenue.	52
<b>Figure.IV.17 .</b> Représentation de la rugosité en surface en fonction du temps de simulation pour une surface de (20x20) sites de notre simulateur version 2.	54
<b>Figure .IV.18.</b> Evolution de la rugosité temporelle (simulation) pour une croissance (a) d'une monocouche (b) de plusieurs couches[27]	54

## Liste des Tableaux

<b>Tableau. II.1.</b> Domaine d'application des couches minces.	14
<b>Tableau III .1.</b> Résultats des nombres P. Aléatoire compris entre [1-100] pour 100 fois de tirages.	32
<b>Tableau III .2.</b> Résultats des nombres P. Aléatoire compris entre [1-100] pour 300 fois de tirages	33
<b>Tableau.IV.1.</b> Présentation des différentes informations obtenu par SPARCC (Simulateur PARTiculaire de la croissance cristalline) sur une surface du substrat $S=20 \times 20$ sites et pour Nbr d'Eve [150-5000] , dans les 2 versions de SPARCC-Basse température.	49
<b>Tableau.IV.2.</b> Méthode d'estimation de la rugosité, dans SPARCC-BT-V2 , au cours de la croissance cristalline via la méthode KMC. ( $h_{ij}=1$ monocouche unité « a » paramètre de maille, et $N=$ dimension du substrat = $20 \times 20$ sites en surface)	53

## Liste des Abréviation

<b>MBE :</b> Moléculaire Beam Epitaxy (Epitaxie par jets moléculaires)
<b>RHEED :</b> Réflexion High Energy Electron Diffraction (Diffraction d'électrons de haute énergie en réflexion)
<b>AES :</b> Auger Electron Spectroscopy (Spectroscopie des électrons Auger)
<b>TEM:</b> Transmission Electronique Microscopy
<b>UHV:</b> Ultra High Vacuum
<b>GN-PA:</b> Générateur des nombres pseudo-Aléatoires
<b>Pvd :</b> (physical Vapor Deposition)
<b>DC :</b> direct courant
<b>VD :</b> voisinage droite.
<b>VG :</b> voisinage gauche.
<b>VH :</b> voisinage haute.
<b>VB :</b> voisinage bas.
<b>Cal-1 :</b> un Calendrier correspondant à l'impact d'atomes depuis la phase gazeuse.
<b>Cal-2 :</b> un Calendrier correspondant au mouvement des atomes sur la surface du dépôt.
<b>P.P:</b> Programme principe.
<b>KMC:</b> Kinetic Monte Carlo.
<b>DFT:</b> Density Functional Theory.

## Remerciements

Le présent travail de mémoire, en physique des matériaux, a été soutenu par le laboratoire de surface interface et couche mince LASICOM, du département de physique, faculté des Sciences- Université Blida 1.

Je remercie Professeur M. Derbal, responsable du master, spécialité physique des matériaux et Dr F. Bouamra directrice du LASICOM qui m'ont donné l'opportunité d'intégrer cette spécialité. Je tiens à exprimer aussi toute ma reconnaissance à ma directrice de mémoire Dr. A. Ali Messaoud, pour son soutien, son encouragement et sa compréhension dans les moments difficiles. Je la remercie chaleureusement pour tous ses efforts, son dévouement dans le suivi de ce travail, ainsi que son professionnalisme et sa grande générosité scientifique et humaine. Je voudrais adresser un Grand Merci à Mr Z.Meliani, enseignant à l'université de Blida-1 qui m'a beaucoup aider et encourager pour terminer mes études. Je remercie aussi Mme S. Saidani, et Mr A.Bencherchali, Pr. R.Bellal, , Pr. B.Rekik, , Dr. B Boutahraoui, de l'université de blida-1. Je remercie Dr. S.Ouir et Dr. Y.Belaroussi qui mon fait l'honneur d'être les rapporteurs de mon travail et ont accepté de consacrer du temps pour la lecture de mon mémoire. Leurs remarques et suggestions pour ce travail de mémoire vont me permettre d'améliorer ce dernier. J'adresse de chaleureux remerciements à toute l'équipe du laboratoire LASICOM pour leurs soutiens ainsi que leurs aides. Je souhaite exprimer ma profonde gratitude à mes parents, mes sœurs et mes frères pour leurs soutiens indéfectible depuis toujours.

Je tiens également à remercier mes collègues qui étaient présents au cours de ce travail, sans oublié K.Kwassi Y. Saadi et Z.Keddam.



## Résumé :

L'augmentation de la puissance informatique disponible permet aux ingénieurs et designers d'aborder par simulation des problèmes de plus en plus complexes (multi-physiques, multi-échelles et géométries étendues). Dans ce contexte, les simulations à base de la méthode de Monte-Carlo cinétique apparaissent plus appropriées, avec l'utilisation des modèles probabilistes et le concept de nombres pseudo-aléatoire implémenté dans la simulation type Monte Carlo pour créer un outil d'aide à la décision permettant ainsi de concevoir et piloter efficacement des systèmes complexe en technologie.

Après une étape d'étude des capacités du générateur des nombres pseudo-aléatoire choisi, l'outil est employé sur des cas d'illustration concernant la croissance d'une monocouche sur un substrat par la technique d'épitaxie par jets moléculaire, limité aux basses températures.

L'outil de simulation réalisé est analysé en termes de l'évolution du taux de couverture et de rugosité en fonction du temps de simulation, et validé par comparaison à des approches théorique (isothermes) et expérimentales (RHEED). Son utilisation peut améliorer les processus d'optimisation et de contrôle en technologie, ou simplifier les approches théoriques les plus utilisées.

**Mots-clés :** Méthode Monte-Carlo cinétique, générateur des nombres pseudo-aléatoire, Croissance cristalline, couches minces, Multi-échelle, Multi-physique, MBE.

## Abstract :

As computing power increases, engineers and designers tackle increasingly complex problems using simulation (multiphysics, multiscale, extended geometries ...). In this context, simulations based on the kinetic Monte-Carlo method seem more appropriate, with probabilistic models and the notion of pseudo-random numbers, to create a support tool allowing the management of complex systems in technology.

After a successful validation step to choose a pseudo-random number generator, this tool is applied to illustrative cases of one monolayer growth's, on a substrate, by the molecular beam epitaxy technique, limited to low temperatures.

This simulation tool realized, is analyzed via the coverage rate and roughness behaviors as a function of the simulation time, which shows its conformity with the theoretical (isothermal) and experimental (RHEED) results. Its use can improve optimization and control processes in technology, or simplify the most widely used theoretical approaches.

**Keywords:** Kinetic Monte-Carlo method, pseudo-random number generator, Crystal growth, thin layers, Multi-scale, Multi-physics, MBE.

## ملخص:

تسمح الزيادة في قوة الكمبيوتر المتاحة للمهندسين والمصممين بالتعامل مع المشكلات المعقدة بشكل متزايد (الفيزياء المتعددة، المقاييس المتعددة، الأشكال الهندسية الممتدة، إلخ) عن طريق المحاكاة. في هذا السياق، تبدو عمليات المحاكاة التي تعتمد على طريقة مونت كارلو الحركية أكثر ملاءمة، مع استخدام النماذج الاحتمالية ومفهوم الأرقام الشبه-عشوائية المطبقة في محاكاة نوع مونت كارلو لإنشاء أداة للمساعدة في اتخاذ القرار وبالتالي اتخاذها في تصميم وتجريب أنظمة معقدة في التكنولوجيا بشكل عام.

بعد نجاح خطوة دراسة قدرات مولد الأرقام الشبه-عشوائية المختارة، يتم استخدام الأداة في حالات توضيحية تتعلق بنمو أحادي الطبقة على ركيزة بتقنية MBE، مقتصرين في ذلك على درجات حرارة منخفضة.

يتم دراسة أداة المحاكاة الناتجة عن طريق رسم تطور معدل التغطية والخشونة بدلالة زمن المحاكاة، و يتم التحقق من النتائج عن المقارنة مع النماذج النظرية (متساوي الحرارة) والتجريبية.

**الكلمات المفتاحية:** طريقة مونت كارلو الحركية، مولد الأرقام العشوائية، النمو البلوري، الطبقات الرقيقة، المقاييس المتعددة، الفيزياء المتعددة،

..MBE

## INTRODUCTION GENERALE

La simulation est la reproduction artificielle du fonctionnement d'un modèle étudié. Elle permet notamment de valider ou d'invalider des hypothèses, d'obtenir des informations quantitatives (qui peuvent venir affiner des informations qualitatives si elles existent), de valider certaines approximations, d'évaluer la sensibilité d'un modèle à certaines hypothèses ou à certains paramètres, ou tout simplement d'explorer le comportement d'un modèle lorsque celui-ci est mal connu ou mal compris.

La simulation connue sous le nom générique de méthode de Monte-Carlo (par référence aux jeux de hasard des casinos) nécessite le recours à des nombres aléatoires. De nombreux phénomènes physiques peuvent cependant, de manière plus ou moins complexe, être traduits en termes de modèles numériques et mise en œuvre avec un modèle mathématique qui se prête le mieux à explorer une situation réelle et étudier ses propriétés.

Nous avons dans ce travail de master, en physique des matériaux, abordé le problème de la simulation de phénomènes aléatoires relativement simples liées à la croissance cristalline des couches minces telle que

1/ le phénomène d'adsorption atomique à partir d'un jet atomique ou moléculaire sur une surface de croissance (cellule d'effusion de Knudsen dans la technique MBE). Ceci a constitué notre première version du simulateur nommé : SPARCC-BT (Simulateur PARTiculaire de la Croissance Cristalline à Basse Température)

2/ et la diffusion équiprobable des adatoms (atomes adsorbés) sur une surface de croissance cristalline (dans la technique MBE). Cette diffusion associée à l'adsorption a constitué une deuxième version du simulateur nommé : SPARCC-DBT (Simulateur PARTiculaire de la Croissance Cristalline avec Diffusion à Basse Température).

Contrairement au modèle déterministe, notre modèle de la croissance cristalline en couches minces par épitaxie par jets moléculaires est une des applications des simulations par la méthode de Monte Carlo cinétique. C'est un processus complexe qui dépend de l'utilisation des nombres aléatoires et permet d'estimer un résultat souvent trop long à déterminer de manière déterministe. Ce modèle est dit « stochastique » et les simulations correspondantes comportent par conception un aspect aléatoire et un aspect déterministe. Pour l'environnement informatique, la simulation numérique stochastique partage la puissance de calcul ou grille de calcul à 50 % du cycle de traitement des données par utilisation des nombres pseudo-aléatoires et 50% à l'aspect déterministe du modèle simulé (calcul énergétique, calcul temporel, ...etc).

### **Choix du langage de simulation :**

L'utilisation d'un langage compilé tel que C++ (ou Fortran) garantit en général une exécution très rapide, ce qui peut être crucial pour effectuer des simulations d'une certaine ampleur, et permet de contrôler le fonctionnement de la simulation dans tous ses détails. En revanche, la conception du programme, l'écriture du code, sa vérification et la correction des erreurs qu'il contient constituent des tâches lourdes et parfois délicates.

L'utilisation de logiciels spécialisés de simulation, par exemple pour la simulation des systèmes à événements discrets, peut rendre la mise en œuvre d'une simulation beaucoup plus facile, au prix d'un moindre contrôle des opérations exécutées, et de limitations éventuelles de capacité (sans oublier le coût d'achat du logiciel et le nécessaire apprentissage de son fonctionnement). Enfin, l'utilisation de logiciels de portée générale comme Matlab, dans ce contexte, est surtout adaptée à un usage pédagogique.

### **Structure du manuscrit :**

Le premier chapitre (Chap. I) nous a permis, tout d'abord, de positionner le système que nous avons considéré dans les différentes versions du simulateur. Après avoir présenté la technique d'épithaxie par jets moléculaire (MBE : Molecular Beam Epitaxy) que nous avons utilisé dans ce travail, nous nous sommes focalisés, dans le deuxième chapitre (Chap. II), sur la présentation des modèles de croissances cristallines en couches minces et les différents mécanismes intervenant dans ce processus. Ces mécanismes avec leurs deux aspects, aléatoire et énergétique, sont nécessaires pour la conception d'un simulateur par la méthode de Monte Carlo cinétique.

Le chapitre trois (Chap. III), a pour but de proposer un outil de gestion de l'aspect aléatoire des événements responsables de la croissance cristalline, à la fois performante et applicable dans le cas général. Il s'agit de la fonction « rand » du compilateur C++, que nous proposons à travers des exemples permettant l'observation de son concept.

Dans le dernier chapitre (Chap. IV) de ce document sont présentés les travaux de développement de la simulation qui ont été réalisés lors de ce travail. L'analyse des résultats du simulateur développé et sa validation par des travaux expérimentaux et théoriques ont été aussi présentées dans ce chapitre.

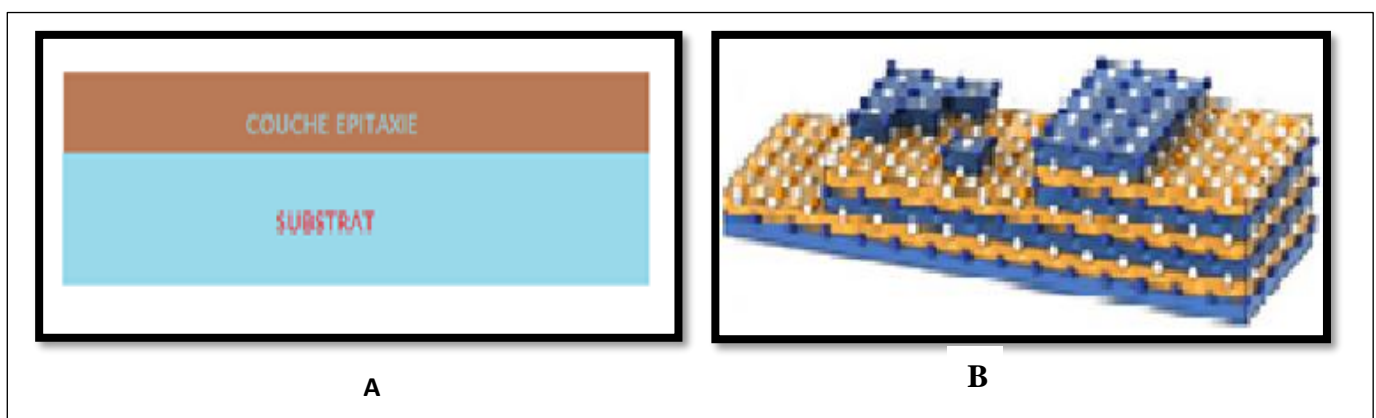
En conclusion et perspectives, et parce que la construction d'un logiciel de simulation passe avant tout par un compromis entre la réalité physique et la rapidité d'exécution des simulations, nous finirons notre manuscrit par discuter des compléments à réaliser pour continuer d'une part à améliorer ce modèle et d'autre part pour nous permettre la validation d'une plus grande diversité de phénomènes physiques intervenant dans l'ensemble de la réalisation.

# Chapitre I

## Techniques Expérimentales

### I.1. Généralités

La réalisation des couches minces se fait par déposition d'atomes ou de molécules sur un substrat ou sur une couche mince antérieurement déposée (Figure.I.1). Pour s'assurer que le processus de fabrication et de contrôle se déroulera correctement, les chercheurs ont développé plusieurs techniques et méthodes pour répondre aux besoins de la recherche et de l'industrie. Parmi ces techniques, on cite l'Épitaxie par Jet Moléculaire qui nous intéresse dans ce travail de mémoire.



**Figure.I.1.** Formation des couches épitaxiées sur un substrat.

### I.2. Croissance par Epitaxie par Jet Moléculaires (EJM)

#### I.2.1. Epitaxie

Par définition, l'« épitaxie » est la formation d'une couche sur un substrat monocristallin (Figure.I.1A) [1]. Elle est utilisée dans l'élaboration des structures en couches minces de très haute qualité qui rentrent dans la fabrication des micros et nano composants électronique (Figure.I.2). La couche épitaxiée et le substrat peuvent être du même matériau (homoépitaxie) et peuvent être différent (hétéroépitaxie) [2].

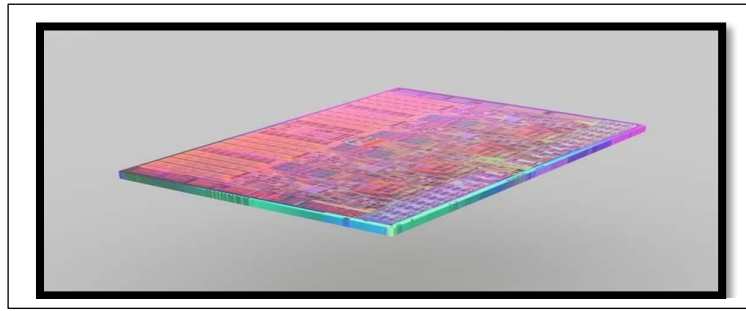


Figure.I.2. image d'un microprocesseur.

## I.2.2. Epitaxie par jets moléculaires

### I.2.2.1. Définition

L'épitaxie par jet moléculaires (EJM ou MBE en Anglais) est une technique de croissance cristalline qui permet de :

- 1- Élaborer des hétéro structures composées d'empilements de très faibles épaisseurs, avec une bonne qualité cristalline [3].
- 2- Réaliser des homoépitaxies, avec une vitesse de croissance relativement lente (de l'ordre de 1 à 2  $\mu\text{m/h}$ ). Ceci correspond à une ou deux monocouches par seconde, sous vide très poussés UHV (Ultra High Vacuum,  $10^{-11} \text{ Torr}$ )[2].
- 3- Présente un certain nombre de caractéristiques originales [3] :
  - ✓ Faible vitesse de croissance,
  - ✓ Arrêt et démarrage des croissances faciles
  - ✓ Possibilité de changements de la composition du dopage
  - ✓ Très bon contrôle des épaisseurs des couches très minces.

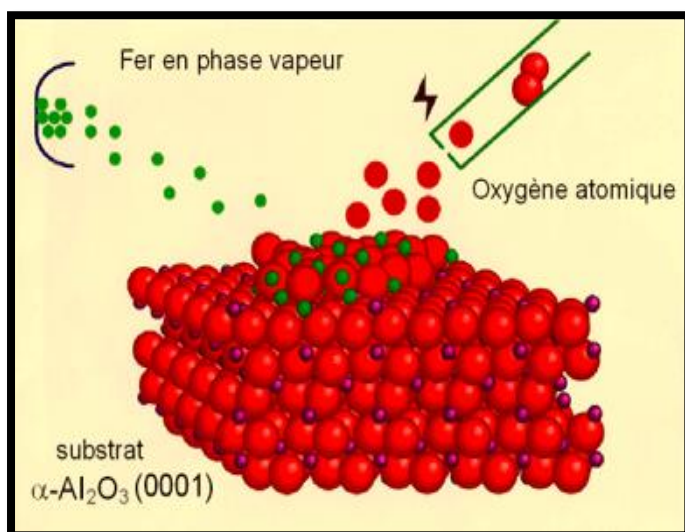


Figure.I.3. Représentation d'une disposition de fer et d'oxygène sur un substrat

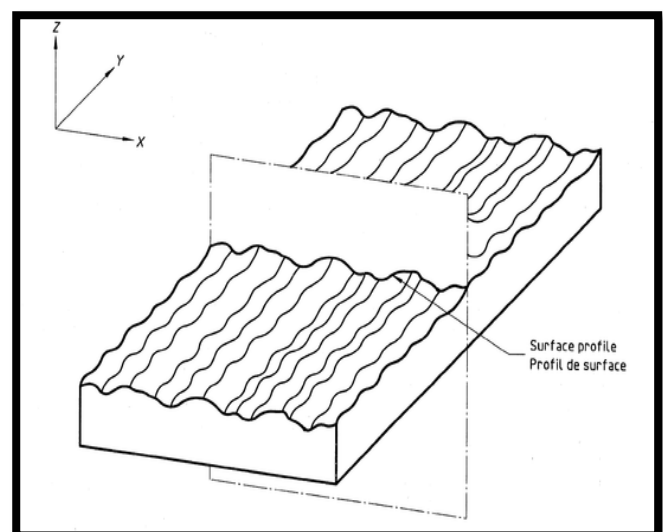


Figure. I.4. Schémas représentant la rugosité d'une surface.



### I.2.2.2. Principe de la technique EJM

Le système d'épithaxie par jets moléculaires englobera les éléments suivants (Figure.I.5,I.6,I.7) [4-5] :

- Un bâti à ultra vide.
- Un ensemble de pompe à vide
- Un porte substrat chauffant
- Des cellules d'évaporation pourvues de caches amovibles
- Des systèmes d'analyse « *in situ* » pour le contrôler et l'analyser de la couche épithaxié.
- Système d'introduction du substrat
- Système de nettoyage (canon à ions Argon)

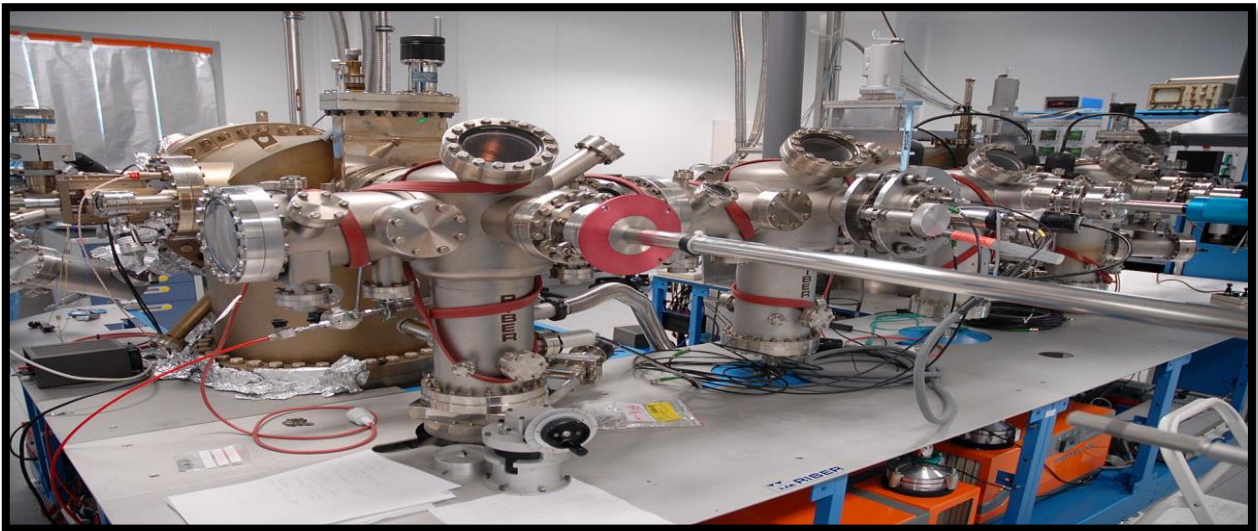


Figure. I.5. Image du bâti d'épithaxie par jets moléculaires.

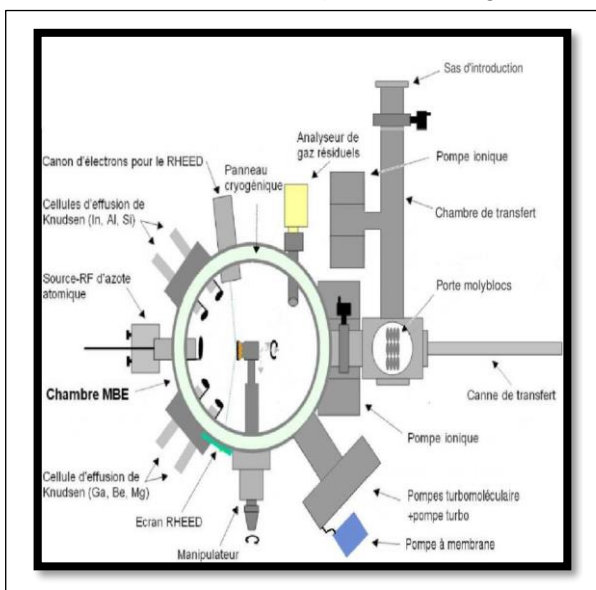


Figure.I.6. vue schématique d'un bâti d'épithaxie

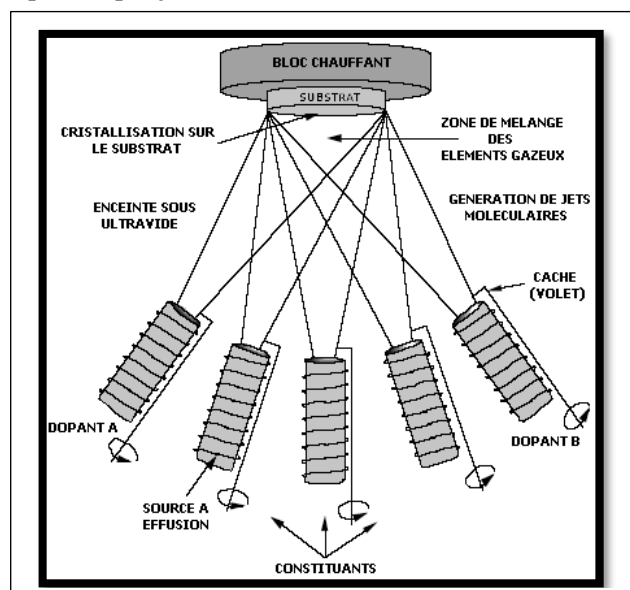


Figure. I.7. Disposition des cellules à effusion et des flux d'atomes près du substrat

## I.3. Techniques de caractérisation

### I.3.1. Caractérisations *in situ*

Lors de la croissance cristalline en couches minces, deux techniques d'analyses (*in situ*) sont importantes pour obtention des informations sur la nature surface ((structure, composition, mode de croissance) [6]:

1. la diffraction des électrons de haute énergie (RHEED)
2. et la spectroscopie d'électrons Auger (AES).

En effet, les électrons qui sont utilisés comme sonde de surface interagissent fortement avec les premiers couches atomiques de l'échantillon, ce qui permet de contrôler et de maîtriser la croissance et la formation des couches minces.

#### I.3.1.1. Caractérisation par diffraction des électrons de haute énergies RHEED

La technique de diffraction d'électrons de haute énergie en incidence rasante (RHEED : réflexion High Energy Electron Diffraction) est une technique utilisée pour le suivi de la croissance épitaxiale. Il s'agit d'une méthode (*in situ*) permettant l'obtention des informations structurelles et morphologique de la croissance

Le principe de RHEED repose sur l'utilisation d'un canon à électrons produisant un faisceau d'électrons monocinétiques de haute énergie (10 à 40 keV) dirigé sur l'échantillon sous incidence rasante (1 à 3°), ce qui permet alors de ne sonder que les premiers plans atomiques. La détection s'effectue par un écran fluorescent. L'information apparaît alors sous la forme d'un diagramme de diffraction. (Figure.I.8)

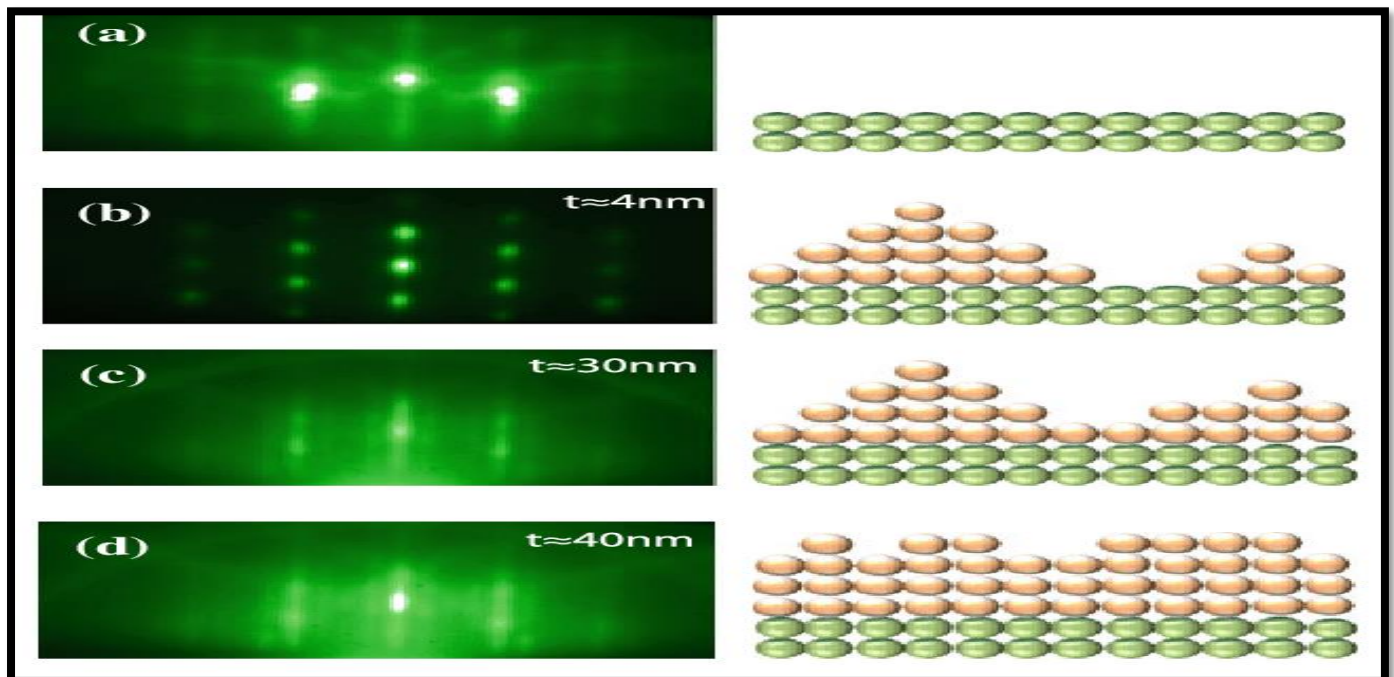


Figure. I.8. Caractérisation d'une surface avec (RHEED).

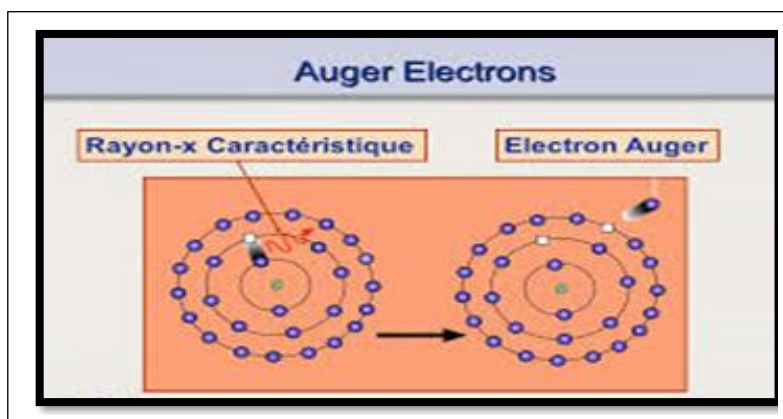
### I.3.1.2. Spectroscopie des électrons Auger (AES)

La spectroscopie des électrons Auger (AES) est une technique utilisée dans l'analyse physico-chimique de la surface. Elle nous renseigne sur la composition chimique de la surface du cristal. Elle s'applique pour la vérification de la propreté d'une surface [7].

#### I.3.1.2.1. Description du dispositif AES

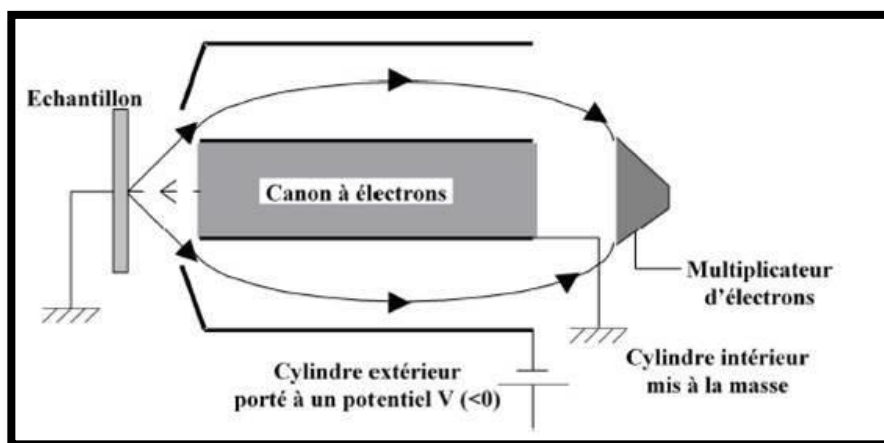
Le dispositif utilisé dans la chambre d'épitaxie est constitué de trois éléments principaux [2,7] :

1- Canon à électrons qui émet des électrons (dits primaires) ayant une énergie entre 2000 et 3000 ev. Ceci entraîne l'ionisation des atomes de surface de l'échantillon et l'arrachage des électrons des niveaux de cœur (figure I.8).



**Figure. I.8.** Représentation des deux phénomènes : émission de Rayons X  
Et émission d'électrons Auger

- 2- Analyseur à miroirs cylindriques (CMA) constitué de deux cylindriques concentriques entre lesquels est appliquée une différence de potentiel (figure.I.9). Seuls les électrons (énergie E) réémis suivant un angle  $\theta$  bien définie peuvent passer au travers de ce filtre, avant d'être focalisés sur la fenêtre d'entrée.
- 3- Multiplicateur d'électrons qui permet de focaliser les électrons, provenant de l'analyseur, et d'amplifier le signal .



**Figure. I.9.** Schéma représentatif de l'analyseur à miroirs cylindriques.



## I.4.1. Caractérisation *ex situ*

La microscopie électronique en transmission (Transmission Électronique Microscopie TEM) est une technique permettant de caractériser non seulement la structure locale des couches minces épitaxiées mais aussi d'étudier les défauts cristallins et la qualité de l'interface.

### I.4.1.1. Microscope électronique en transmission TEM

#### I.4.1.1.1 Description et principe de fonctionnement

Le microscope électronique à transmission est un outil pour l'étude des films minces obtenus par épitaxie (figure.I.10). Un faisceau d'électrons est comparable à un rayonnement électromagnétique, donc la longueur d'onde est définie par la relation suivante :

$$\lambda = \frac{h}{\sqrt{2m_0 \cdot e \cdot V \left(1 + \frac{e \cdot V}{2m_0 \cdot c^2}\right)}} \quad \text{I.1}$$

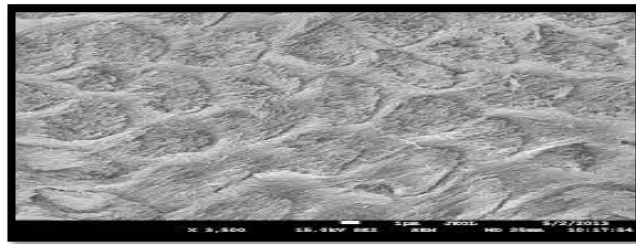
où  $h$ ,  $e$ ,  $m_0$  et  $m_0c^2=511\text{keV}$  sont respectivement la constante de Planck, la charge, la masse et l'énergie au repos de l'électron. Cette relation donne la longueur d'onde des électrons à partir de la tension d'accélération.



**Figure.I.10.** Image d'un microscope électronique en transmission.

Un faisceau d'électrons à haute tension (300KV), est émis par un canon. Les lentilles électromagnétiques sont utilisées pour focaliser le faisceau d'électrons sur l'échantillon. En traversant l'échantillon et les atomes qui

le constituant, le faisceau d'électrons produit différentes sortes de rayonnements. En générales seuls les électrons transmis sont alors analysés par le détecteur, qui traduit le signal en image contrastée (Figure.I.11).

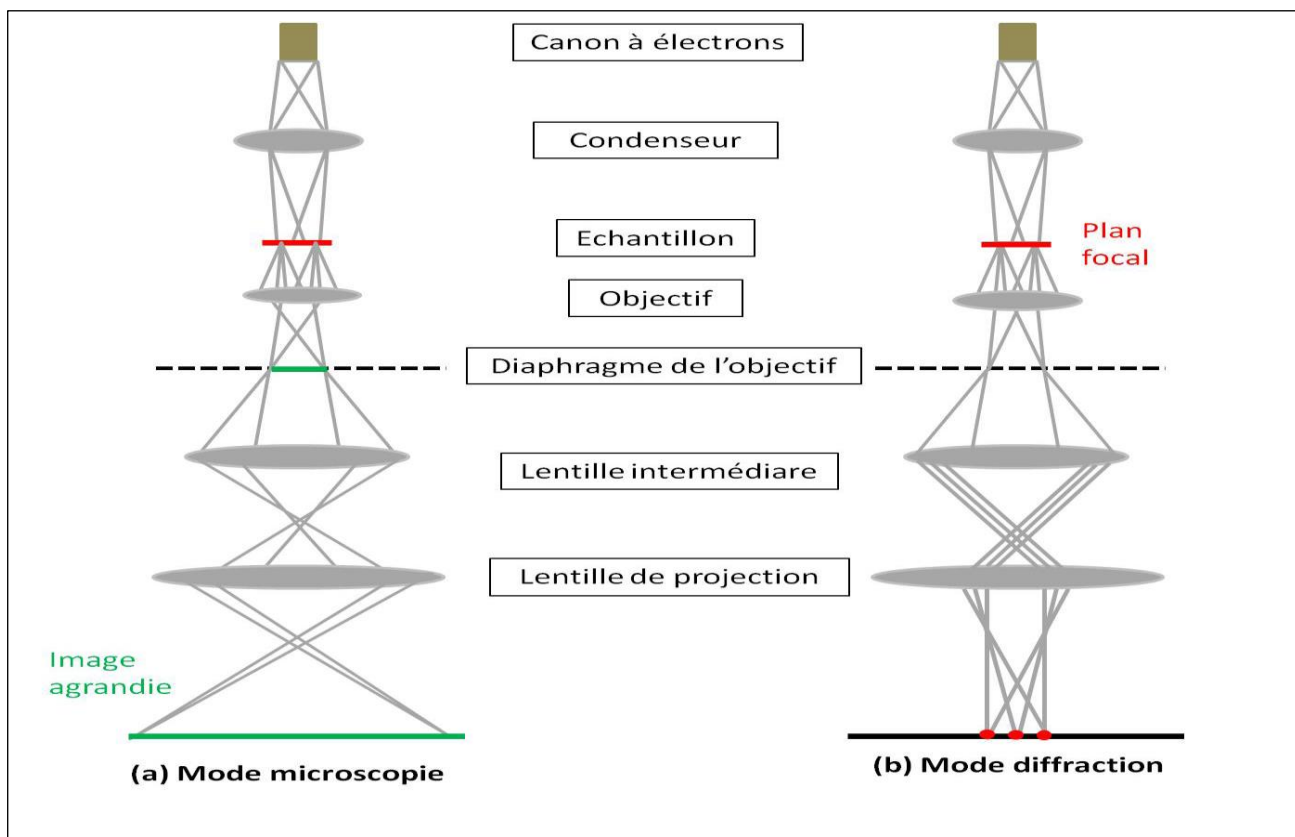


**Figure.I.11.** Image de la surface d'un matériau obtenue par un TEM

Pour sonder l'échantillon, puis, pour faire apparaître la structure du matériau, le faisceau d'électrons doit être émis de façon qu'il passe par une série de lentilles définies comme suite :

- 1- Les lentilles condenseurs
- 2- Le système objectif
- 3- Les lentilles intermédiaires et de projection

Ces lentilles forment ainsi trois zones principales qui se distinguent par leurs rôles dans la formation de l'image. Un schéma du principe de fonctionnement est représenté sur la (figure.I.12)



**Figure.I.12.** Principe de fonctionnement de TEM (a) en mode microscopie (b) en mode diffraction [8].

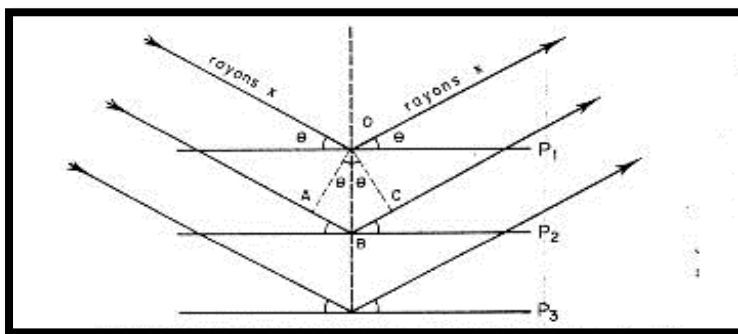
### I.1.1.2. Diffraction des rayons X

La diffraction des rayons X est une méthode de caractérisation structurale dans le domaine des matériaux. Cette méthode de caractérisation ne s'applique qu'aux matériaux cristallisés (mono ou poly cristallins). Cette étude a pour but de :

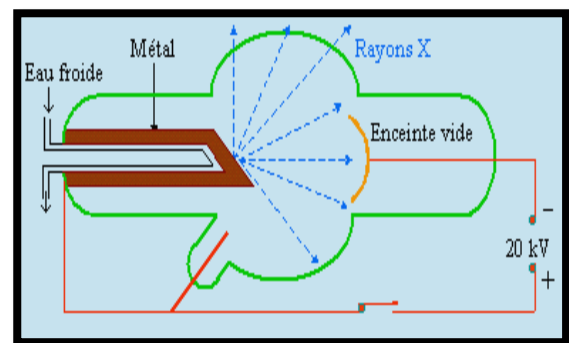
- ✓ Préciser la structure des couches minces
- ✓ Mesurer les paramètres de maille et la taille des cristallites
- ✓ Permettre d'examiner l'état de contrainte des dépôts.

Le principe repose sur la diffraction des rayons X monochromatiques par les plans atomiques des cristaux du matériaux étudié (figure.I.13).

Les rayons X incidents sont diffractés par l'échantillon et récoltés par le détecteur du photon. Ce détecteur compte le nombre de photons reçus en fonction de l'angle  $2\theta$ . La longueur d'onde des rayons X dépend de la nature de l'anticathode utilisée pour leur génération (Figure I.14).



**Fig.I.13.** Schéma représentant le principe de la diffraction des RX par les plans réticulaires d'indice (h,k,l)



**Figure.I.14.** Schémas représentant une anticathode utilisée pour la production des rayons X

### I.4.1.1. Diffractomètre RX

Le diffractomètre représenté sur la figure (1.15) comprend :

- ✓ Tube à rayons X monochromatique (Figure.I.14),
- ✓ Support d'échantillon
- ✓ Détecteur des photons
- ✓ Goniométrie sur lequel se déplace le détecteur des rayons X et un cercle de focalisation sur lequel le rayonnement diffusé est concentré avant la détection.



**Figure.I.15.** Diffractomètre utilisé pour la diffraction des rayons X

### I.4.1.2. Diffraction rasante

La diffraction rasante est utilisée dans le cas des couches minces. La source est fixée à une très faible inclinaison (angle  $< 1$  degré) et le détecteur est animé d'un mouvement de rotation (angle  $2\theta$ ). Un balayage des angles est alors effectué. Lorsqu'un angle, correspondant à une famille de plans (hkl), satisfait la relation de Bragg, le détecteur enregistre une augmentation de l'intensité diffractée. Une fois les photons détectés, le compteur les transforme en charge électrique, puis ils sont amplifiés par un système électronique. Le signal électrique est envoyé vers un ordinateur qui donne l'allure du spectre avec les différents pics de diffraction. La diffraction a lieu seulement dans les matériaux cristallins et lorsque la relation de Bragg est vérifiée [9]

$$2d \sin\theta = n\lambda \quad \text{I.2}$$

Avec  $d$  : Distance inter réticulaire, c'est-à-dire distance séparant deux plans consécutifs d'indice (hkl) ;  $\theta$  : Angle d'incidence des rayons X sur la surface des matériaux étudiés ;  $n$  : Ordre de la diffraction ;  $\lambda$  : Longueur d'onde du faisceau de rayons X.

Ainsi, pour obtenir la valeur des différentes distances inter-réticulaires, il suffit de déterminer les valeurs de  $\theta$  pour lesquelles le matériau diffracte.

# Chapitre II

## Généralités sur les couches minces

### II.1. Introduction

Une couche mince d'un matériau donné est un empilement d'atomes déposé sur un autre matériau appelé "substrat" dont l'une des dimensions, l'épaisseur, a été fortement réduite (Figure.II.1).

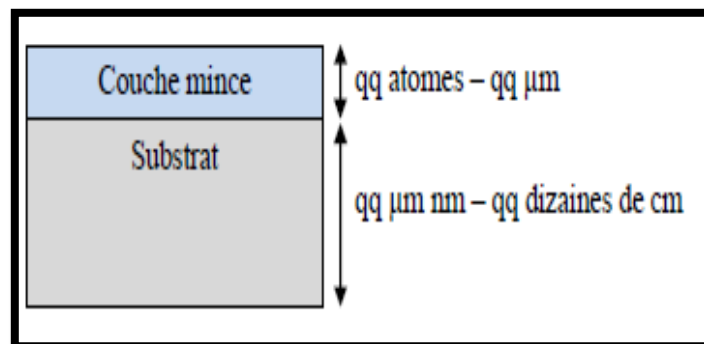


Figure.II.1. Dépositions d'une couche minces sur un substrat

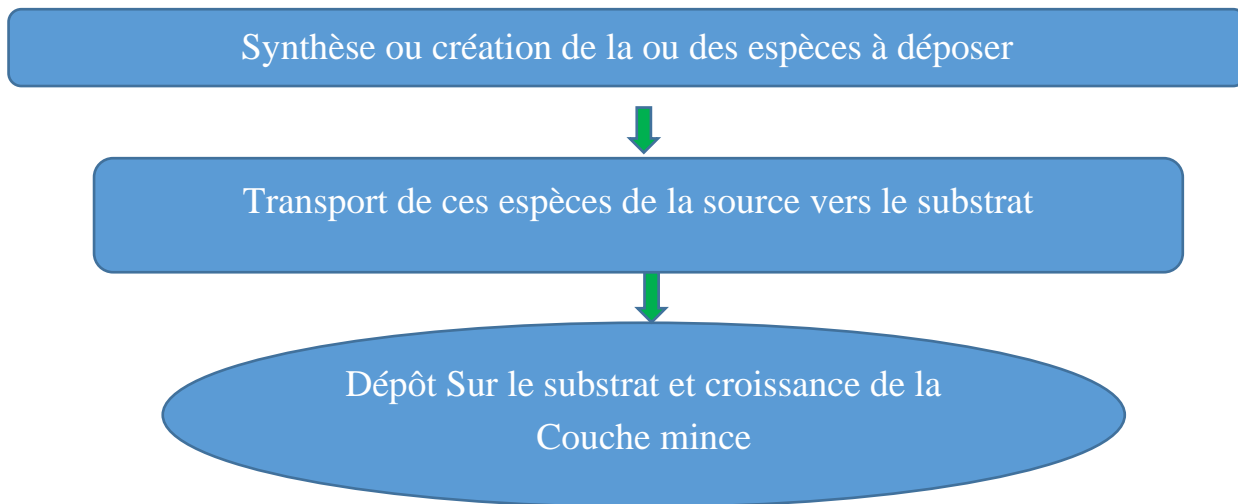
Une couche mince peut être considérée comme étant bidimensionnelle (2D). Cette faible distance entre les deux surfaces limites entraîne une perturbation des propriétés physiques, telle que la réflexion optique et la conductivité électrique.

### II.2. Généralités sur les couches minces

La différence essentielle entre le matériau à l'état massif et celui en couches minces est liée au fait que dans l'état massif on néglige généralement le rôle des limites dans les propriétés, tandis que dans une couche mince ce sont au contraire, les effets liés aux limites des surfaces qui sont prépondérants.

Il est assez évident que plus l'épaisseur sera faible plus cet effet de bidimensionnelle sera important. En revanche, lorsque l'épaisseur d'une couche mince dépassera un certain seuil, l'effet d'épaisseur deviendra minime et le matériau retrouvera les propriétés bien connues du matériau massif [1].

La seconde caractéristique importante d'une couche mince est sa dépendance de la structure du substrat sur lequel est épitaxiée et du processus de dépôt [13]. Le processus de dépôt d'une couche mince s'effectue en trois étapes, comme le montre le diagramme de la figure II.2. schématisé ci-dessous.






**Figure. II.2.** Les étapes de réalisation du processus de fabrication des couches minces

### II.3. Application des couches minces

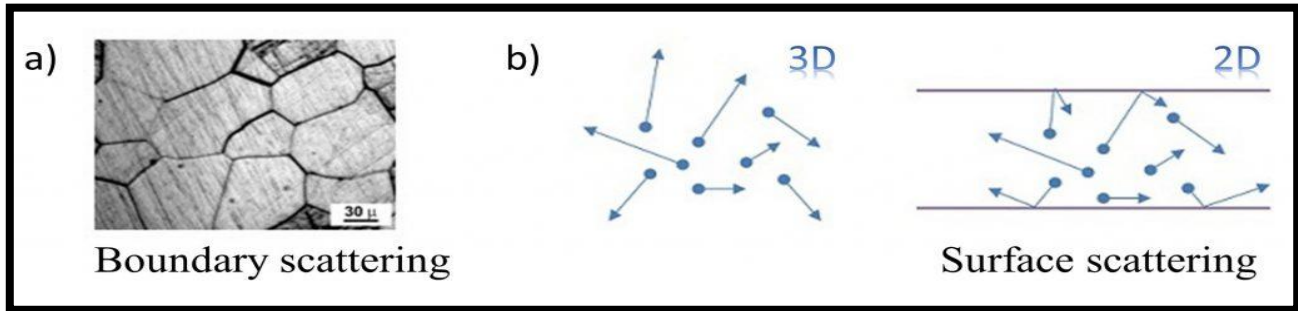
La modification de la surface d'un substrat permet de coupler les propriétés du matériau de cœur (ou substrat) et du matériau de surface : ce dernier peut être soit directement issu du substrat, dans le cas d'un traitement de surface, ou d'un matériau d'apport pour un revêtement de surface. La modification de la surface d'un substrat apporte une ou plusieurs propriétés physicochimiques supplémentaires à ce dernier (résistance à la corrosion, isolation électrique...) et permet de réduire le coût de fabrication tout en ayant les propriétés physico-chimiques de surface requises. Rappelons que les premières modifications de surface avaient un but esthétique (peinture, plaquage d'or, émaillage...) tandis que des applications techniques concernaient la métallurgie (cémentation, nitruration...)[14] . Au cours du XXème siècle, des applications plus pointues se sont diversifiées dans les domaines résumés sur le tableau II.1.



Tableau. II.1. Domaine d'application des couches minces

Domaine	Des exemples	Image
<b>Microélectronique</b>	elle a pu se développer à partir des années 1960 grâce à la mise en œuvre de couches de plus en plus minces conductrices ou isolantes, et on peut les trouver sous types de couche passives (contact électronique), jonction PN, diode, transistor, matériau piézoélectrique, lampe LED, supraconducteur.	 <p>Couches minces conductrice</p>
<b>Optique</b>	tout en conservant les applications esthétiques, les applications optiques des couches ont permis de développer des capteurs de rayonnements plus efficaces, comme les couches <u>anti-reflet</u> dans les cellules solaires, tain du miroir, traitement <u>anti-reflet</u> des objectifs d'appareil photo, photo détection, affichage-écrans plats, applications ophtalmiques, guide optiques (contrôles d'énergie – architecture, véhicules, conversion d'énergie...).	 <p>Capteur de rayonnement</p>
<b>Mécanique</b>	revêtements tribologiques (lubrification <u>sèche</u> ; résistance à l'usure, l'érosion, l'abrasion; barrières de diffusion).	 <p>Revetemnt tribologiques</p>
<b>Mécanique</b>	revêtements tribologiques (lubrification <u>sèche</u> ; résistance à l'usure, l'érosion, l'abrasion; barrières de diffusion).	 <p>Revetemnt tribologiques</p>
<b>Chimie</b>	les principales applications des revêtements de surface sont orientées vers une meilleure tenue à la corrosion par la création d'un film étanche (résistance à la corrosion), capteur à gaz, revêtements catalytiques, couches protectrices.	 <p>Bande anti corrosion</p>

Certains phénomènes comme la diffusion aux limites de la surface peut réduire les propriétés de transport ou entraîne un changement de moment dipolaire magnétique entre 2D (couches minces) et 3D (matériaux massifs) (Figure.II.3). De même la conductivité thermique et électrique des matériaux en couches minces est généralement plus faible que celle de leurs homologues massifs, parfois de façon spectaculaire.



**Figure.II.3.** Changement des propriétés microscopique magnétique entre un matériau massif et une couche mince.

De telles réductions de la conductivité thermique se produisent généralement pour deux raisons fondamentales :

1. Par rapport aux monocristaux, de nombreuses technologies de synthèse de couches minces entraînent plus d'impuretés, de désordre et de limites de grains, ce qui tend à réduire la conductivité thermique.
2. Même une couche mince atomiquement parfaite est censée avoir une conductivité thermique réduite en raison de la diffusion aux limites : des fuites de phonons et des interactions connexes. Ces deux mécanismes de base affectent différemment le transport dans le plan et le transport transversal (anisotropie).

## II.4. Mécanisme de formation d'une couche mince

La formation d'une couche mince s'effectue par une combinaison de trois processus qui sont :

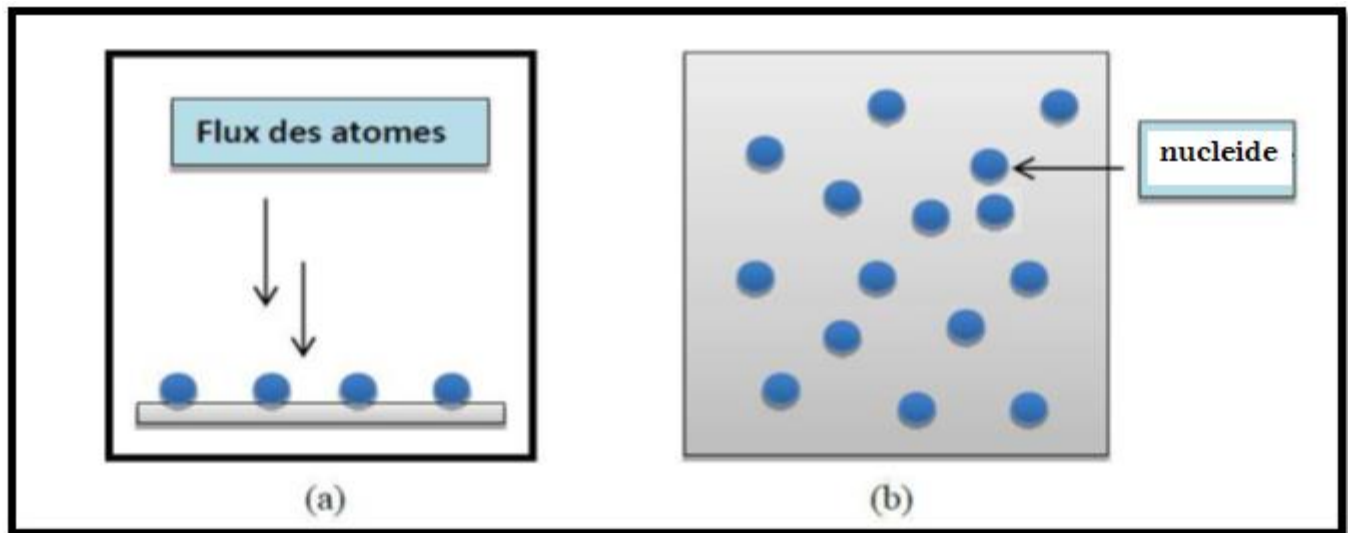
- 1) Nucléation
- 2) Coalescence
- 3) Croissance

### II.4.1. Nucléation

Les espèces pulvérisées arrivant sur le substrat sont physiquement adsorbées sur la surface du substrat. Ces espèces ne sont pas thermodynamiquement stables et se déplacent sur toute la surface du substrat. Dans cet



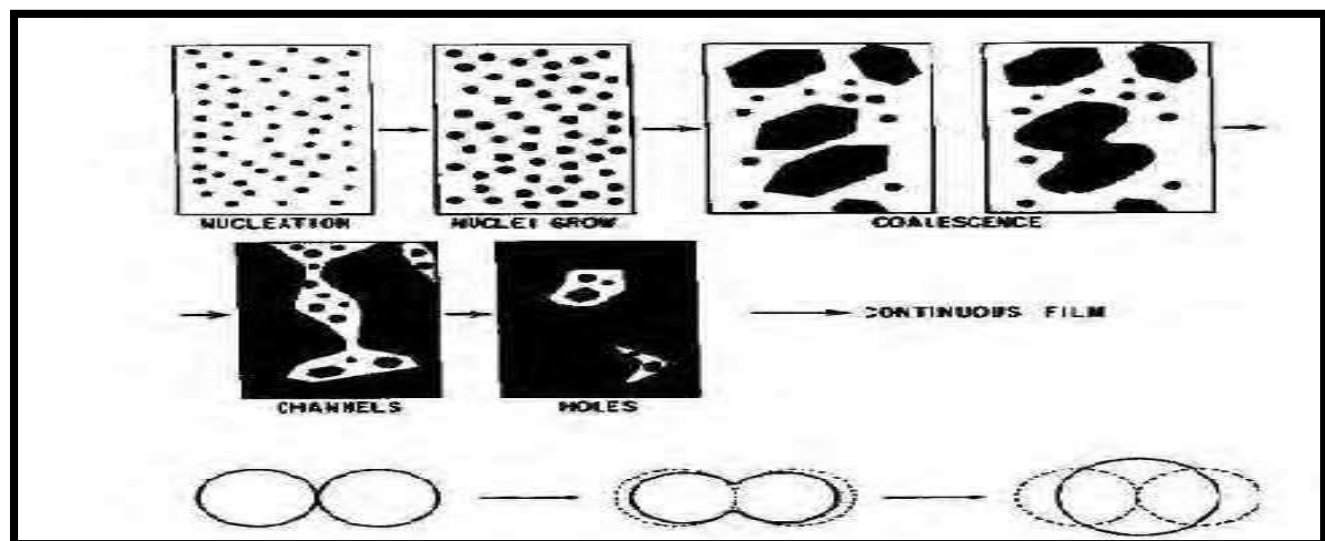
état, elles interagissent entre elles et forment ce que l'on appelle de "**clusters**" ou "**amas**" appelés également nucléide, parfois instables, les espèces physisorbées tendent à se désorber. Sous certaines conditions de dépôt, elles entrent en collision avec d'autres espèces adsorbées et commencent à croître. Après avoir atteint une taille critique, ces clusters deviennent thermodynamiquement stables (Figure.II.4)



**Figure II.4.** Représentation de l'étape de la nucléation des couches minces, (a) arrivés des atomes sur un substrat, (b) morphologie du substrat [16].

## II.4.2. Coalescence

Le mécanisme de **coalescence** est indiqué sur la figure II.5. Il conduit à une augmentation de la surface recouverte par le mécanisme de germination ou de coalescence avec les îlots déjà formés. Ce mécanisme est responsable de l'augmentation de la taille des grains avec l'épaisseur du film [17].



**Figure II.5.** Étapes de germination et de coalescence lors de la croissance cristalline en couches minces [13.]

### II.4.3. Croissance cristalline

Dans la dernière étape de coalescence des amas, la surface du substrat passe du type discontinu au type continu [15], comme le montre la figure II.6.

Peu à peu, une couche continue se forme lorsque les trous et les canaux se remplissent. et l'opération se finit par la formation d'une couche mince qui couvre la majorité du substrat. [15].

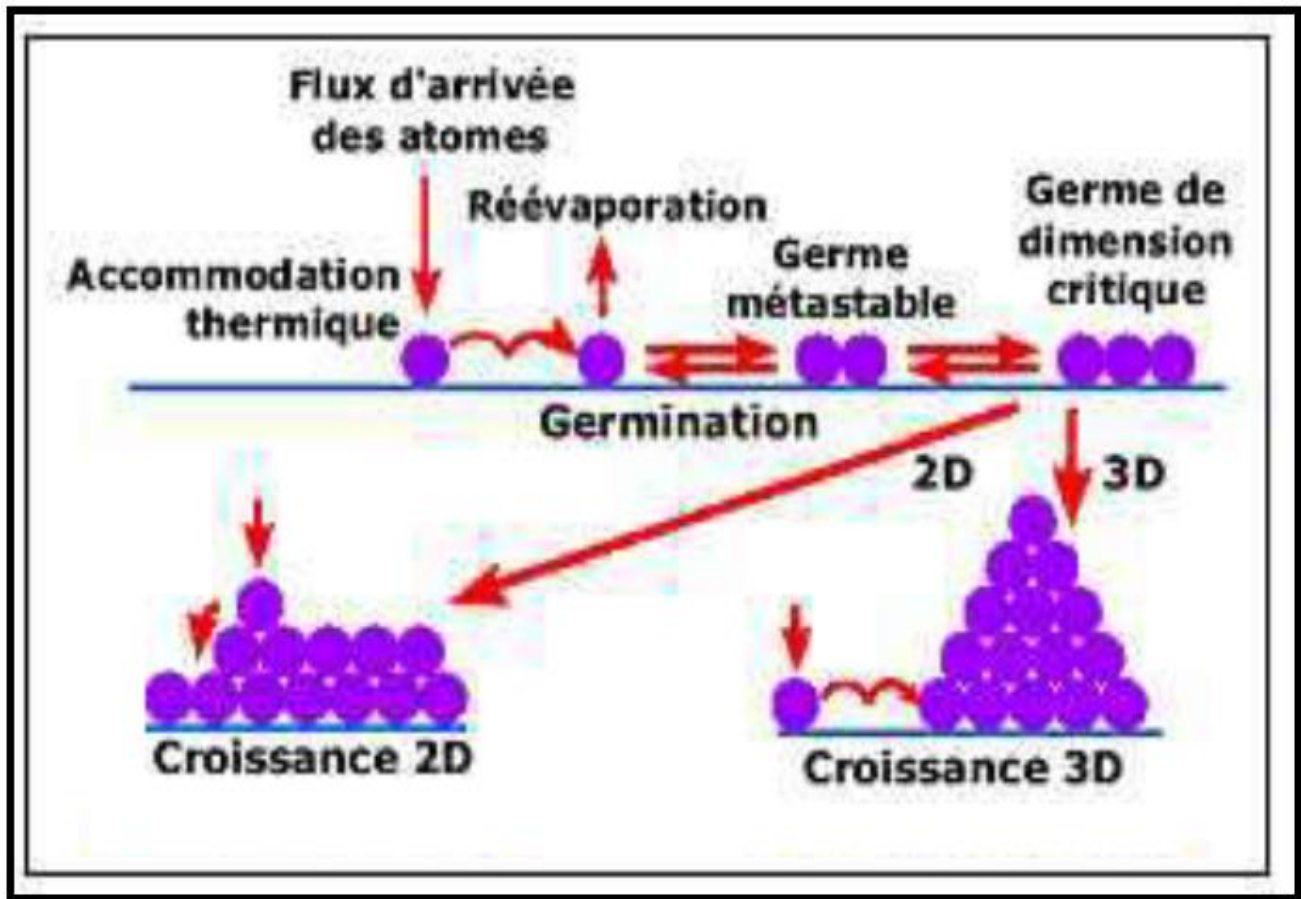


Figure II.6. Les différentes étapes et modes de croissance d'une couche mince [18].

## II.5. Méthodes de dépôt des couches minces

Les techniques utilisées pour le dépôt des couches minces peuvent être divisées en deux groupes basés sur la nature du processus physique ou chimique du dépôt (figure.II.7). Dans notre description, nous nous intéressons essentiellement aux méthodes physiques.

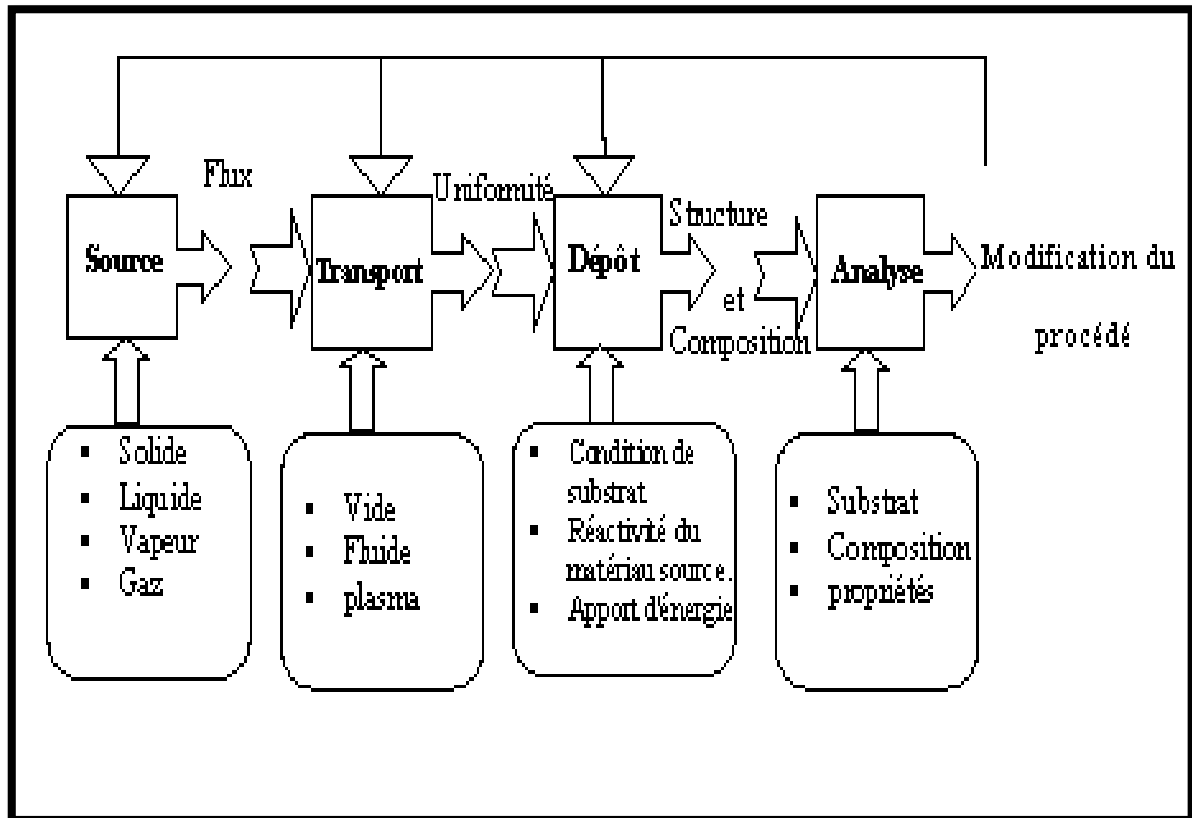


Figure II.7. Diagramme des étapes du procédé de fabrication de couches minces.

## II.5.1. Méthodes physiques :

Ces méthodes incluant le dépôt physique en phase vapeur dite "PVD" (Physical Vapor Deposition) sont :

1. Pulvérisation Cathodique "Sputtering".
2. Épitaxie par jet moléculaire "MBE"
3. Ablation laser,
4. Évaporation sous vide.

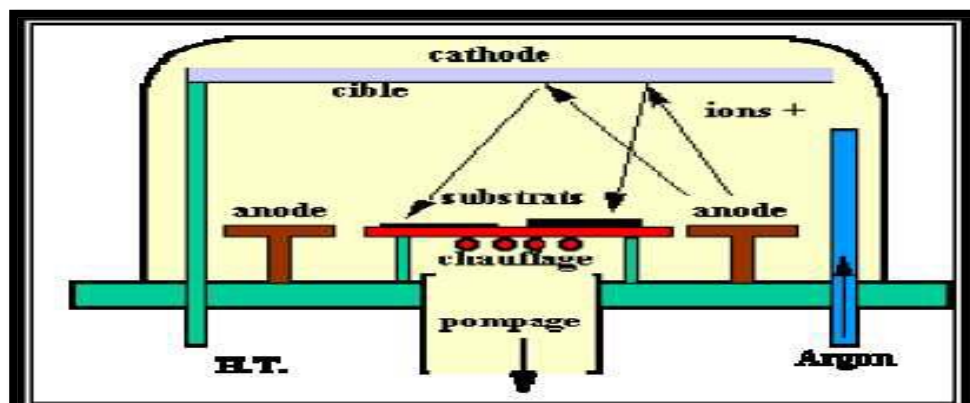
### II.5.1.1. Pulvérisation cathodique

Le phénomène de **pulvérisation** a été observé en 1852 par Grove et Pulcker en étudiant la décharge électrique dans les gaz [19]. Sous l'effet de bombardement ionique, la cathode se pulvérise en se déposant sous forme de couche mince sur les parois de l'enceinte en verre. Aujourd'hui, le schéma simplifié d'un **pulvérisateur** consiste à placer, en configuration d'un condensateur plan, le substrat (anode) en face d'une plaque du matériau à déposer (cathode). Le milieu ambiant est rempli d'un gaz pulvérisant, généralement de l'argon. Quand la décharge électrique est établie entre les deux électrodes, la cathode (ou cible) est

constamment bombardée par les ions d'argon ce qui permet l'éjection d'atomes ou groupe d'atomes qui viendront se condenser sur le substrat pour constituer la couche mince.

Cette technique est connue sous le nom de pulvérisation cathodique et son développement a conduit à l'apparition de plusieurs types d'appareils qui permettent de pulvériser divers matériaux séparément ou simultanément [19]. Les dispositifs les plus utilisés sont :

- **pulvérisation diode en tension continue (ou DC)** qui se limite à la pulvérisation des matériaux conducteurs. La pulvérisation RF ou la tension DC est remplacée par une tension radiofréquence ( $f = 13.56$  MHz ou égale à ses harmoniques). Cette dernière méthode permet la pulvérisation des matériaux conducteurs et non conducteurs.
- **La pulvérisation peut avoir l'option magnétron** quand un champ magnétique est appliqué dans le plan de la cible, et perpendiculairement au champ de décharge et qui a l'avantage d'augmenter la vitesse de dépôt et d'uniformiser la couche mince.
- **La pulvérisation est dite réactive** si on ajoute un gaz additionnel (par exemple:  $N_2$ ,  $O_2$ ) pour synthétiser des nitrures ou oxydes de matériaux.



**Figure II.8 :** Bâti de dépôt par pulvérisation cathodique. Les ions Argon, créés par l'excitation haute tension, arrachent de la cible les composés à déposer sur la surface des substrats [13].

### II.5.1.2 Epitaxie par Jets Moléculaires

La technique MBE dont nous avons déjà parlé est une technique qui permet de réaliser des dépôts monocristallins. Cette technique a été développée pour la croissance des semi-conducteurs (Si, GaAs, CdTe, ZnSe,...) car elle permet de réaliser des homoépitaxies (matériau A sur support A) à basse température (400-600 °C pour Si) contrairement à la CVD (1000 °C). On élimine ainsi les problèmes de diffusion des dopants par exemple. En outre, les basses températures d'épitaxie impliquent des faibles vitesses (qq. Å/S) de

croissance afin de laisser le temps aux atomes arrivant à la surface de migrer par diffusion de surface vers des sites cristallographiques. L'épitaxie en phase liquide est aussi une variante de la MBE [20].

### II.5.1.3 Ablation Laser

Cette technique est proche de l'Épitaxie par Jets Moléculaires (MBE) puisqu'elle ne diffère de cette dernière que par le moyen d'évaporer le matériau à déposer. On utilise en effet un faisceau laser, de forte intensité, envoyé sur une cible constituée du matériau que l'on veut déposer sur le substrat (figure II.10). La première différence avec la MBE est qu'on évapore des amas d'atomes.

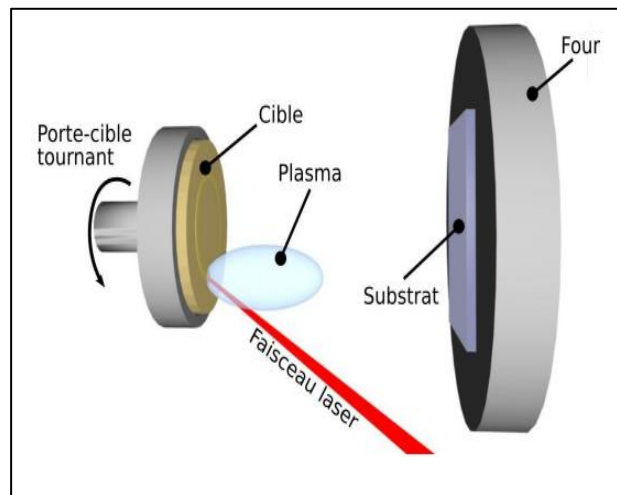


Figure II.9. Principe du dépôt de couches minces par ablation laser.

L'avantage de cette technique réside a priori dans sa capacité à conserver la stœchiométrie d'un alliage constituant la cible. En outre, il existe une deuxième différence importante entre l'ablation laser et la MBE concernant l'énergie des atomes se déposant sur le substrat. Dans la technique MBE, l'énergie d'activation des mécanismes intervenant dans le processus de croissance est purement thermique, donc très faible (de l'ordre de 0,17 eV pour 1000 °C) alors qu'elle est beaucoup plus importante en ablation laser [21].

### II.4.1.4 Évaporation sous vide

L'évaporation d'un matériau à déposer est obtenue en le chauffant par l'un des moyens suivants :

- 1/ effet Joule,
- 2/ induction (couplage d'un générateur haute fréquence),
- 3/ canon à électrons,
- 4/ faisceau laser
- 5/ ou arc électrique (Figure.II.11).

Dans le but d'augmenter la vitesse du film mince, l'évaporation est effectuée sous un vide poussé (pression de l'ordre de  $10^{-4}$  Pa) [22].

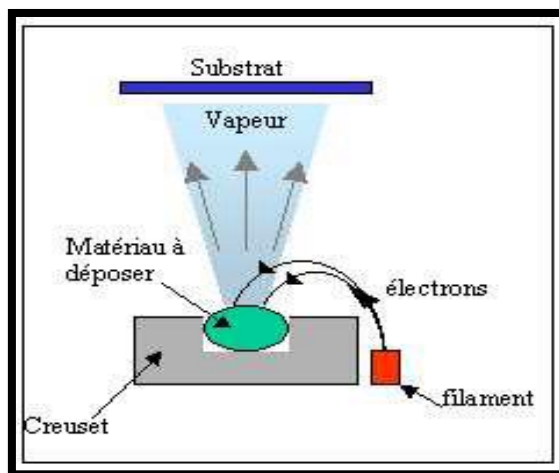


Figure II.10. Schéma conventionnel d'une évaporation thermique.

Lorsque la pression n'est pas suffisamment basse, les dépôts sont peu adhérents et souvent amorphes. Comme le flux de vapeur est localisé et directionnel, il est souvent nécessaire de donner au substrat un mouvement de rotation ou de translation par rapport à la source d'évaporation, de manière à réaliser un dépôt homogène et d'épaisseur uniforme. Les meilleurs résultats sont obtenus sur des surfaces pratiquement perpendiculaires au flux de vapeur [23]. D'une manière générale, les principaux problèmes rencontrés lors d'une évaporation sont :

- ✓ Dissociation des oxydes.
- ✓ Réaction des matériaux à évaporer avec ceux avec lesquels ils sont en contact.
- ✓ Dégazages, la décomposition, les micro-explosions des matériaux à évaporer.
- ✓ Difficulté d'obtenir des couches d'alliages ayant la même composition que l'alliage de départ.

L'évaporation reste, toute fois, une méthode particulièrement appréciée car on élabore ainsi des matériaux très purs et d'autant plus purs que la pression est faible, ce qui est le cas pour le procédé d'épitaxie par jets moléculaires. Cependant, cette méthode, d'évaporation sous vide, ne convient pas à la fabrication de films hors équilibre thermodynamique.

# Chapitre III

## Tirage des nombres aléatoires

### III.1. Introduction

Les méthodes de Monte Carlo sont des méthodes statistiques basées sur le tirage des nombres aléatoires, suivant des lois de probabilité qui décrivent les processus physiques mis en jeu. La croissance cristalline, la nature stochastique des processus d'émission, de transport et de détection font que les méthodes de Monte Carlo soient bien adaptées aussi bien à la physique nucléaire qu'aux sciences des matériaux[24].

Devant la complexité croissante des expériences de physique des matériaux et de croissance cristalline, il est nécessaire d'avoir des logiciels de simulation robustes et performants. Le compilateur Dev C++ a été développé pour répondre à cette demande.

### III. 2. Définition d'un générateur de nombres pseudo-aléatoires

Dans beaucoup d'applications informatiques (la simulation en particulier), il est nécessaire de tirer des nombres au hasard pour initialiser différents algorithmes. Or, toutes les informations que fournit un ordinateur sont calculées, et le hasard n'a pas de sens en informatique. Alors, pour nous fournir des données aléatoires, l'ordinateur doit **simuler** le hasard. On parlera alors de données **pseudo-aléatoires** pour souligner leur différence par rapport aux véritables suites de variables aléatoires indépendantes. Il existe deux types de générateurs :

- ✓ **Les générateurs à sorties imprédictibles** : ils génèrent des nombres pseudo-aléatoires dont on ne peut prévoir la valeur,



- ✓ **Et les générateurs à sorties prédictibles** : dans ces générateurs, on initialise (on dit nourrir) l'algorithme à partir d'un nombre connu appelé « graine » et le générateur produira toujours la même suite s'il est initialisé avec la même valeur.

### III.3. Source de nombres pseudo-aléatoires :

Les simulations Monte Carlo ont pour but d'imiter le comportement d'un système réel en toute rigueur. La simulation de chaque étape d'un processus physique (génération d'une particule, interaction de la particule) nécessite un tirage d'une série de nombres pseudo-aléatoires. La source de ces nombres est un procédé algorithmique qui produit des suites de nombres pseudo-aléatoire (de bonne qualité) possédant les propriétés suivantes :

- a) Les séquences des nombres pseudo-aléatoires ne doivent pas présenter des corrélations (exemples : avoir que des chiffres pairs ou que des chiffres impairs, parmi d'autres corrélations).
- b) La reproductibilité : lorsqu'on teste des programmes de simulations, il est nécessaire de les répéter.
- c) La période doit être suffisamment longue : la répétition ne doit avoir lieu qu'après une génération d'une très grande séquence de nombres aléatoires, idéalement le nombre aléatoire ne devrait pas se répéter.
- d) La séquence de nombres pseudo-aléatoires doit être uniforme.
- e) La vitesse de calcul doit être minimale.

On s'attend cependant à ce qu'un procédé de génération algorithmique correct produise des suites de nombres qui « ressemblent » d'assez près à des nombres aléatoires indépendants et uniformes sur  $[0,1]$  pour que la validité des résultats obtenus par simulation soit assurée.

### III.4. Générateurs des nombres pseudo-aléatoires (GNPA)

Le générateur (GNPA) fonctionne en itérant une fonction « f » à partir d'un état initial  $S_0$  appelé « graine », choisi par l'utilisateur. Le choix de la graine  $S_0$  détermine entièrement le comportement de la suite de nombres pseudo-aléatoires produite par le générateur. La majorité des générateurs utilisés sont basés sur le principe de la graine.

#### A) Générateur de Von Neumann (1946)

Le générateur de Von Neumann, proposé en 1946, consiste à élever le nombre  $S_n$  au carré et à ôter les premiers et les derniers chiffres, de manière symétrique, de sorte à ce que le nombre obtenu soit compris entre 0 et 9999, par exemple :  $S_n = 1315$  alors  $(S_n)^2 = (1315)^2 = 1729225$  donc  $S_{n+1} = 292$ .



## B) Générateurs à congruence linéaire (1948)

Les générateurs à congruence linéaire ont eu beaucoup de succès. L'idée est d'appliquer une transformation linéaire suivie d'une opération de congruence. Dans le cas discret :

$$S_n = f(S_{n-1}) = a \cdot S_{n-1} + b \pmod{m} \quad \text{III.1}$$

avec  $S = \{0, \dots, m-1\}$ .

Si on souhaite se ramener à l'intervalle  $U = [0,1]$ , on appliquera  $S_n/m$ . comme par exemples :

⇒ **Générateur connu sous le nom « Standard Minimal »** : Il s'agit du générateur défini avec les paramètres  $a = 16807$ ,  $b = 0$ ,  $m = 231 - 1$ .

⇒ **Générateur dit « RANDU »** : qui prend pour paramètres  $a = 65539$ ,  $b = 0$ ,  $m = 231$ .

Ces méthodes ayant toutefois montré leurs limites, d'autres générateurs pseudo-aléatoires ont été générés selon les besoins de chaque simulation. D'autres types de congruence peuvent être considérés. Par exemple, la fonction « rand () » de la bibliothèque standard de C/C++ utilise par défaut une congruence de type polynomial. Elle est supposée produire à chaque appel un nombre entier pseudo-aléatoire uniformément distribué entre 0 et la constante RAND\_MAX (égale par exemple à  $2^{32}$  dans certaines système), et indépendant des nombres produits par les appels précédents. Cette fonction est celle utilisée dans notre travail de master qui consiste à simuler un phénomène physique via un générateur des nombres pseudo-aléatoires.[25]

## III.5. Méthodologie de tirage des nombres pseudo-aléatoires en C++

Dans ce paragraphe, avant la description de la méthodologie utilisée, dans notre programme de simulation Monte Carlo cinétique, pour le tirage des nombres pseudo-aléatoires à partir de fonctions intégrées dans le langage C++ que nous avons choisi, nous allons présenter le compilateur utilisé pour la réalisation de notre travail de simulation.

### III.5.1. Présentation de l'environnement Dev C++ 5.11

Dev C++ 5.11 est un environnement de développement intégré permettant de programmer en C++ sous le système d'exploitation Windows[21]. En cliquant sur l'icône Dev C++ (Figure.III.1.A) une fenêtre apparaît pour permettre à son utilisateur d'ouvrir un **fichier source d'extension .cpp** (Figure III.2B)



Figure.III.1.A : Icône de DEV.C++

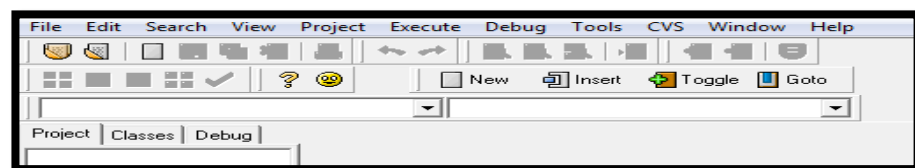


Figure.III.1.B : Environnement de DEV C++

### III.5.2. Méthode d'utilisation de Dev C++ 5.11 pour l'écriture du GN-PA

L'ouverture d'un nouveau fichier d'extension .cpp, pour l'écriture d'un nouveau programme, est faite en suivant le chemin illustré par la figures III.1.C.

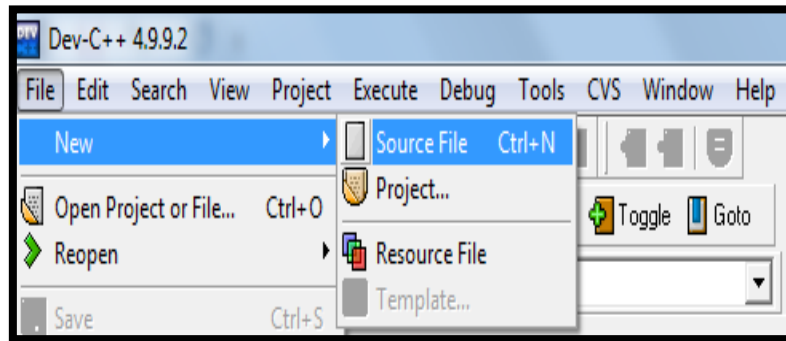


Figure.III.1.C : Chemin d'ouverture d'un nouveau fichier source (de programmation)

Pour ouvrir un programme déjà existant dans l'ordinateur, le chemin à suivre est visualisé par les figures III.1.D. et III.1.E, respectivement

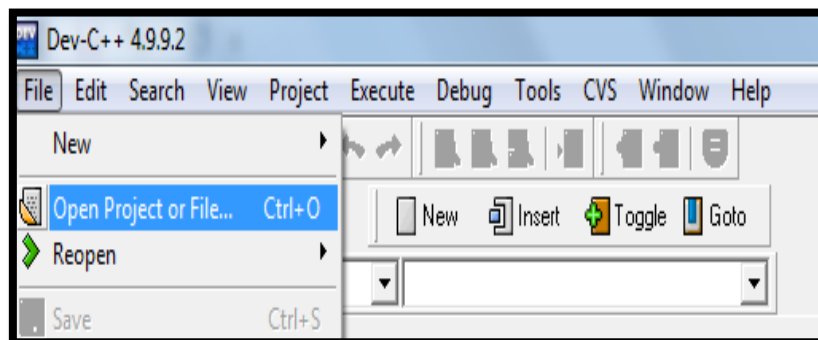


Figure.III.1.D: Chemin d'ouverture d'un fichier source (de programmation) déjà existant.

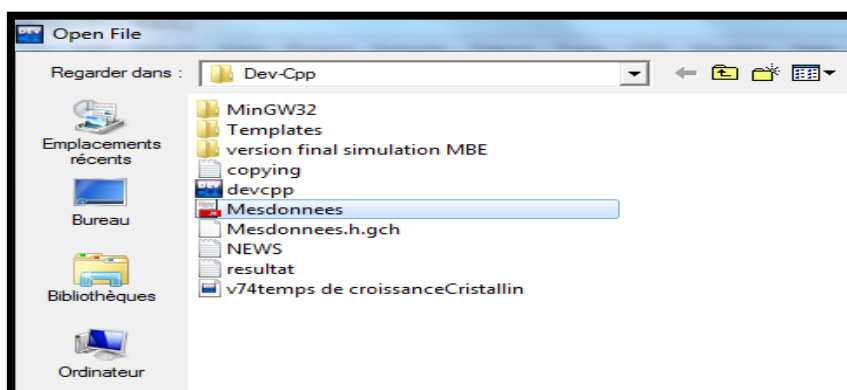
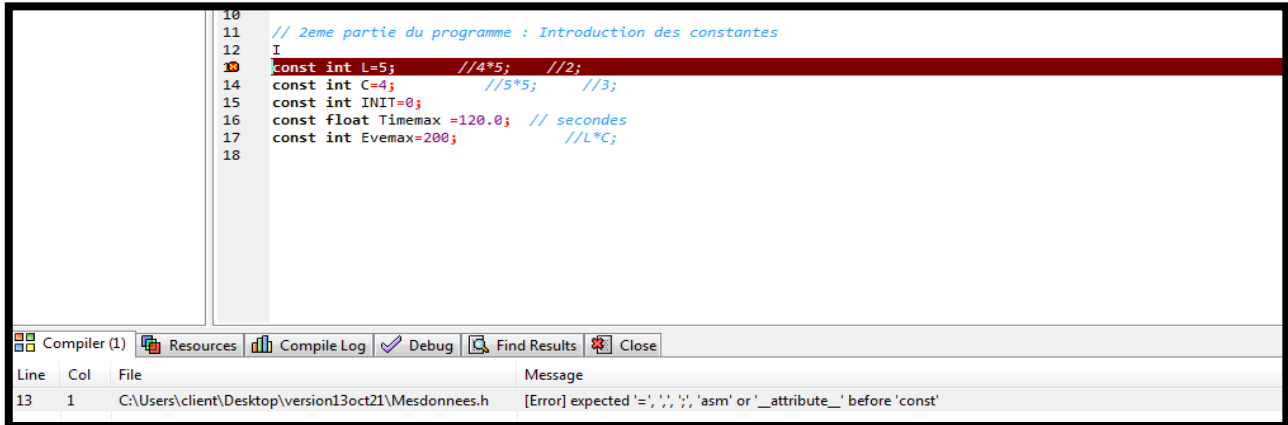


Figure.III.1.E : Chemin d'ouverture d'un fichier source (de programmation) déjà existant.

Les figures III.1. F et III.1.G visualisent, respectivement, un exemple de programme en C++ et le chemin correspondant sur DEV-C++ pour sa compilation.



```
10 // 2eme partie du programme : Introduction des constantes
11 I
12
13 const int L=5; //4*5; //2;
14 const int C=4; //5*5; //3;
15 const int INIT=0;
16 const float Timemax =120.0; // secondes
17 const int Evemax=200; //L*C;
18
```

Compiler (1) Resources Compile Log Debug Find Results Close

Line	Col	File	Message
13	1	C:\Users\client\Desktop\version13oct21\Mesdonnees.h	[Error] expected '=', ',', ';', 'asm' or '__attribute__' before 'const'

Figure.III.1.F : Exemple de programme en .cpp

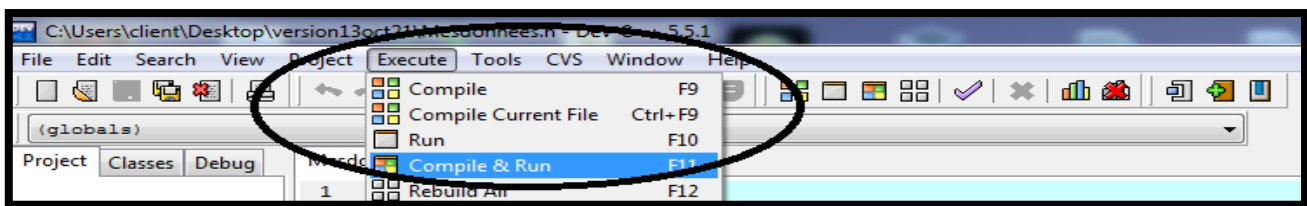


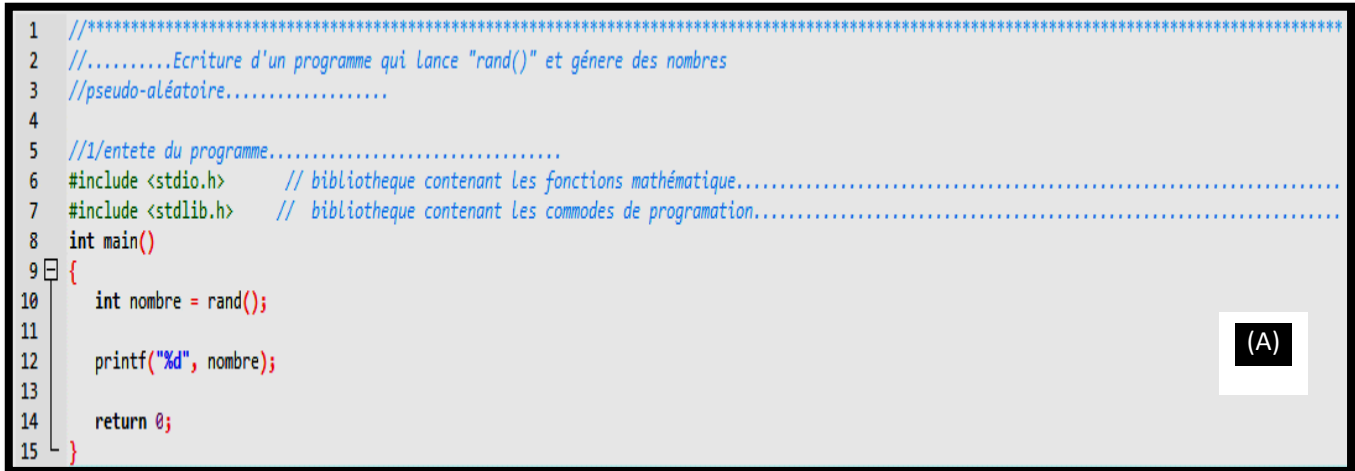
Figure.III.1.G : Chemin de compilation d'un programme en cpp

Le résultat d'exécution du programme apparaîtra sur une fenêtre noire dite « console »

### III.5.3. Application d'un code pour tirage des nombres aléatoires

Pour commencer, on inclue la bibliothèque standard "**stdlib.h**" pour pouvoir appeler des fonctions telle que « **rand ()** » et la bibliothèque "**stdio.h**". ces fichiers d'extension .h sont des bibliothèques du C++ nécessaires pour que le programme reconnait les commandes informatiques telle que **printf**, **for**,.... Un extrait du programme de simulation utilisant le générateur des nombres pseudo-aléatoires « **rand** » est visualisé sur la figure III. 2, ainsi que sa fenêtre d'exécution.

```
1 //*****
2 //.....Ecriture d'un programme qui lance "rand()" et génère des nombres
3 //pseudo-aléatoire.....
4
5 //1/entete du programme.....
6 #include <stdio.h> // bibliotheque contenant Les fonctions mathématique.....
7 #include <stdlib.h> // bibliotheque contenant Les commodes de programmation.....
8 int main()
9 {
10     int nombre = rand();
11
12     printf("%d", nombre);
13
14     return 0;
15 }
```



```
41
-----
Process exited with return value 0
Press any key to continue . . .
```



**Figure III.2.** (A) Programme écrit en langage C++ (sous Windows) pour générer **un seul nombre pseudo-aléatoire** Avec la fonction {rand () } de la bibliothèque standard de ce langage, (B) la fenêtre de l'exécution du programme (A).

Après l'exécution du programme, le résultat était limité à un seul chiffre qui est « 41 ». Maintenant, si on ferme la console et on recommence l'exécution du même programme, on voit que la valeur obtenue précédemment ne change pas, bien que le générateur « rand » a été exécuté pour la deuxième fois. Cela s'explique par le fait que la fonction « rand » ne génère des nombres pseudo-aléatoires que si le programme fait appel à cette fonction en plusieurs fois, dans le même programme, avant l'apparition de la console qui visualise le résultat final. Pour cela, on va utiliser dans le programme de la figure III.2.(A) une boucle "for" pour générer une série de valeurs pseudo-aléatoires et ainsi changer la valeur précédente « 41 » qui s'est répétée à chaque exécution. La figure III.3 (A) montre le programme permettant de générer sept valeurs pseudo-aléatoires à travers l'appel répétitif de la fonction rand (). Le résultat est exposé sur la figure III.3 (B).

```

1 //*****
2 //.....Ecriture d'un programme qui lance "rand()" et génère des nombres
3 //pseudo-aléatoire.....
4
5 //1/entete du programme.....
6 #include <stdio.h> // bibliotheque contenant les fonctions mathématique.....
7 #include <stdlib.h> // bibliotheque contenant les commodes de programmation.....
8 //2/La boucle "for"
9 int main()
10 {
11     int i;
12
13     for (i = 0; i < 7; i++)
14         printf("%d\n", rand());
15
16     return 0;
17 }
18

```

(A)

```

41
18467
6334
26500
19169
15724
11478
-----
Process exited with return value 0
Press any key to continue . . .

```

(B)

**Figure III.3.** (A) Programme écrit en langage C++ (sous Windows) pour générer **sept nombres pseudo-aléatoires** avec la fonction `rand()`, (B) la fenêtre de l'exécution du programme (A).

Après l'exécution du programme de la (figure III.3 (A)), on remarque que les sept valeurs générées (figure III.3 (B)) ne se ressemblent et semble bien être des nombres pseudo-aléatoires recherchés. Mais cette série de nombre restera toujours la même à chaque exécution du programme de la figure III.3 (A).

Cela s'explique par le fait que les nombres pseudo-aléatoires que fournit la fonction (**rand**) sont calculés à partir d'une donnée « **seed** (en anglais) » qui veut dire « graine (en français) ». Si cette dernière n'est pas modifiée, la suite des nombres restera toujours la même. La solution est donc d'initialiser la fonction (**rand**) avec une valeur toujours différente, à chaque démarrage du programme, c'est-à-dire, à l'aide de la fonction « **srand()** ». Prenons l'exemple de la fonction **srand(122)** qui est nourrit avec la graine de valeur 122. Le programme et son exécution sont représentés sur la figure III.4 pour une seule valeur de nombre pseudo-aléatoire. Avec ce nouveau code de la figure III.4, nous avons obtenu une valeur différente (qui est « 437 »),

mais elle sera toujours La même à chaque exécution du programme. Cela s'explique par le fait que la fonction « `srand ()` » ne doit pas être initialisée avec la même graine (une valeur constante).

```

1 //*****
2 //.....Ecriture d'un programme qui lance "rand()" et genere des nombres
3 //pseudo-aléatoire.....
4
5 //1/entete du programme.....
6 #include <stdio.h> // bibliotheque contenant Les fonctions mathématique.....
7 #include <stdlib.h> // bibliotheque contenant Les commodes de programmation.....
8 int main()
9 {
10 //3/faire rentrer la fonction srand
11 srand(122); // 122 est la donnée seed
12 printf("%d", rand()); // rand renvoie un nombre calculé à partir de la donnée seed
13 return 0;
14 }

```

(A)

```

437
-----
Process exited with return value 0
Press any key to continue . . .

```

(B)

**Figure III.4.** (A) Programme en C++ (sous Windows) pour générer **un seul nombre pseudo-aléatoire** avec la combinaison des fonctions `srand` et `rand ()` de la bibliothèque standard de ce langage, (B) la fenêtre de l'exécution du programme (A).

La question qui se pose est la suivante : avec quelle variable peut-on initialiser la fonction " `srand ()` " ?

L'un des modèles les plus anciens et les plus influents utilisé en réponse à la question sus citée est la fonction « `time ()` » qui renvoie le nombre de secondes, de l'heure interne de l'ordinateur, à l'instant où elle est appelée par le programme. Cette fonction est répertoriée dans la bibliothèque « `time.h` » du langage C++.

```

1 //*****
2 //.....Ecriture d'un programme qui lance "rand()" et genere des nombres
3 //pseudo-aléatoire.....
4
5 //1/entete du programme.....
6 #include <stdio.h> // bibliotheque contenant Les fonctions mathématique.....
7 #include <stdlib.h> // bibliotheque contenant Les commodes de programmation.....
8 #include <time.h> // La fonction time
9 int main()
10 {
11 srand(time(NULL)); // Initialisation de la donnée seed
12 printf("%d", rand()); // rand renvoie un nombre calculé à partir de la donnée seed
13
14 return 0;
15 }
16

```

**Figure III.5.** Programme en C++ (sous Windows) pour générer **un seul nombre pseudo-aléatoire** avec la combinaison des fonctions `srand()`, `time(NULL)` et `rand ()` de la bibliothèque standard de ce langage.

Bien que la fonction `time ()` renvoie un entier, elle doit être initiée par un paramètre (graine). La valeur `NULL` (c'est une constante valant 0), est une des graines utilisées pour cette fonction. La figure 3.4 reproduit le code précédant en ajoutant les nouvelles fonctions telles que « `time (NULL)` ».

Après exécution du code représenté sur la figure III.5, nous remarquons que les nombres générés changent après chaque exécution, comme le montre la figure III.6



**Figure III.6.** Trois consoles représentant chacune une exécution du programme contenant les trois fonctions (`srand()`, `time()` et `rand()`).

### III.6. Tirage des nombre pseudo-aléatoire dans un intervalle $[0, N_{\max}-1]$

Dans le langage C++, "`rand ()`" nous donne un nombre pseudo-aléatoire entre 0 et `RAND_MAX`, dont la valeur dépend du système utilisé : Linux, Unix ou Windows. . Pour faire un tirage de nombre entier compris entre 0 et  $N_{\max} - 1$ , on utilise la fonction "modulo" notée « % ». Cette dernière nous donne le reste de la division du nombre aléatoire par  $N_{\max}$ .

Dans notre cas, l'écriture « `rand ()% 101` » nous donne un nombre compris entre 0 et 100. Si on veut éviter le zéro dans le tirage, on ajoutera « 1 » à la fin de l'écriture « `(rand ()% 100)+1` » comme le montre le code présenté sur la (figure III.6).

Pour le code source final, nous allons utiliser des entiers dans l'intervalle  $[Min \text{ et } Max]$  sur la figure III.7, dont les valeurs de `Min` et de `Max` seront choisies par l'utilisateur selon le type de simulation à réaliser.

```

1 //.....Ecriture d'un programme qui calcule le reste de la division entière
2
3
4 //1/entete du programme.....
5 #include <stdio.h> // bibliotheque contenant les fonctions mathematique.....
6 #include <stdlib.h> // bibliotheque contenant les commandes de programmation.....
7 #include <time.h> // la fonction time.....
8 int main()
9 {
10     int nombre = 0;
11     srand(time(NULL)); // Initialisation de la donnée seed
12
13     nombre = rand() % (100 + 1);
14     printf("%d", nombre); // rand renvoie un nombre calculé à partir de la donnée seed
15
16     return 0;
17 }
18

```

**Figure III.7.** Programme en C++ (sous Windows) pour générer un seul nombre pseudo-aléatoire dans l'intervalle  $[0,100]$  avec la combinaison des fonctions "`srand ()`," `time(NULL)`," `rand ()` "et "modulo", de la bibliothèque standard de ce langage.



```

1  /**.....Ecriture d'un programme (finale) qui génèrent des nombres entier (pseudo aléatoire)
2
3
4  //1/entete du programme.....
5  #include <stdio.h> // bibliotheque contenant les fonctions mathématique.....
6  #include <stdlib.h> // bibliotheque contenant les commodes de programmation.....
7  #include <time.h> // La fonction time.....
8  int main()
9  {
10     int nombre = 0;
11     const int MIN = 1, MAX = 100;
12     srand(time(NULL)); // Initialisation de la donnée seed
13
14     nombre = (rand() % (MAX + 1 - MIN)) + MIN; // MIN <= nombre <= MAX
15     printf("%d", nombre); // rand renvoie un nombre calculé à partir de la donnée seed
16
17     return 0;
18 }

```

**Figure III.8.** Programme en C++ (sous Windows) pour générer **un seul nombre pseudo-aléatoire dans l'intervalle [Min, Max]** avec la combinaison des fonctions "srand ()", "time(NULL)", "rand ()" et "modulo", de la bibliothèque standard de ce langage.

On rentre Maintenant la boucle "for" dans le programme qui situé dans la figure III.7. Pour générer toute une série de 300 Nombres P. Aléatoire compris entre [1-100].

```

3  //1ere partie du programme : Introduction des Bibliothèques
4
5  #include <stdio.h> /*bibliotheque contenant l'instruction printf , for ; int ... */
6  #include <conio.h> /*bibliotheque contenant l'instruction getch () */
7  #include <stdlib.h> /*bibliotheque contenant l'instruction rand et ran_max () */
8  #include <math.h> /*bibliotheque contenant l'instruction ln (Logarithme) */
9  #include <time.h>
11 int main()
12 {
13     int i, nombre = 0, Nbrcycle=300;
14     FILE *f0; //...Declaration de Fichier externe.....
15
16     const int MIN =1, MAX=100;
17     srand(time(NULL)); //Initialisation de la donnée seed
18     for (i=MIN;i<=Nbrcycle;i++){
24     nombre = (rand()%MAX)+MIN; // MIN<= nombre <= MAX
25
26     printf("%d",nombre); // rand renvoie un nombre calculé à partir de la donnée seed
27     getch();

```

**Figure III.9.** Programme en C++ (sous Windows) pour générer **300 nombre pseudo-aléatoire dans l'intervalle [Min, Max]** avec la combinaison des fonctions "srand ()", "time(NULL)", "rand ()" et "modulo", de la bibliothèque standard de ce langage.



### III.7 Test sur le tirage des nombres pseudo-aléatoires

Le but de ce test est d'analyser la distribution d'une sortie de nombres, afin de vérifier leur validité pour un usage de simulation. Pour cela, nous avons effectué 100 tirages de nombres pseudo-aléatoires que nous avons mis dans un tableau de dimension 10X10 (Tableau III.1), ensuite, le même travail a été effectué pour 300 tirages de nombres pseudo-aléatoires mis, ligne par ligne, dans un tableau de dimension 15X20 (Tableau III.2).

#### a) Cas de 100 tirages

Dans le cas des 100 tirages, nous avons divisé notre échantillon de 100 valeurs en 10 intervalles de même largeur. Chaque intervalle porte une couleur donnée, comme le montre le tableau suivant :

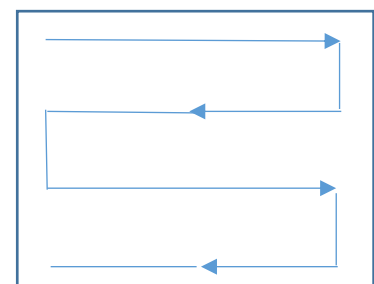
intervalle	1-9	10-19	20-29	30-39	40-49	50-59	60-69	70-79	80-89	90-100
%	12%	10%	11%	9%	9%	9%	9%	10%	9%	12%
Couleur	jaune	rouge	bleu	vert	gris	orange	noir	bleu	jaune	rouge

Nous pouvons remarquer que les couleurs (représentant un intervalle de valeur) ont eu, presque le même nombre d'apparition sur le tableau III.1 de 100 tirages.

Tableau III.1. Résultats des nombres P. Aléatoire compris entre [1-100] pour 100 fois de tirages.

33	51	33	15	89	14	41	84	94	77
96	57	36	56	14	29	5	78	67	67
76	35	57	48	89	82	20	71	63	65
51	59	93	42	1	27	33	36	24	38
12	79	20	60	9	76	37	92	45	68
97	46	16	6	47	10	45	94	82	14
76	23	99	77	95	95	8	1	39	61
76	8	77	8	66	20	64	97	29	34
24	42	88	12	85	59	15	6	85	3
51	24	52	93	5	27	58	94	3	85

Remplissage du Tableaux III.7 et III.8 par 100 tirage de nombres pseudo-aléatoires



En traçant la fréquence d'apparition de chaque intervalle (ou chaque couleur), en fonction de la largeur de l'intervalle correspondant, pris sur une échelle de 100 (valeur maximale du nombre pseudo-aléatoire), nous obtenons le tracé de la figure III.10.

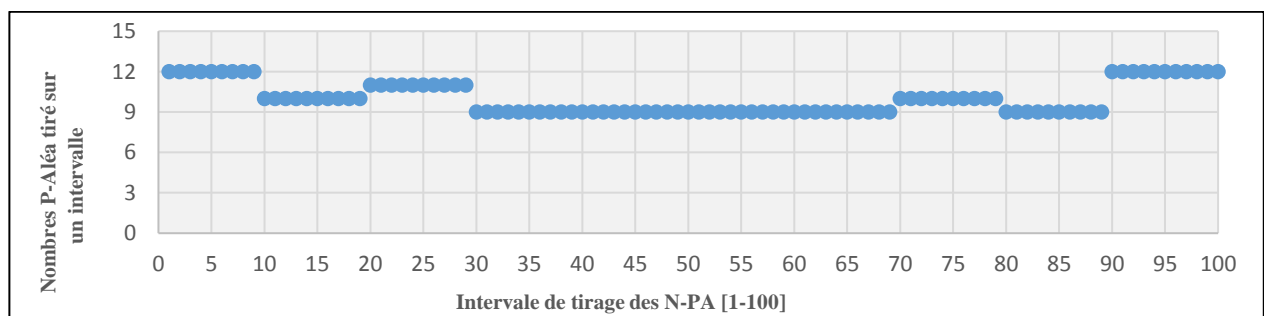


Figure.III.10. Tracé d'uniformité du GN-PA pour 100 tirages dans l'intervalle [1-100]

**b) Cas de 300 tirages**

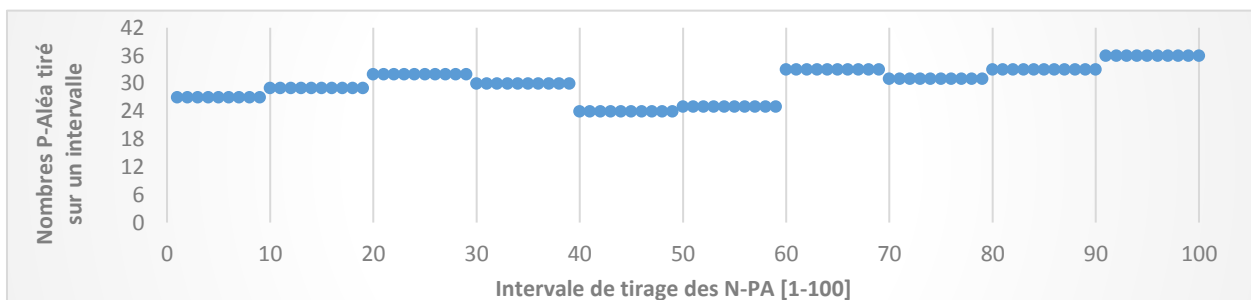
Dance ce cas, nous avons effectué 300 tirages de nombres pseudo-aléatoires, compris entre 1 et 100, que nous avons mis dans un tableau de dimension 15X20 (Tableau III .2), rempli ligne par ligne. Ensuite, nous avons divisé notre échantillon de 300 valeurs en 10 intervalles de même largeur. Chaque intervalle porte une couleur donnée, comme le montre le tableau suivant :

Intervalle	1-9	10-19	20-29	30-39	40-49	50-59	60-69	70-79	80-89	90-100
N	27	29	32	30	24	25	33	31	33	36
%	9%	9,66%	10,66%	10%	8%	8,33%	11%	10,33%	11%	12%
Couleur	jaune	rouge	bleu	vert	gris	orange	noir	bleu	jaune	rouge

**Tableau III .2.** Résultats des nombres P. Aléatoire compris entre [1-100] pour 300 de tirages.

36	61	9	88	50	91	23	64	2	93	59	21	99	63	74
64	92	86	58	23	80	28	40	43	46	36	51	18	91	66
13	93	3	84	85	32	32	84	11	24	89	91	78	84	26
22	24	16	74	98	29	91	46	17	15	44	80	37	41	51
46	57	11	64	7	24	86	74	23	30	36	48	73	10	29
96	66	20	87	29	31	58	79	7	93	66	49	56	60	66
42	81	66	92	28	61	74	79	26	1	53	54	57	77	63
51	90	46	61	22	52	93	62	19	62	39	99	61	16	93
91	77	17	90	95	2	12	81	7	18	94	77	21	42	31
36	5	65	32	11	5	47	56	92	2	3	10	36	36	9
87	63	50	97	77	25	80	40	31	42	20	26	95	5	76
19	13	76	43	54	81	75	97	78	36	39	43	19	71	57
22	82	7	27	44	68	59	42	46	56	24	78	55	1	23
97	36	5	33	70	11	33	79	65	54	1	66	2	10	92
3	35	19	47	88	37	5	81	84	39	27	99	37	89	84
25	90	39	7	21	89	18	75	65	59	45	67	63	90	68
18	61	16	73	70	10	14	37	30	40	57	17	73	86	80
60	69	96	98	88	69	85	82	39	94	39	69	24	72	80
58	90	79	81	1	20	24	74	36	93	74	1	41	55	76
95	70	2	4	60	97	15	76	83	84	13	21	65	80	1

En traçant la fréquence d'apparition de chaque intervalle (ou chaque couleur), en fonction de la largeur de l'intervalle correspondant, pris sur une échelle de 100 (valeur maximale du nombre pseudo-aléatoire), nous obtenons le tracé de la figure III.11.



**Figure.III.11.** Tracé d'uniformité du GN-PA pour 300 tirages dans l'intervalle [1-100]

### III.8. Analyse des résultats du GN-PA

Selon les résultats des tableaux (III.1 et III.2) et des figures III.10 et III.11, nous remarquons que la fréquence d'apparition des nombres pseudo-aléatoire (NPA), de notre générateur, est uniforme. Leurs fluctuations par rapport à leur valeur moyenne étant négligeable, pour les valeurs prises dans l'intervalle [1 - 100]

Nous remarquons aussi, sur les tableaux (III.1 et III.2), qu'il n'y a pas de répétitions de valeurs, et dans le cas d'existence d'une répétition la période est suffisamment longue.

En conclusion, nous pouvons dire que l'uniformité du générateur des nombres pseudo-aléatoire est vérifiée, les intervalles d'échantillonnage de ces nombres PA ne présentent aucune corrélations entre eux et la vitesse de tirage est minimale. Ce générateur est favorable pour un usage de simulation.

# Chapitre IV

## Résultats et Discussion

Dans ce chapitre nous allons décrire notre simulateur de la croissance cristalline d'une couche mince par épitaxie par jets moléculaires. Il s'agit d'une simulation élaborée en deux versions que nous détaillerons comme suite :

- **Version 1:** correspond à une croissance d'une monocouche à basse température : Interaction Gaz/Surface
- **Version 2 :** correspond à une croissance d'une monocouche à basse température : Phénomène de surface (Diffusion ou migration en surface).

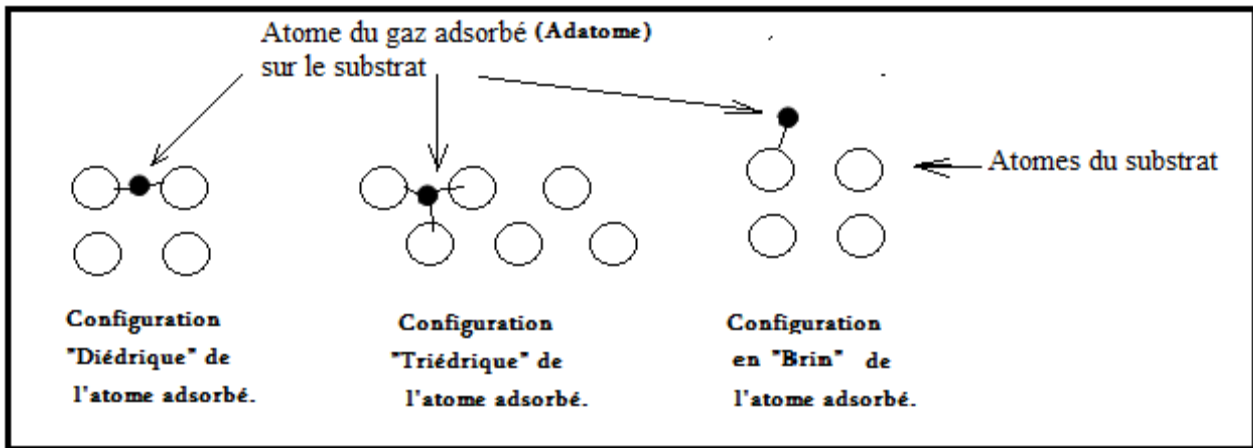
### IV.1. Procédure de simulation de la croissance cristalline d'une couche mince

#### IV.1.1. Version 1 du simulateur : Simulation de l'Interaction Gaz/Surface (Adsorption) à basse température

Un des points cible de ce travail est de faire apparaître le caractère dynamique de la croissance cristalline d'une couche mince sur un substrat par l'application de la méthode de Monte Carlo cinétique. Cette méthode permet de définir la cadence d'apparition des mouvements atomique en surface.

**IV.1.1.1. Procédure de gestion des atomes arrivant à la surface du substrat**

Lorsque l'atome arrive à la surface, il s'adsorbe, généralement, soit en configuration diédrique ou triédrique ou en brin. (Figure.IV.1).



**Figure.IV.1.**Type de configuration d'Adsorption en surface "

Le passage d'une configuration à l'autre est aussi probable selon la température et la nature chimique du système de croissance. Ainsi la simulation évolue comme une suite d'événements se réalisant à des instants précis. A chaque transition atomique (dans ce cas c'est le passage des atomes de la phase gazeuse au substrat par adsorption) est attribué un temps appelé "temps de réalisation " de l'événement, qui est un paramètre important pour la gestion des événements ayant lieu sur la surface depuis l'impact jusqu'à l'adsorption et l'incorporation.. Ce temps est défini par un tirage aléatoire de Monte Carlo et est déterminé selon la loi de poisson [26] [27]

$$t = - \frac{\ln(z)}{\lambda} \quad \text{IV.1}$$

"z" : est un nombre aléatoire uniformément réparti entre 0 et 1.

"λ" : est la probabilité de réalisation de l'événement par unité de temps.

La procédure de calcul de λ change selon que l'événement traité est une adsorption ou un événement de surface.

**Dans le cas de l'adsorption**, la probabilité λ est déterminée par la cinétique dans la phase gazeuse [2] :

$$\lambda = \frac{3.5 \cdot 10^{23} \cdot P \cdot S}{\sqrt{M \cdot T}} \quad \frac{\text{atome}}{s} \quad \text{IV.2}$$

"S" est la surface d'un site, "P" est la pression en Torr, "M" est la masse molaire de l'atome incidente sur la surface de croissance et "T" est la température du flux.

Pour les "événements de surface", les probabilités de transition sont exprimées à partir de la loi d'Arrhenius

$$\lambda = \nu e^{-\frac{\Delta E}{k_B \cdot T}} \quad \text{VI.3}$$

$\nu = 10^{13} \text{ s}^{-1}$  est la fréquence de tentative de saut. Cette grandeur pondère le terme exponentiel.  $\Delta E$  est l'énergie barrière associée à un mécanisme (en eV)  $k_B = 8,62 \cdot 10^{-5} \text{ eV.K}^{-1}$ , la constante de Boltzmann, "T" la température en Kelvin.

" $\nu$ " : est considérée comme uniforme et stationnaire car aux températures d'intérêt dans ce travail, les variations du taux de transition dépendent principalement de  $\Delta E$ , énergies d'activation, plutôt qu'aux variations pouvant apparaître dans la fréquence  $\nu$ .

### IV.1.1.2. Procédure de programmation

(A)

```
Mesdonnees.h

//1ere partie du programme : Introduction des Bibliotheques

#include <stdio.h> /*bibliotheque contenant l'instruction printf , for ; int ...
#include <conio.h> /*bibliotheque contenant l'instruction getch () */
#include <stdlib.h> /*bibliotheque contenant l'instruction rand et ran_max () */
#include <math.h> /*bibliotheque contenant l'instruction ln (logarithme) */

// 2eme partie du programme : Introduction des constantes

const int L= 20; // L:Lignes: dimension longitudinale du substrat (en nombre de sites)
const int C= 20; //C: Colonnes:dimension transversale du substrat (en nombre de sites)
const int INIT=0; //INIT constante qui vaut ZERO
const float Timemax =200; //Temps final de simulation (temps pour arrêter le programme)
const int Evemax=1000; //Nombre maximale des evenements dans le processus de croissance
```

(B)

**Cal 1 :**

Site (i,j)-----temps « Arrivée »

(0,0) -----t(0,0)

(1,0)-----t(1,0)

.....

(i,0)-----t(i ,0)

Figure IV.2. (A) Fichier « Mesdocument » représentant les Inputs du Simulateur. (B) Structure du calendrier des « Arrivés » noté Cal 1. La dimension de cal 1 est égale à la dimension du substrat (en nombre de sites).

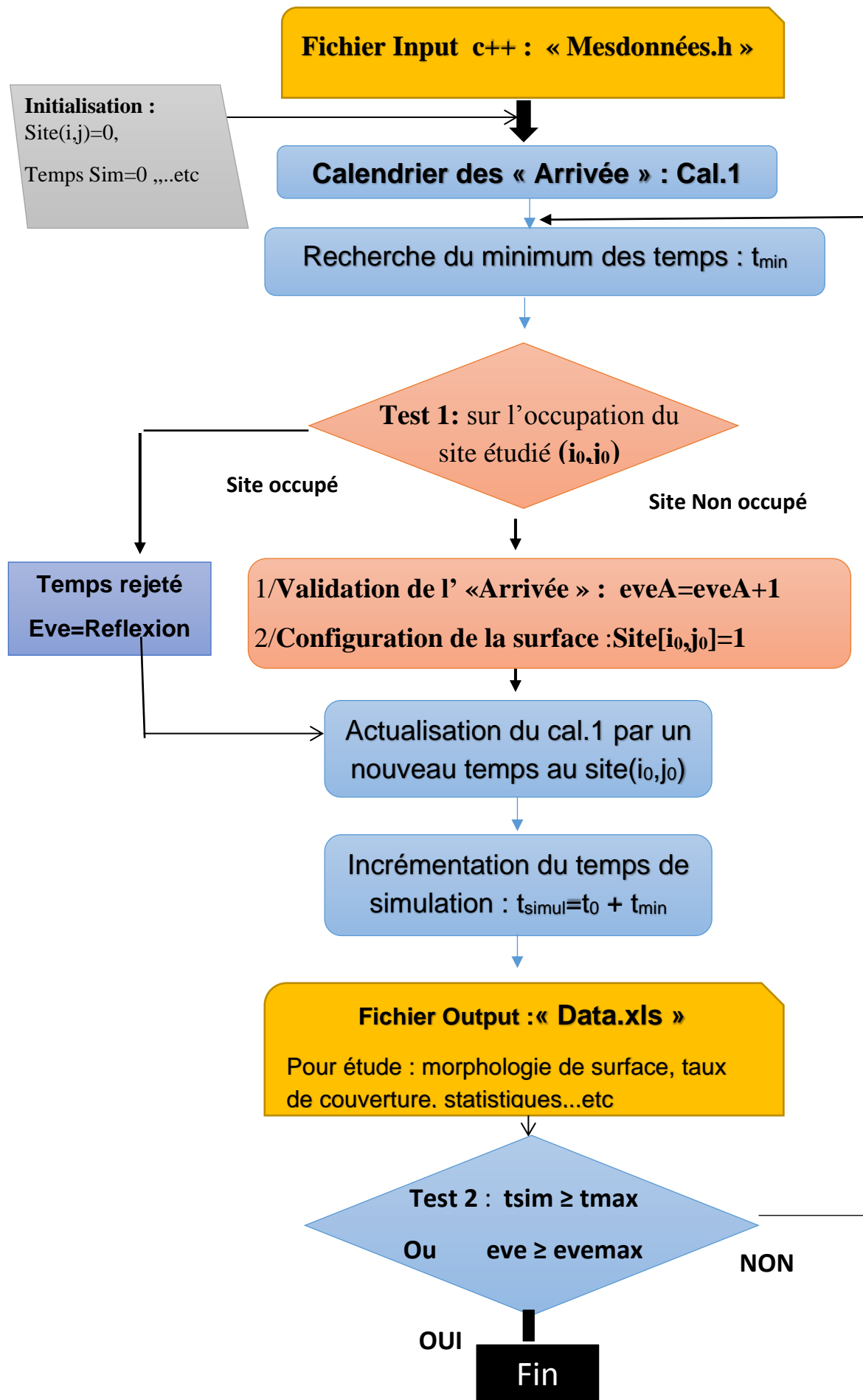


Figure IV.3. Organigramme du simulateur de la croissance cristalline par la méthode KMC SPARCC– Version 1 : Interaction Gaz/Surface par « Adsorption » des atomes en surface à basse température



Les étapes de déroulement de la simulation sont représentées par l'organigramme de la figure IV.3.

Le Programme Principale "P.P". :

- Lit toutes les données (Dimension du substrat, T, P, M,  $v$ ,  $\Delta E$ , ..etc) . Ces données sont introduites dans le programme sous forme d'un fichier « Mesdonnées.h » comme le montre la figure IV-2A
- Mémoire la configuration initiale du substrat (structure cristalline et orientation)
- Calcule la séquence temporelle des atomes, en communication avec les autres programmes (en particulier le Générateur RAND)
- Et génère le calendrier « Cal.1 » du déroulement des événements « Arrivée » (Figure IV-2B).
- Le programme "Configuration" réactualise l'état d'occupation de la surface en insérant l'événement en cours.

Pour le calendrier « cal.1 », le temps est calculé pour chaque site du substrat quel que soit son occupation (vacant ou occupé) .

L'organigramme du simulateur version 1 relatif à l'interaction Gaz/Surface par adsorption atomique des atomes du gaz est schématisé sur la figure IV.3 et une partie du programme réalisé en C++ est présenté sur la figure IV.3.

Sur l'organigramme de la figure IV-3, nous avons utilisé les couleurs pour montrer les modules liés à la gestion du Gaz et ceux liés à la gestion de la configuration de surface. En effet, nous désignons les programmes de gestion du Gaz par le Bleu, la gestion de la configuration de surface par le Rouge, l'initialisation des grandeurs physiques et mathématiques utilisées par le Gris et les fichiers d'entrée et sorties des données par le Jaune. Enfin la rupture du programme est désignée par « Fin » sur fond Noir.

#### IV.1.1.3. Résultats et validation du simulateur version 1 : Adsorption à Basse Température

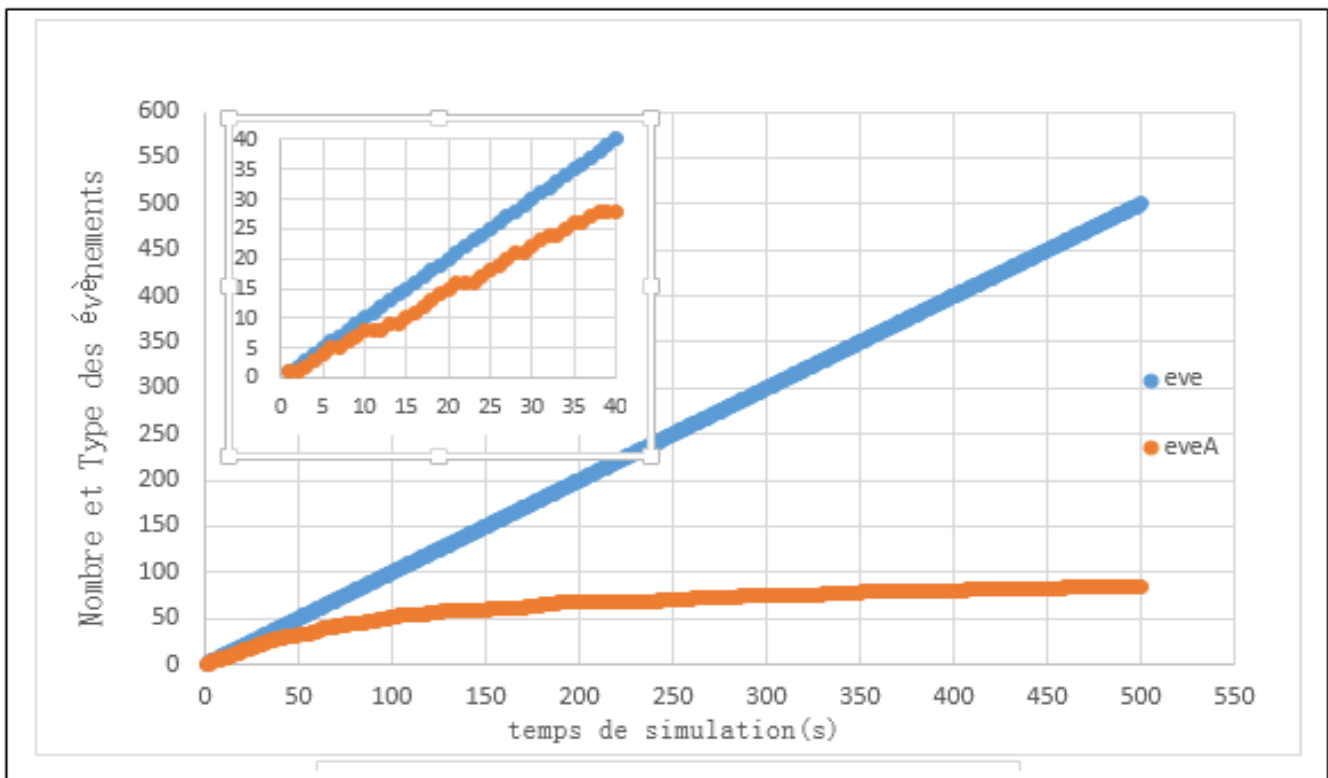
Tous les calculs présentés dans ce chapitre ont été menés sur une surface plane. Le réseau cristallin à deux dimensions est alors carré. La représentation du système étant périodique dans les deux directions du plan de croissance. Pour optimiser le temps de calcul, les dimensions du réseau 2D choisies sont 10x10 sites, ensuite 20x20 sites.

Notre objectif est de mieux comprendre l'influence du type d'évènement et la taille du substrat sur la distribution pseudo-aléatoire des atomes du flux incident sur le substrat. Nous allons pour cela analyser la morphologie de surface en fonction du temps de simulation et l'importance du type d'évènement sur l'uniformité de la couche réalisé à travers la simulation Monte Carlo cinétique.

**A) Etude de l'Evolution de la croissance en fonction du temps de simulation**

Dans cette version de la conception du simulateur de la croissance cristalline, un seul mécanisme est proposé pour la structuration de la surface. Il s'agit de l'adsorption. En traçant les courbes représentant le nombre d'évènement total en fonction du temps de simulation, on obtient le résultat représenté de la figure IV-4.

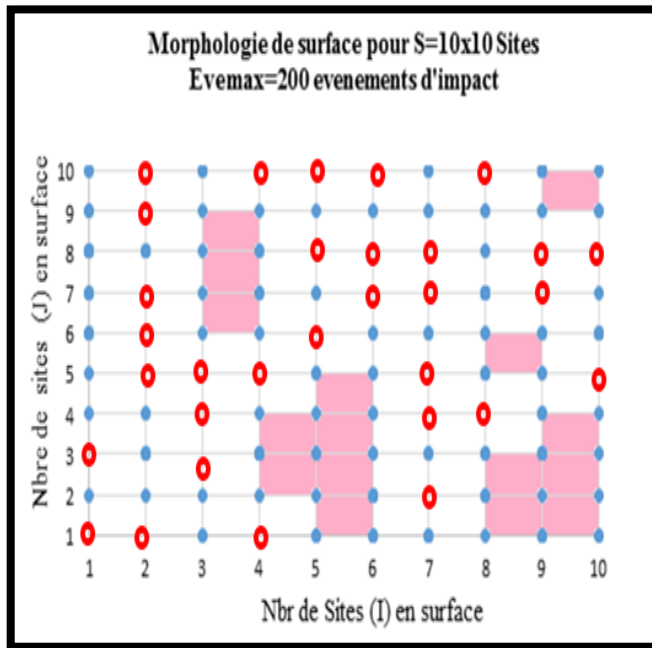
L'analyse de ces courbes nous permet de voir que **lors de la première phase de germination** (temps de simulation inférieur à 10 seconds correspondants à un nombre d'atomes adsorbé de l'ordre de huit) le libre parcours moyen en surface est élevé, ce qui revient à dire que la probabilité de réalisation d'un évènement « Arrivée » est bien élevée. En effet, dans cette première phase, le flux incident est presque complètement adsorbé de façon aléatoire sur les sites du substrat. Quant à la **deuxième phase**, considérée à partir d'un temps de simulation supérieur à 10 secondes), le taux de couverture de la surface de croissance va dépendre non seulement du nombre d'atomes incidents mais également de la configuration de la surface disponible pour leur déposition. Ainsi, on peut faire l'hypothèse que durant la première phase chaque impact est efficace et que les atomes adsorbés ne repartent pas dans le gaz par réflexion sur sites occupés. Ensuite, durant la deuxième phase, à partir de 10 secondes, la croissance s'effectue par coalescence des atomes adsorbés et se stabilise après 200 secondes (3 minutes). L'écart entre les deux courbes désigne la partie du flux réfléchi et non adsorbé.



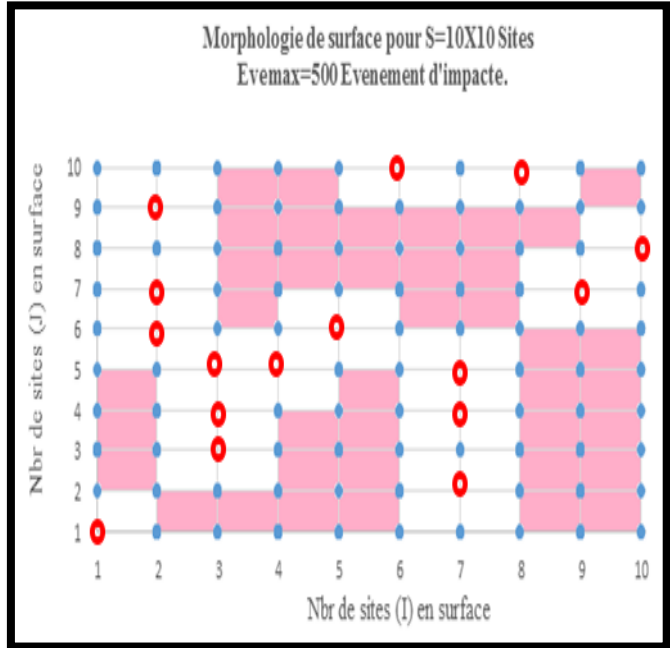
**Figure IV.4.** Tracé de l'évolution du nombre des évènements en fonction du temps de simulation. eve : est le nombre d'atomes incidents, et eveA : est le nombre d'atomes adsorbés pour une surface de 10x10 et Eve max=500.

**B) Etude de la morphologie de la surface au cours de la croissance d'une monocouche**

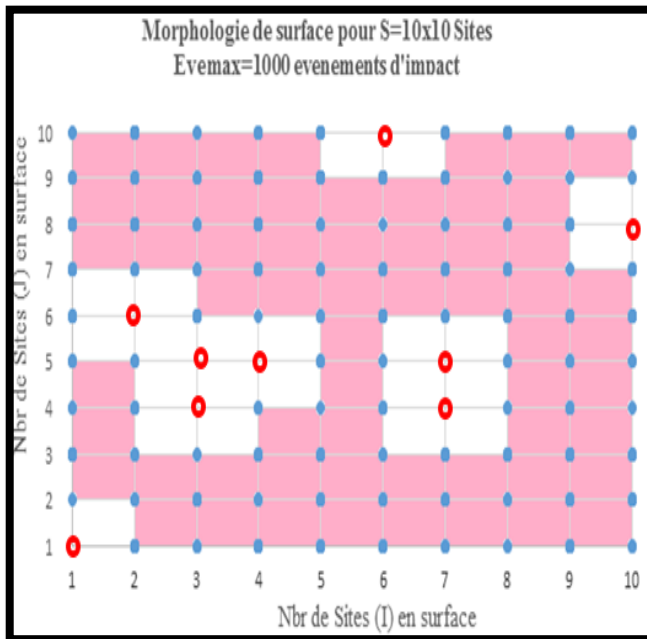
En effectuant des expériences de simulation sur un substrat de dimension 10x10 sites, avec un nombre d'évènement d'arrivée variant de 50 à 3000 évènements et en représentant la morphologie de surface pour différentes valeurs du nombre d'évènements sus-cité , on obtient les résultats représentés sur la figure IV-5.



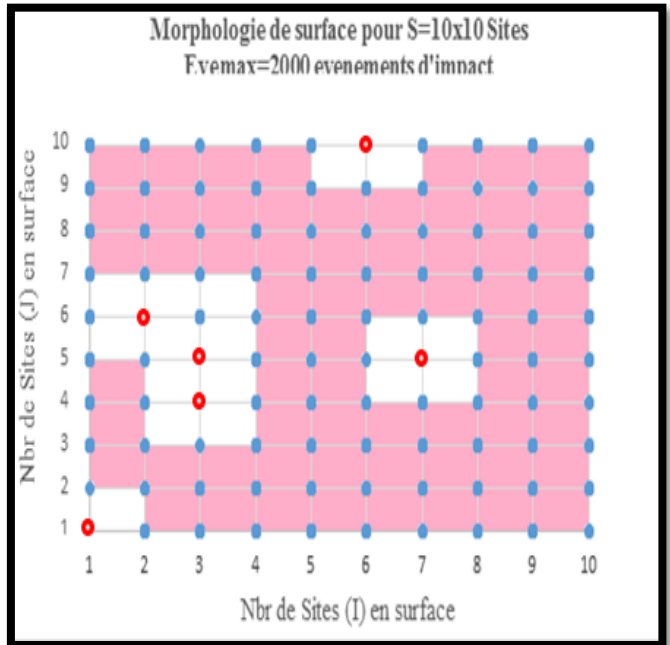
(a) Nombre de sites vacants=31, Taux  $\Gamma=0.67$ .



(b) Nombre de sites vacants=16, Taux  $\Gamma=0.82$ .



(c) Nombre de sites vacants=9, Taux  $\Gamma=0.92$ .



(d) Nombre de sites vacants=6, Taux  $\Gamma=0.94$ .

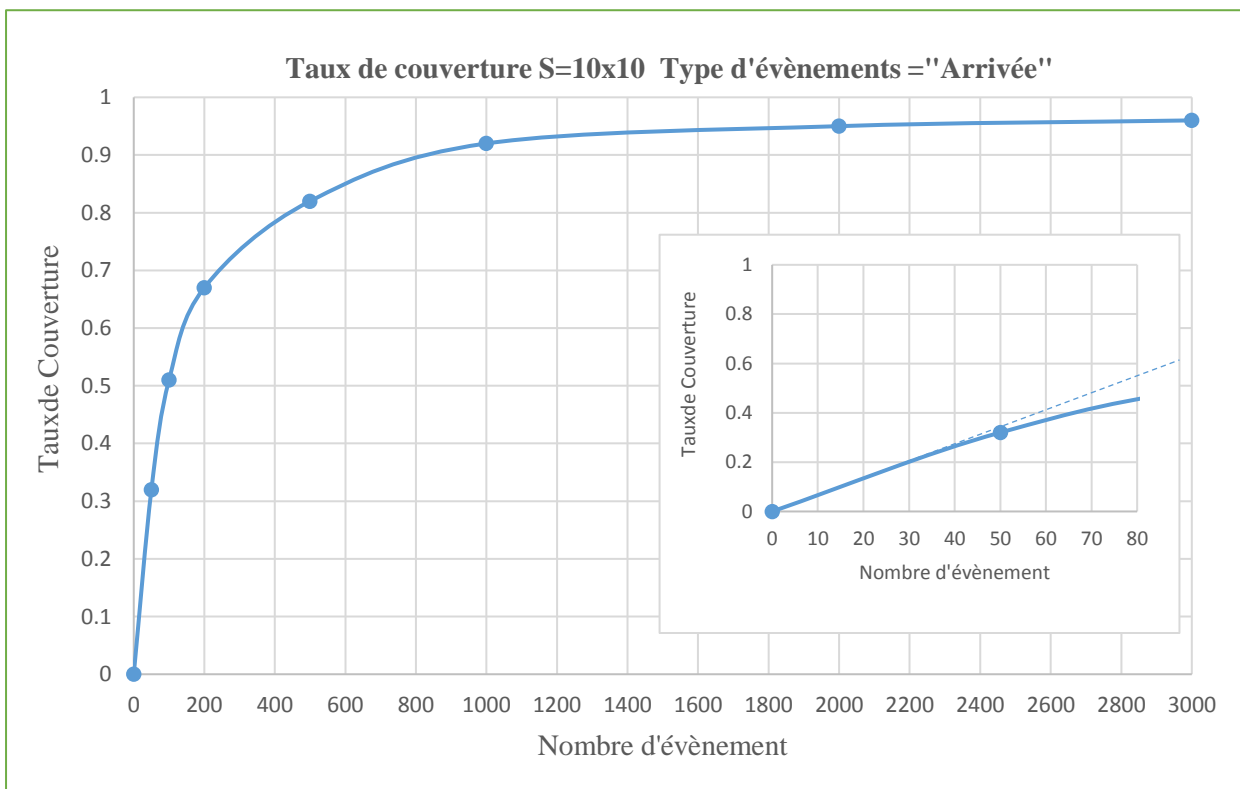
**Figure IV.5.** Evolution de la morphologie de surface pour différentes valeurs du nombre d'évènement total. Les sites vacants sont représentés en rouges et les sites occupés en bleu. Les surfaces colorées en violet représentent la couverture en cours de croissance sur le substrat représenté avec la surface blanche.

On peut remarquer sur la première morphologie (Fig.IV-5a) que la croissance de la monocouche est formée par une répartition aléatoire des germes ou petit amas (4 atomes) sur la surface, ensuite, sur la deuxième et troisième morphologie (Fig.IV-5b-5c), la croissance a évolué par coalescence des amas et formation d'un grand amas. La dernière morphologie (Fig.IV-5d) la monocouche à 94% de recouvrement.

**C) Etude de l'Evolution du taux de couverture de la surface de la croissance**

Les sites vacants (non occupés) sont représentés en rouge. Leurs nombres est mentionné en bas de chaque configuration de surface obtenue.

Nous pouvons voir sur la figure IV.6.que le taux de couverture évolue de façon non linéaire en fonction du nombre d'évènement. Ce tracé est formé de trois parties différentes que nous définissons comme suite :



**Figure IV.6.** Représentation du taux de couverture d'une surface 10X10 en fonction de Nbres d'évènements. (A) Tracé dans l'intervalle [0-3000] évènements (B) Tracé dans l'intervalle [0-80] évènements.

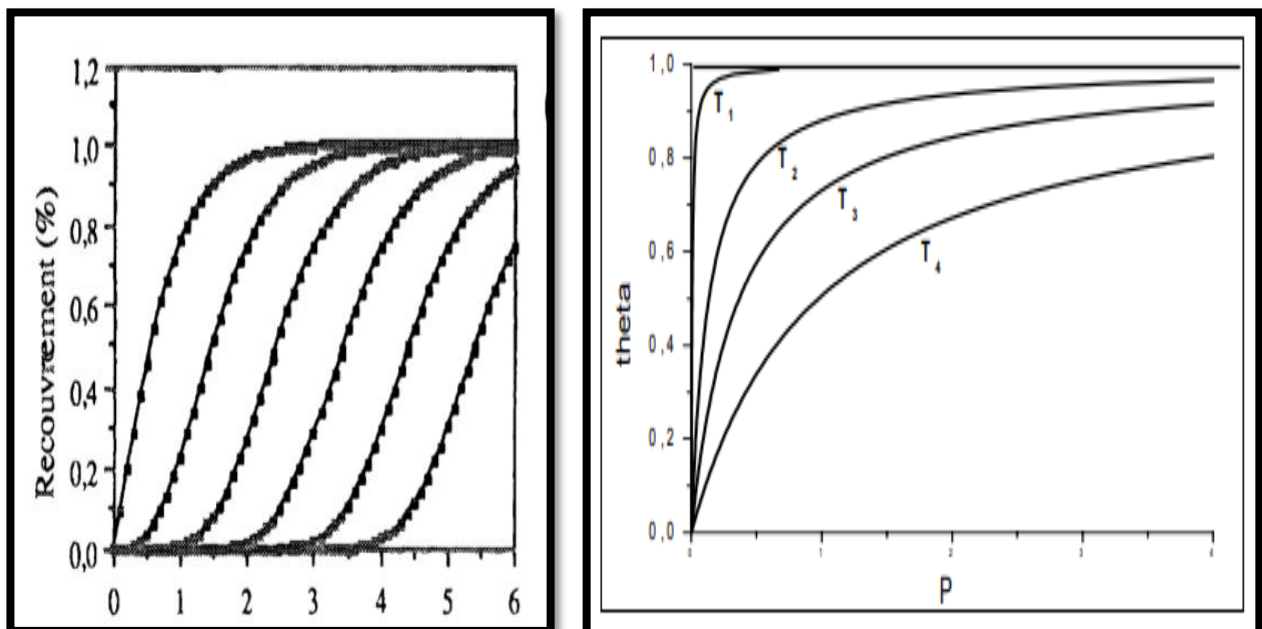
Dans la première partie de la courbe, figure IV-6B,  $0 < \text{Nbr. d'évènement} < 50$ , la variation est quasi linéaire. Ensuite, sur la figure IV-6A, une phase d'incubation est suivi aussitôt d'une phase stationnaire pour des valeurs du nombre d'évènement supérieures à 2000, la valeur du taux de couverture tend, dans cette dernière phase, vers une limite (environ 95% de taux de couverture). Cette limitation est par ailleurs

accompagnée de l'apparition de quelques sites vacants isolés (ou défaut) en surface, représentant les 5% des sites restants vacants.

Ainsi, en utilisant un modèle simplifié, basé uniquement sur des considérations géométriques du substrat, sans tenir compte d'interactions physiques ou chimiques entre les adatoms, il a été possible de retrouver le taux maximal de couverture lors de la croissance cristalline en couche mince, en accord avec les données réelles.

La valeur obtenue étant d'environ 95 %. Sachant que la limite observée expérimentalement est d'environ 90% [28] on peut raisonnablement conclure que le taux de recouvrement maximal est atteint.

La forme de la courbe de la figure IV.6 est connue sous le nom d'isotherme d'adsorption de Langmuir [29]. Notre tracé coïncide bien avec l'allure de la courbe prise à la plus basse température T1.



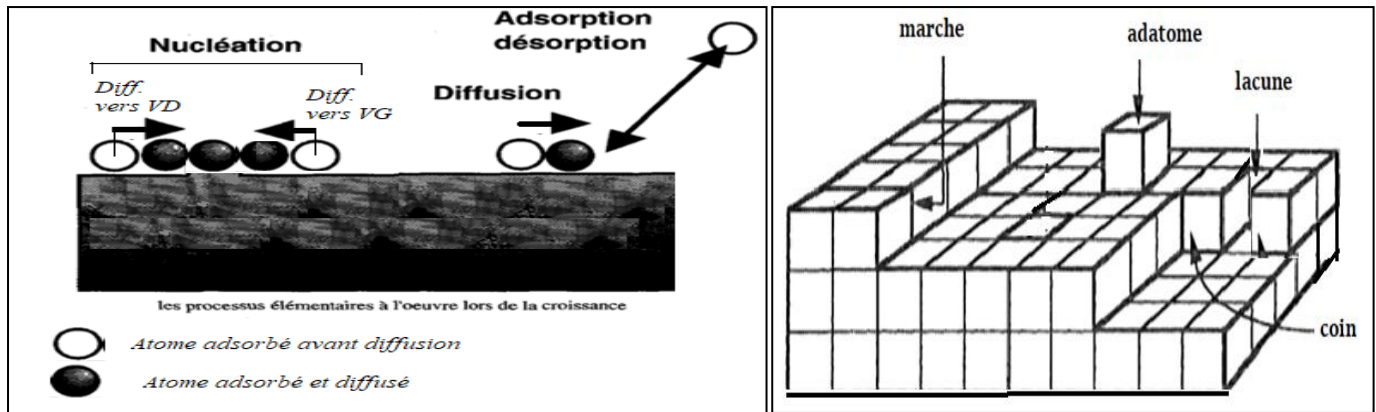
**Figure IV.7.** (A) modèle multicouche théorique et de simulation d'autres auteurs Madhukar [30] et S.V. Ghaisas [31] (croissance couche par couche, chaque courbe est une monocouche) (B) Isothermes d'adsorption de Langmuir :  $T_1 < T_4$  [32]

Une isotherme d'adsorption  $\theta = f(P)$  représente la variation du taux de recouvrement de la surface par adsorption d'un gaz, en fonction de la pression de ce dernier (c.à.d. à l'échelle atomique en fonction du nombre d'arrivée par unité de surface) pour différentes valeurs de températures (figure IV.7).

### IV.1.2. Version 2 du simulateur : cas de Diffusion en surface

#### A) Processus élémentaires

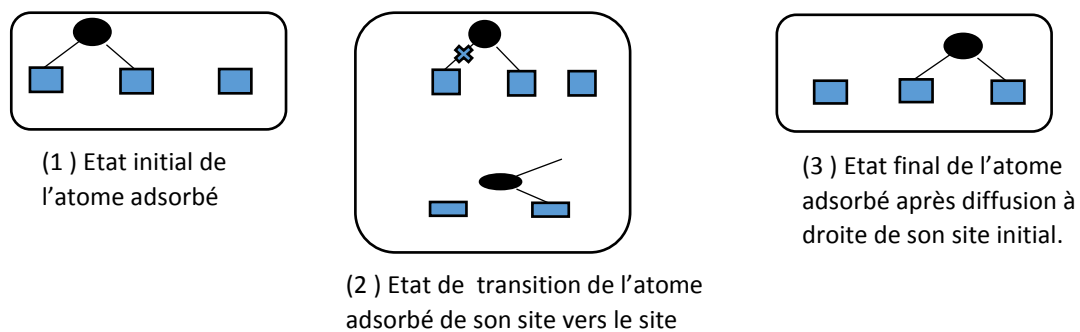
Pour réellement comprendre la croissance, il faut s'intéresser aux processus élémentaires au niveau atomique. Figure(IV.8)



(A) (B)  
**Figure IV.8 :** Schéma du processus élémentaire de la croissance cristalline en couche mince.  
 (A) Modèle de boule, (B) Modèle de cube de Kossel (Solid On Solid)

Les espèces provenant du flux incident sont adsorbées, elles peuvent aussi être désorbées. La désorption étant négligeable pour les matériaux que l'on dépose à basse ( $T < T_{ambiante}$ ). La diffusion atomique sur la surface (diffusion des adatoms) assure une redistribution des atomes.

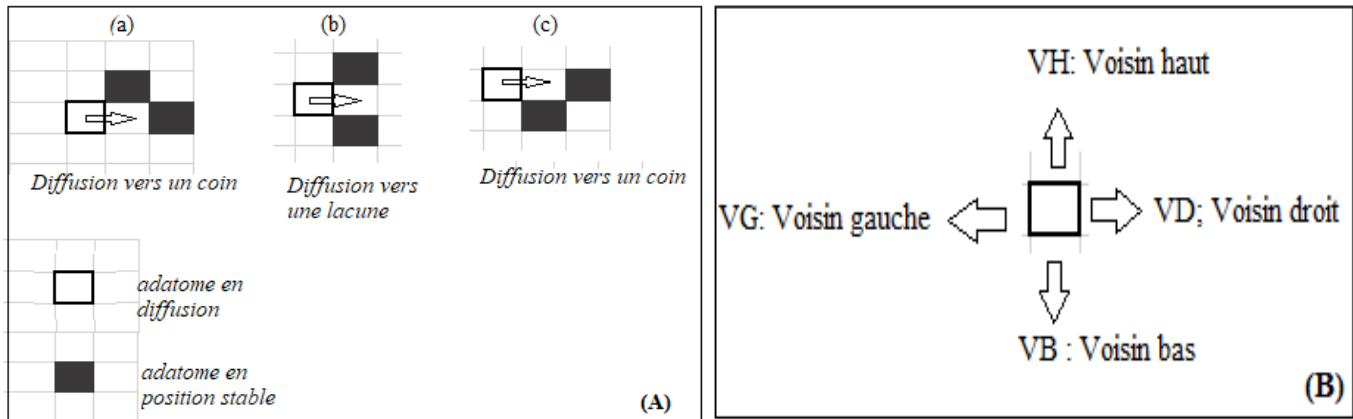
La figure IV.9 montre le processus de diffusion, à droite, d'un adatome.



**Figure IV.9 :** processus de diffusion d'un adatome vers le site de son voisin droit.

Les barrières d'activation de tous ces processus sont les paramètres clé de la croissance. Ils font maintenant l'objet de calculs numériques utilisant des potentiels d'interaction atomiques ou la DFT [35, 41] et pourront fournir les ingrédients de base pour des simulations numériques de type Monte Carlo cinétique de la croissance (voir annexe 2).

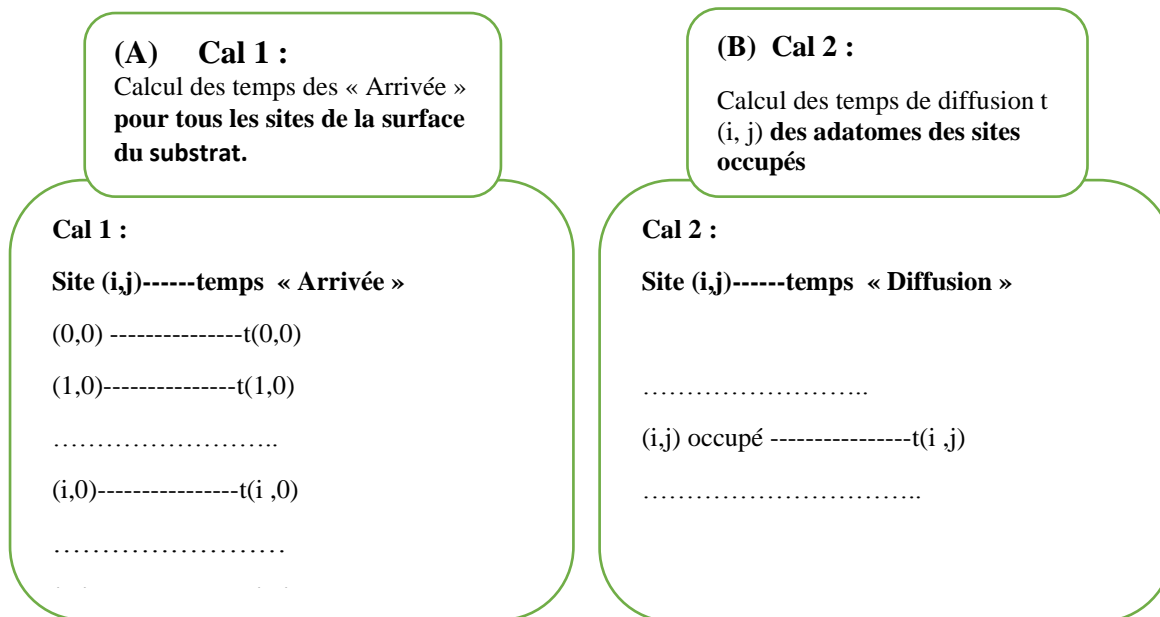
Dans notre situation qui a lieu à basse température, nous avons choisi la croissance par nucléation à travers la diffusion préférentielle (voir annexe 2 – Barrières énergétiques)) des adatoms vers des sites vacants ayant au moins deux liaisons avec leurs voisins proches (un coin ou une lacune surfacique) comme le montre la figure IV.10. Cette situation est considérée dans les quatre directions de la surface, à savoir : Diffusion à droite, Diffusion à gauche, Diffusion vers le haut, et Diffusion vers le bas de la surface.



**Figure IV.10 :** (A) Les situations de diffusion d'un adatome vers un site vacant ayant une structure de coin ou de lacune. (B) Les quatre positions de proches voisins en surface d'un adatome en vu de sa diffusion.

### B) Architecture du simulateur -Version 2

Dans cette deuxième version du simulateur, deux calendriers sont nécessaires pour la gestion des événements formant le processus de croissance, comme le montre le schéma de la figure IV.11:



**Figure IV.11.** Structure des calendriers :(A) « Arrivés » noté Cal 1, et (B) « Diffusion » noté Cal 2. La dimension de cal 1 est égale à la dimension du substrat (en nombre de sites). La dimension de Cal 2 dépend du nombre des adatoms.



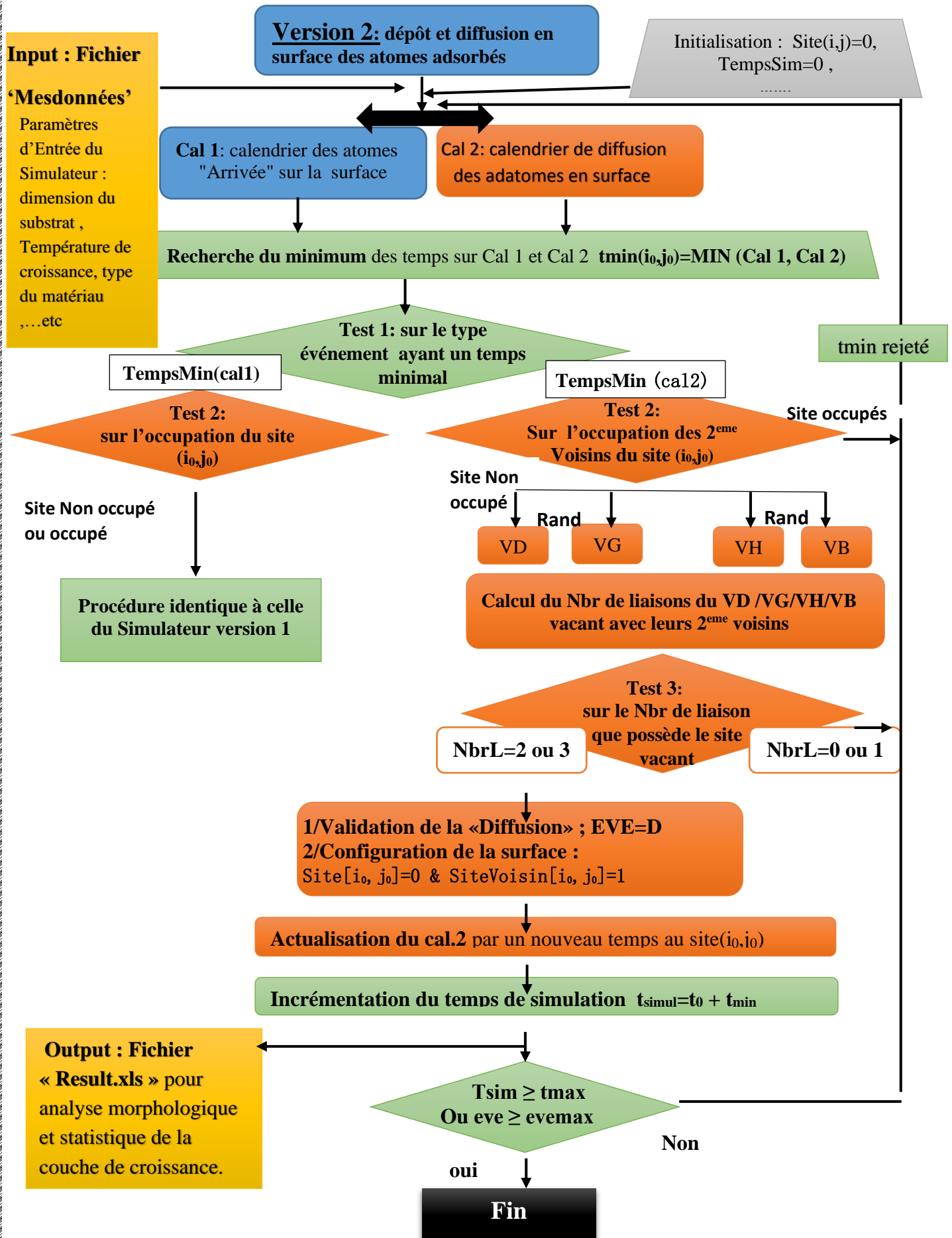


Figure IV.12. Organigramme du simulateur de la croissance cristalline par la méthode KMC – Version 2 -Adsorption et diffusion des atomes adsorbés en surface, à basse température.

- ⇒ Un Calendrier correspondant à l'impact d'atomes depuis la phase gazeuse :  
noté « Cal-1 » (identique à celui de la première version)
- ⇒ Un Calendrier correspondant au mouvement des adatoms sur la surface de croissance :  
noté « Cal-.2 » (calendrier pour la diffusion ou migration en surface)

la figure IV-11 montre la structure des deux calendriers.

### C ) Procédure programmation du simulateur - version 2

Pour cal.1, le temps est calculé pour chaque site du substrat quel que soit son occupation (vacant ou occupé) tandis que pour Cal.2, le temps est calculé uniquement pour les sites occupés par les adatoms qui ont une probabilité de diffusion (adatoms non incorporés).

L'organigramme de la simulation de la croissance cristalline d'une monocouche avec diffusion atomique en surface, est représenté sur la figure IV.12.

Les étapes de déroulement de la simulation sont représentées par la figure IV.12. Ainsi le programme principal :

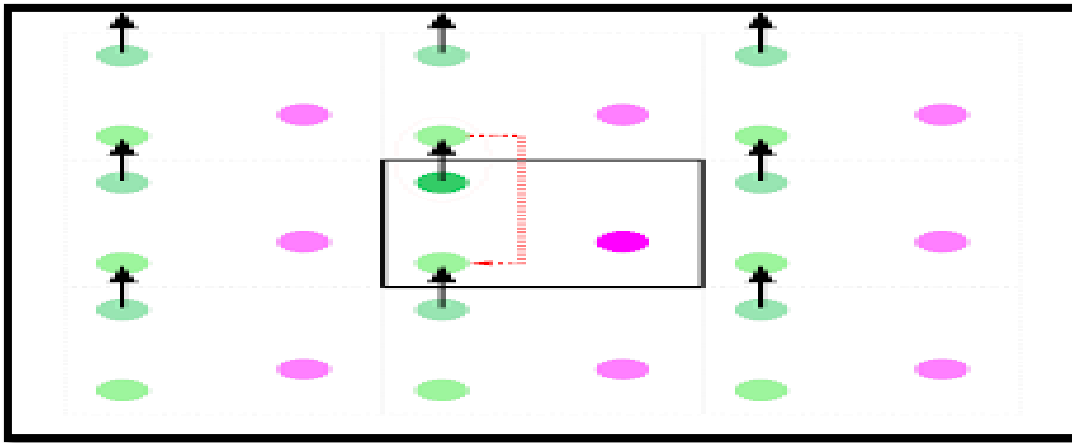
- ⇒ Lit toutes les données à travers le fichier d'input « Mesdonnées.h »
- ⇒ Mémorise la configuration initiale du substrat et le voisinage (VD, VG, VH, VB) des sites occupés par les adatoms,
- ⇒ Calcule la séquence temporelle des atomes du flux incident (Cal 1), en communiquant avec les différents programmes à travers des tests (en particulier le programme Rand)
- ⇒ Et génère le calendrier de déroulement des événements de diffusion en surface (Cal 2).
- ⇒ Le programme de configuration réactualise l'état d'occupation de la surface en insérant l'événement en cours.

Le calcul des énergies d'activations relatives à chaque configuration atomique est attribué au calcul ab-initio (DFT) résumé en Annexe 2. Dans le cas de notre simulateur version 2, les événements de surface retenus, à basse température, sont les diffusions vers les sites de coins ou de lacunes (Figure IV-10 et Annexe 2).

Les événements de surface tel que la désorption, la diffusion inter couches, la création et annihilation de défauts (contraintes) seront prises en charge ultérieurement dans les prochaines versions. Le programme correspondant à cette version 2 du simulateur est présenté en Annexe 3

### D) Conditions aux limites périodiques

D'une façon générale, les systèmes que l'on peut simuler sont de taille finie. Le nombre maximum d'atomes que l'on peut envisager dans une simulation est de l'ordre de quelques milliers. Ce sont la capacité mémoire et la puissance des ordinateurs qui limitent cette taille.



**Figure IV.13** : Représentation schématique du découpage d'un l'échantillon réelle de façon périodique et application des conditions aux limites sur la partie simulée de l'échantillon

Le meilleur moyen de contrôler les effets de taille est de prendre une géométrie de boîte de simulation compatible avec les conditions aux limites périodiques. Par exemple, un réseau cubique à trois dimensions peut être simulé par une boîte cubique. Dans notre cas, pour conserver les propriétés du cristal infini, le cristal est en quelque sorte découpé en sections identiques, et le calcul de simulation peut porter sur une seule section, avec les conditions aux limites périodiques dans les directions latérales x, y. La figure IV.13 montre ce principe par le fait que lorsqu'un atome sort de la section étudiée, un atome y rentre par le côté opposé.

### VI.1.3. Résultat de la simulation : Cas de diffusion en surface de croissance cristalline via la méthode KMC.

En effectuant des simulations sur un substrat de dimension 20x20 sites, avec un nombre d'évènements d'arrivée variant de 150 à 5000 évènements, on obtient les valeurs regroupées dans le tableau IV.1.

#### IV.1.3.1 Etude du taux de couverture de la surface de croissance

##### A) Etude de l'évolution du taux de couverture de la surface de croissance

En représentant l'évolution du taux de couverture de la surface de croissance en fonction du nombre d'évènement, à partir du tableau de valeur Tab. IV.1, nous obtenons la courbe de la figure IV.14A.

**Tableau.IV.1.** Présentation des différentes informations obtenu par SPARCC (Simulateur PARTiculaire de la croissance cristalline) sur une surface du substrat S=20x20 sites et pour Nbr d'Eve [150-5000] , dans les 2 versions de SPARCC-Basse température.

N	Nbr d'Eve	Version 1 Mécanismes : « Arrivée »					Obs	Version 2 Mécanismes : « Arrivée » + « Diffusion en surface »				
		Nbr de lacune	Nbr des Atomes Adsorbés sur la surface	Nbr des Atomes (réfléchis)	Taux de couverture Max ( $\theta$ )	Tems (S) nécessaire qui correspond Ou Taux de couverture Max		Nbr de lacune	Nbr des Atomes Adsorbé sur la surface	Nbr des Atomes (réfléchis)	Taux de couverture ( $\theta$ )	Tems (S) nécessaire qui correspond Ou Taux de couverture Max
1	150	<b>290</b>	<b>110</b>	<b>40</b>	<b>0.27</b>	<b>11.94</b>		<b>255</b>	<b>145</b>	<b>5</b>	<b>0.37</b>	<b>11.04</b>
2	300	231	169	131	0.42	38.82		221	179	121	0.45	36.97
3	400	<b>202</b>	<b>198</b>	<b>202</b>	<b>0.47</b>	61.31	<b>A</b>	<b>195</b>	<b>205</b>	<b>195</b>	<b>0.51</b>	59.04
4	700	146	254	446	0.63	143.12		140	260	440	0.64	139.13
5	1000	107	293	707	0.73	246.03		103	297	703	0.74	241.09
6	1500	87	313	1187	0.78	446.92		81	319	1181	0.79	442.08
7	2000	72	328	1672	0.82	635.21		67	333	1667	0.83	629.53
8	2300	60	340	1960	0.85	764.19		56	344	1956	0.86	758.04
9	2500	54	346	2154	0.87	846.05		53	347	2153	0.87	839.51
10	2800	47	353	2447	0.88	976.83		45	355	2445	0.89	973.25
11	3000	44	356	2644	0.89	1069.59		42	358	2642	0.9	1056.52
12	3300	37	363	2937	0.90	1193.42		34	366	2934	0.91	1187.06
13	3500	<b>35</b>	<b>365</b>	<b>3135</b>	<b>0.91</b>	1283.88	<b>B</b>	<b>34</b>	<b>366</b>	<b>2934</b>	<b>0.91</b>	1278.54
14	3800	32	368	3432	0.91	1414.82		30	370	3430	0.92	1412.92
15	4000	30	370	3630	0.92	1505.25		28	372	3628	0.93	1499.82
16	4100	30	370	3730	0.92	1548.92		28	372	3728	0.93	1545.04
17	4300	30	370	3930	0.92	1637.51		28	372	3928	0.93	1633.27
18	4500	<b>28</b>	372	4128	0.93	1720.80		27	373	4127	0.93	1715.82
19	4800	26	374	4426	0.93	1862.68		26	374	4426	0.93	1854.50
20	5000	<b>25</b>	375	4625	0.94	1960.14		24	376	4624	0.94	1954.93

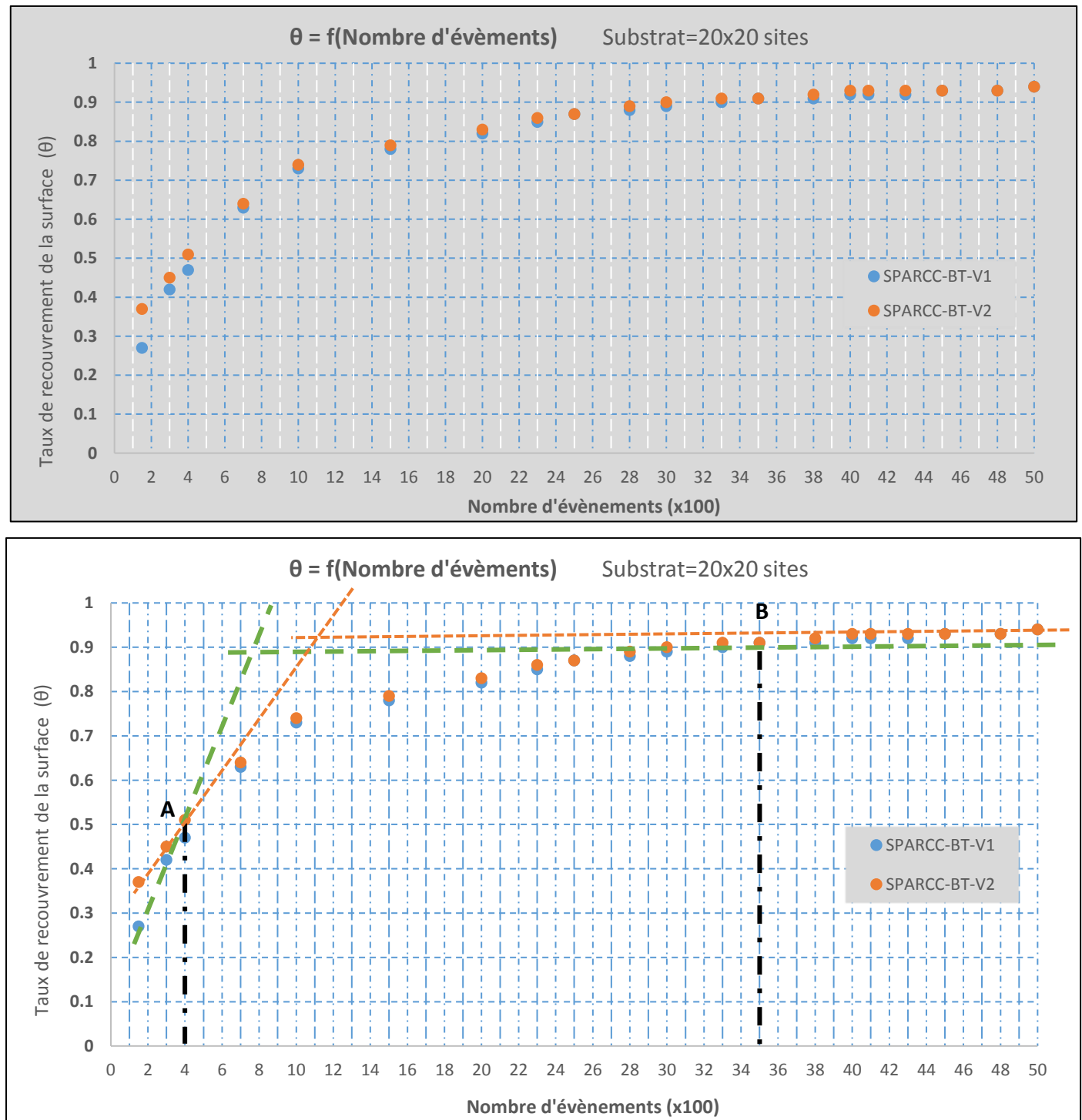
L'analyse des courbes des figures IV.14A et IV-14B, montrent qu'elles sont formées de trois parties que nous décrivons comme suite :

- **Première phase** : 1<sup>ère</sup> phase de croissance initiale :

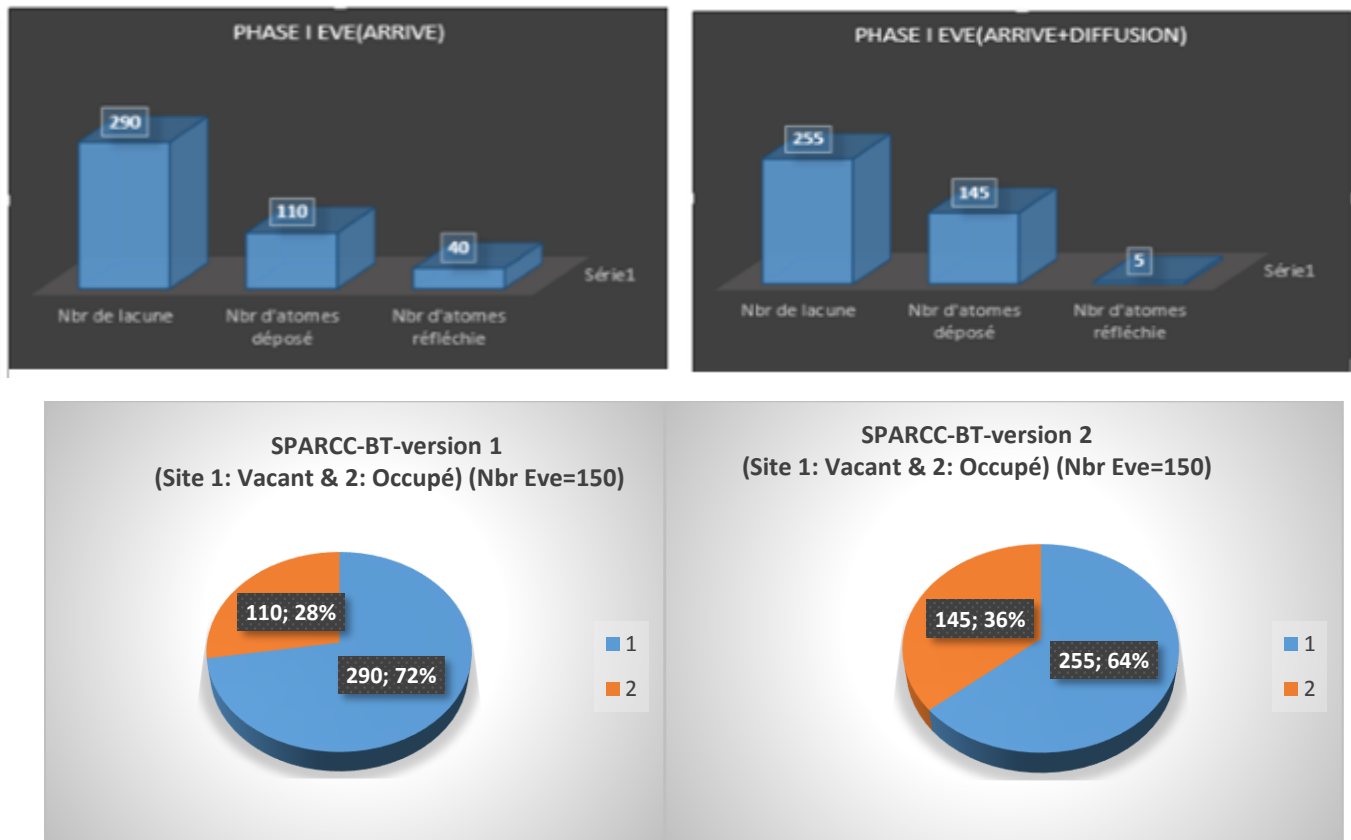
Durant la phase initiale de la croissance où le nombre d'événements s'étale à 400 événements, en choisissant un point de cette phase correspondant à un nombre de 150 événements, on remarque sur les représentations statistiques de la figure IV.15 que sur 290 événements repartis entre l'arrivée (adsorption/Reflexion) et la diffusion, seule 110 ont été adsorbé soit un taux de couverture de la surface de l'ordre de 28% dans SPARCC-

BT-version 1 alors qu'elle atteint 37% dans SPARCC-BT-version 2 , avec 145 atomes adsorbé sur une surface du substrat de 400 sites.

Le nombre de reflexion des atomes de gaz sur les sites occupés de la surface etant moins élevé en prenant en compte le mécanisme de diffusion dans le simulateur de la croissance cristalline, à basse température.



**Figure IV.14.** Représentation du taux de recouvrement d'une surface 20X20 en fonction de Nombre d'évènements. (A)Tracé globale (B) Analyse du Tracé



**Figure IV-15 :** Statistiques relevées durant la phase initiale de la croissance. (Haut) type d'évènements et type de sites en surface (Bas) Type de sites sur la surface de croissance en %

- **Deuxième phase :** pour un nombre d'évènement variant de 400 à 3500 :

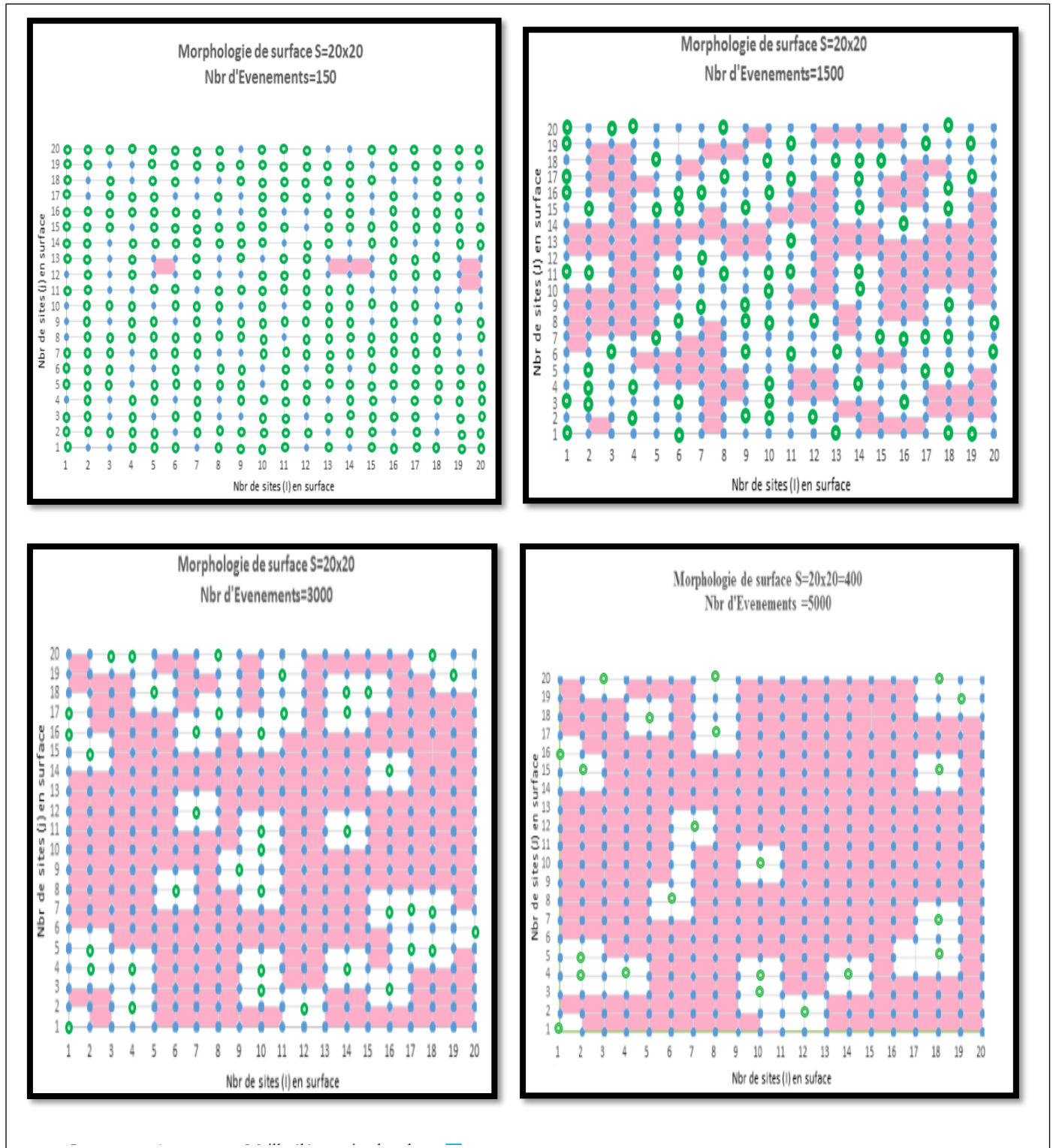
Cette phase peut être décrite à travers l'étude morphologique de la surface. En effet, durant cette phase la croissance évolue par coalescence des amas. Cela veut dire que les germes (petit amas) créés durant la Première phase de croissance (Figure IV-16A), commence à grandir par fusion entre eux, durant cette deuxième phase de croissance (Figure IV-16B-C), pour former des grands amas. Ces derniers grands amas gênent l'adsorption de nouveaux atomes comparaison à cas de la première phase.

- **Troisième phase :** pour un nombre d'évènement supérieur à 3500 :

Sur la figure IV-16D, nous notons la présence d'un nombre relativement fixe des sites non recouverts par les atomes « Arrivé », indiquant ainsi, que la surface s'est stabilisée durant cette troisième et dernière phase de croissance. Ceci est une conséquence du fait que l'adsorption des atomes « Arrivé » sur la surface est limitée par sa nouvelle configuration (présence de lacunes, défauts de surface) qui est devenue non favorable à de



nouvelle adsorption. C'est ce qui limite l'initiation d'une couche suivante, à basse température, dans le cas d'une croissance multicouche. L'élévation de la température se traduit, en simulation, par l'activation de nouveaux mécanismes, telle que les diffusions intercouches, qui ne fait pas l'objet de ce travail de master



**Figure.IV.16. Evolution** de la morphologie de surface d'une surface=20X20 sites, pour certains Taux de couverture (A, B,C,D) et détermination du nombre de sites vacants en fonction du nombre d'Evènements (Arrive+diffusion) . Les sites vacants sont représentés en vert, les lacunes en bleus, leurs nombres sont mentionnés en bas de chaque configuration de surface obtenue.



### B) Etude de la rugosité de surface et validation du résultat

Dans la croissance cristalline, l'atome qui arrive sur le substrat reste sur le site d'impact, ou diffuse avec de faible longueur de diffusion (du site d'impact vers son voisin le proche en surface), en parallèle avec l'incidence du flux de gaz, ceci entrainera une rugosité rapidement croissante.

▪ **Formulation de la rugosité**

Dans cette modélisation de la croissance type SOS (solid on solid), soumise à un flux poissonnien d'atomes [33]; la rugosité serait l'écart-type de cette distribution, c'est à dire qu'elle serait proportionnelle à la racine carrée de l'épaisseur.

$$\sigma_h(t) = \frac{[\sum_{ij}[h_{ij}(t) - \overline{h}(t)]^2]^{1/2}}{N}$$

$h_{ij}(t) - \overline{h}(t)$  Étant l'écart type de la hauteur de la surface au site (i, j) par rapport à la hauteur moyenne qui vaut :

$$\overline{h}(t) = \frac{\sum_{ij} h_{ij}}{N}$$

Où N est le nombre de sites en surface. Le tableau IV-2 montre la méthode de programmation de la rugosité, de façon itérative.

**Tableau.IV.2.** Méthode d'estimation de la rugosité, dans le simulateur-V2 , au cours de la croissance cristalline via la méthode KMC. ( $h_{ij}=1$  monocouche unité « a » paramètre de maille, et N= dimension du substrat = 20x20 sites en surface)

Coordonnées du Site de l' « Arrivée »	Temps de simulation	Hauteur du site (i,j) ou de la couche en croissance	Hauteur moyenne de la couche en croissance (unité « a » paramètre de maille) $\overline{h}(t) = \frac{1}{N} \sum_{ij} h_{ij}$	Variation locale de la hauteur du site par rapport à la hauteur moyenne $\Delta h_{ij} = h_{ij}(t) - \overline{h}(t)$	Rugosité $\sigma_h(t) = \left[ \frac{1}{N} \sum_{ij} (h_{ij}(t) - \overline{h}(t))^2 \right]^{1/2}$
(i,j)	t <sub>sim</sub>	$h_{ij}(t)$	$\overline{h}(t)$	$\Delta h_{ij}$	$\sigma_h(t) = \left[ \frac{1}{N} \sum_{ij} (\Delta h_{ij})^2 \right]^{1/2}$
(i <sub>0</sub> ,j <sub>0</sub> )	t <sub>0</sub>	$h_{00} = 1$	$\overline{h}(t) = \frac{h_{00}}{N} = \frac{1}{400} = 0.25 \cdot 10^{-2}$	$h_{00} - \overline{h}(t) = 1 - \frac{1}{400} = \frac{399}{400}$	$\sigma_h(t) = \left[ \frac{1}{N} \left( \frac{399}{400} \right)^2 \right]^{1/2} = \left[ \frac{1}{400} \left( \frac{399}{400} \right)^2 \right]^{1/2} =$
(i <sub>1</sub> ,j <sub>1</sub> )	t <sub>1</sub>	$h_{11} = 1$	$\overline{h}(t) = \frac{h_{00} + h_{11}}{N} = \frac{2}{400} = 0.5 \cdot 10^{-2}$	$h_{11} - \overline{h}(t) = 1 - \frac{2}{400} = \frac{398}{400}$	$\sigma_h(t) = \left[ \frac{1}{N} \left( \frac{398}{400} \right)^2 \right]^{1/2} = \left[ \frac{1}{400} \left( \frac{398}{400} \right)^2 \right]^{1/2} =$
(i <sub>1</sub> ,j <sub>2</sub> )	t <sub>2</sub>	$h_{12} = 1$	$\overline{h}(t) = \frac{h_{00} + h_{11} + h_{12}}{N} = \frac{3}{400} = 0.75 \cdot 10^{-2}$	$h_{11} - \overline{h}(t) = 1 - \frac{3}{400} = \frac{397}{400}$	$\sigma_h(t) = \left[ \frac{1}{N} \left( \frac{397}{400} \right)^2 \right]^{1/2} = \left[ \frac{1}{400} \left( \frac{397}{400} \right)^2 \right]^{1/2} =$
*****	*****	*****	*****	*****	*****

Le résultat de notre simulation est représenté par le tracé de la figure IV.17. Nous remarquons, sur ce tracé, une première phase, correspondant à un temps de simulation inférieur à 0.05s, caractérisée par une rapide évolution de la rugosité, suivie d'une phase d'amortissement de la rugosité (asymptote de faible angle avec l'axe des abscisses) s'expliquant par le fait que la croissance de la première couche se fait sur une surface plane (pas de marches) et stabilisée en sites vacants, ensuite la coalescence des amas durant la deuxième phase de croissance (Figure-IV-16) entraîne l'amortissement de la rugosité

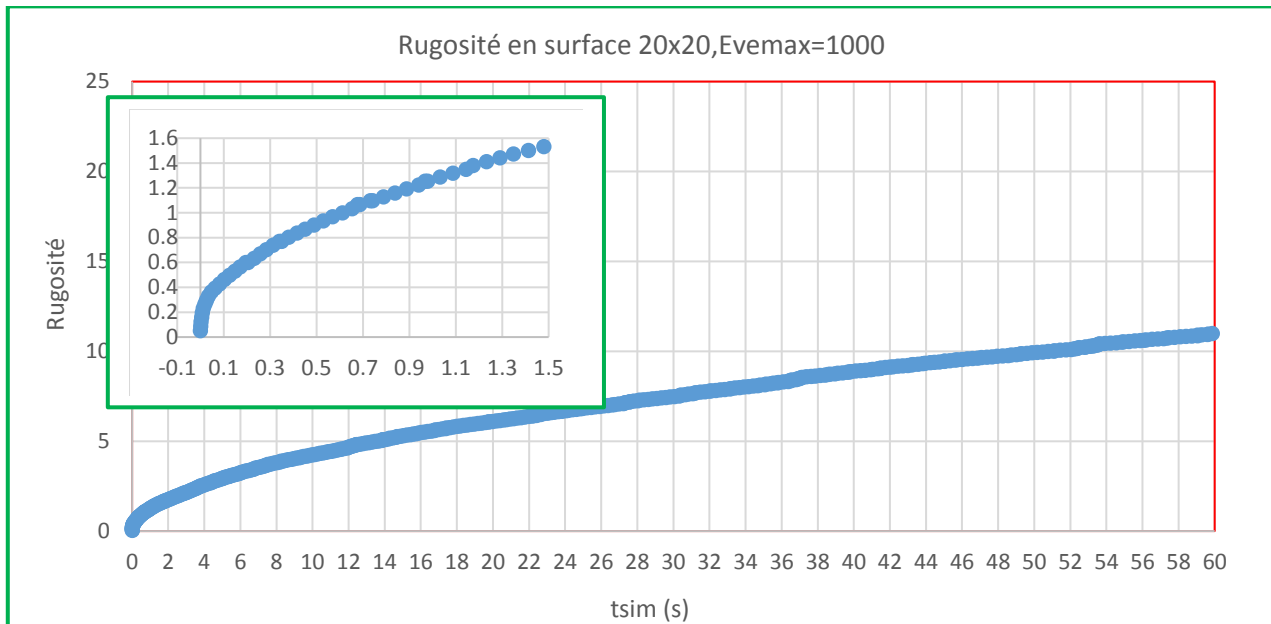


Figure.IV.17 .Représentation de la rugosité en surface en fonction du temps de simulation pour une surface de (20x20) sites de notre simulateur version 2.

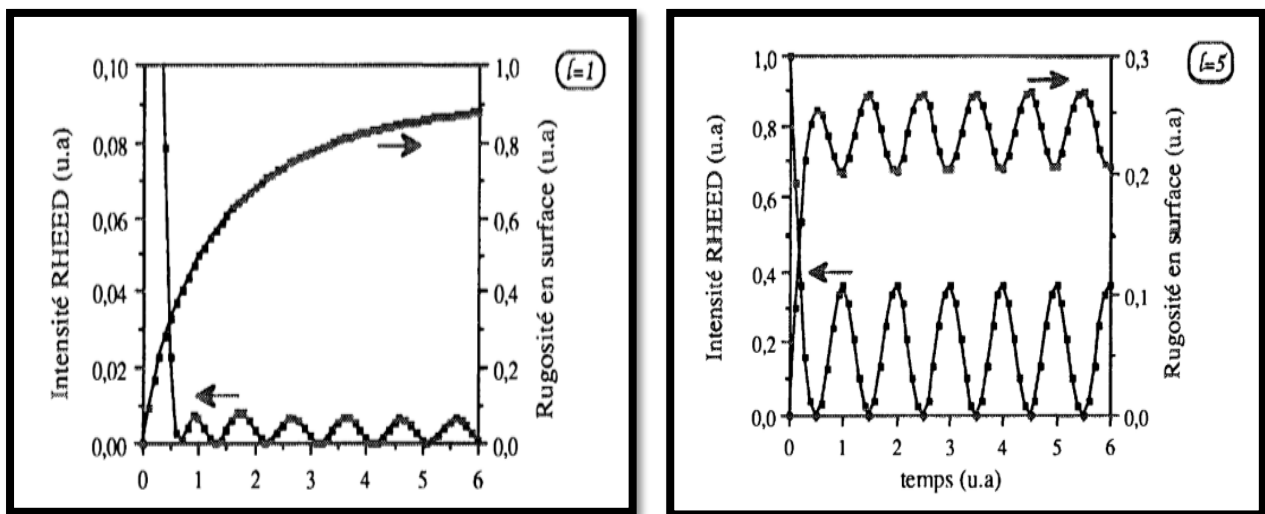


Figure .IV.18. Evolution de la rugosité temporelle (simulation) pour une croissance (a) d'une monocouche (b) de plusieurs couches[27]

Par comparaison aux travaux expérimentaux et de simulation de certains auteurs telle que A. Madhukar et S.V. Ghaisas[34], cette rapide augmentation de la rugosité en surface, lors de la phase initiale de croissance d'une monocouche, est en accord avec ces derniers et son allure coïncide avec l'approche et les observations représentées sur la figure IV-18.

#### IV.4. Conclusion

Le simulateur SPARCC-BT (Simulateur Particulaire de la Croissance Cristalline à Basse Température) ainsi développé, en ses deux versions, via la méthode de Monté Carlo cinétique, utilisant :

- ❖ L'aspect aléatoire aussi bien dans la gestion de l'interaction Gaz/Surface que pour certains mécanismes de diffusions équiprobables en surface,
- ❖ et l'aspect déterministe dans la gestion énergétique de certaines configurations de surface( valeurs issues via la méthode DFT- Annexe 2)

a permis l'analyse des phénomènes pouvant perturber le processus de croissance cristalline d'une monocouche, et chaque version a formulé plusieurs hypothèses pour expliquer ses observations, tout en restant qualitatives et en accord avec les résultats théorique et empiriques.

Une étude plus détaillée de l'aspect et de l'origine de certaines observations sera présentée dans des travaux ultérieurs.

# Conclusion Générale

Les matériaux qui remplissent leur fonction comme prévu pendant toute la vie sont d'une importance cruciale pour les technologues. Pour cela, il faut d'abord savoir les élaborer, du stade du laboratoire au stade industriel, puis les caractériser, depuis l'échelle macroscopique jusqu'à l'échelle atomique.

Avec l'augmentation de la puissance des supercalculateurs, il est maintenant possible d'espérer développer des modélisation-simulations, à différentes échelles, qui serviront de cadre pour comprendre, prévoir, guider la conception expérimentale des matériaux.

Notre démarche dans ce mémoire de master en physique des matériaux, concerne la simulation, par la méthode Monte Carlo cinétique, de la croissance cristalline des matériaux cristallins en couches minces, comme une nouvelle perspective technologique de suivi in-situ à l'échelle atomique. La sensibilité de cette méthode à la structure cristalline de la surface et à la composition du gaz la rend tout à fait adaptée à l'étude du processus d'épitaxie et ainsi à devenir une technique complémentaire aux techniques de caractérisation réelles.

Dans notre mémoire, nous avons tenté d'argumenter en faveur de la technique de Monte Carlo cinétique pour le suivi en temps réel de la croissance cristalline en couches minces.

Notre objectif était double et consistait d'une part à développer un outil informatique (en langage C++) par la méthode KMC permettant de gérer l'aspect aléatoire des « arrivées » des atomes du gaz sur les sites du substrat et d'autre part à valider, en fin de développement, les résultats de cet outil réalisé par des modèles théoriques ou résultats expérimentaux.

Nous avons alors consacré l'essentiel de nos efforts à explorer des pistes pour le choix, à l'échelle atomique, des mécanismes élémentaires (dits « évènement ») compatibles aussi bien avec la méthode KMC qu'avec la dynamique de croissance. Nous avons abouti à mettre en œuvre deux approches l'une basée sur le mécanisme d'« adsorption » seul et l'autre basée sur des considérations de la réactivité de surface en cours de croissance,

en introduisant un deuxième mécanisme qui est la migration en surface (ou « diffusion ») de l'adatom avant son incorporation dans les amas ou îlot formés pendant le temps de simulation. Dans chacune de ces approches, le modèle de croissance a été validé par des données théoriques (isothermes pour le modèle d'adsorption) ou expérimentales (Rugosité et taux de couverture maximal)..

La première approche correspond à un simulateur que nous avons dénommé SPARCC-BT, qui utilise la méthode de Monte Carlo pour reproduire le mouvement aléatoire de chacun des atomes du gaz présents sur la surface, durant toute la période de croissance homoépitaxiale.

La seconde approche élaborée est plus fine et correspond à un simulateur que nous avons dénommé SPARCC-DBT. Ce dernier est basé sur la théorie de la cinétique chimique qui exprime la réactivité des atomes adsorbés avec surface et décrit la formation des amas en surface et les diffusions intracouches.

Soulignons enfin que ces outils de simulation de la croissance ne constituent, à l'heure actuelle, qu'un outil de première expérimentation qui doit préparer un outil de seconde génération plus performant, sous ultra-vide, et reposant sur les mêmes principes de base que les deux premiers.

Enfin, ce sujet a fait l'objet de plusieurs thèses de doctorat, en simulation, au laboratoire d'analyse et d'architecture des systèmes-LAAS-CNRS à Toulouse. Ces thèses se sont limitées jusqu'ici à des analyses de la croissance cristalline des couches minces obtenues par épitaxie par jets moléculaire [33, 38] ou oxydation thermique du silicium [39,40]

## Perspectives

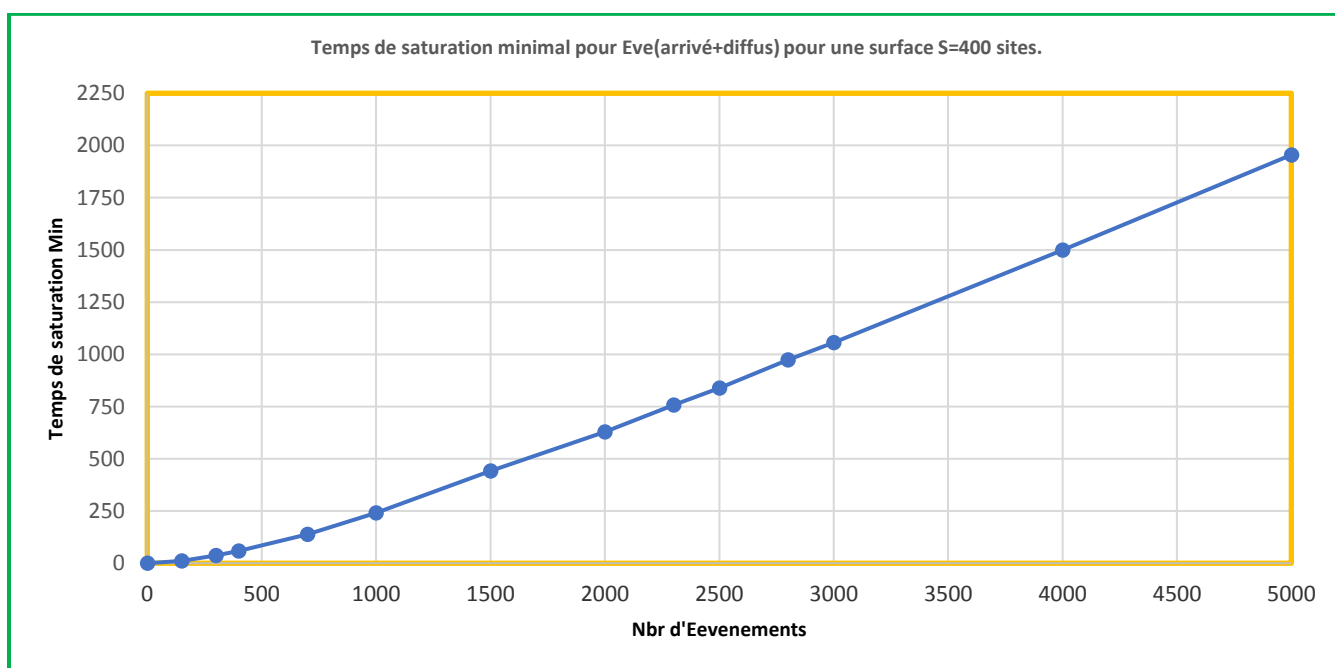
- 1/ Ces logiciels de simulation de la croissance à partir d'espèces atomique dans la phase gazeuse, peuvent être adapté à un système de croissance bien défini et améliorés par l'introduction des configurations locales favorables à l'incorporation des atomes ou molécules du système étudié.
- 2/ Introduire la notion des échanges intercouches par saut de marches est aussi un mode de croissance conditionné par les propriétés de diffusion des atomes, à prévoir son étude ultérieurement.
- 3/Prévoir un support pour une représentation graphique animée à trois dimensions. Cette représentation est un moyen utile et commode pour le traitement des résultats obtenus (à travers un fichier en output "résultat.xls" qui est très volumineux) et pour une meilleure compréhension de la croissance cristalline.

# Annexe A

## Résultat du simulateur – version 2

### Relation entre le temps minimale de saturation de la couverture et le nombre d'évènements

#### Représentation du temps minimale de saturation en fonction du nombre d'évènement

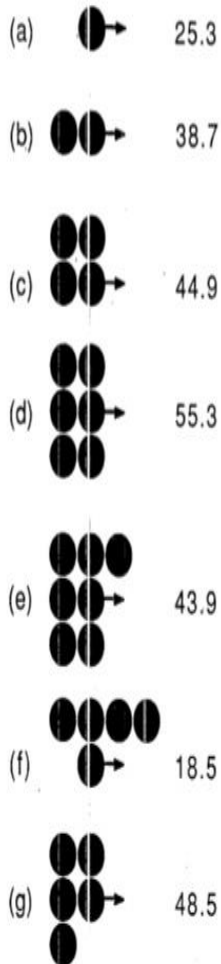


Cette évolution du temps minimal de saturation en fonction du nombre d'évènement est obtenue par notre simulateur **SPARCC-DBT** (*Simulateur PARTiculaire de la croissance cristalline avec Diffusion prise à Basse avec Température*) comme courbe d'étalonnage du temps de simulation (temps fictif de la machine) par le nombre d'évènement.

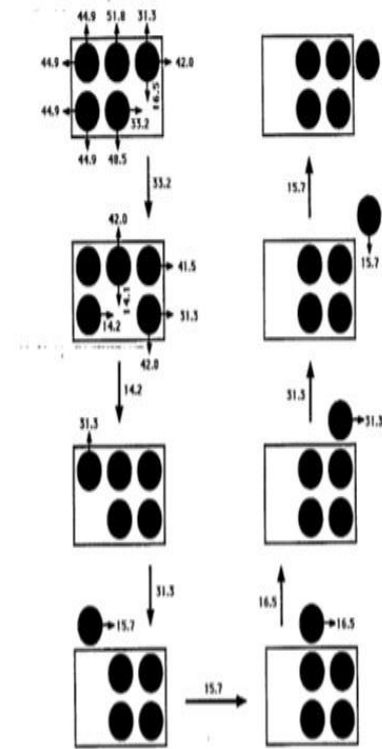
Selon cette courbe, la loi est linéaire entre le temps et le nombre des évènements lorsque le nombre est supérieur 500 pour une surface de croissance de 400 sites en surface.

# Annexe B

## Modèles Energétiques et estimation des énergies d'activation de la diffusion des adatoms en surface (DFT) [41]

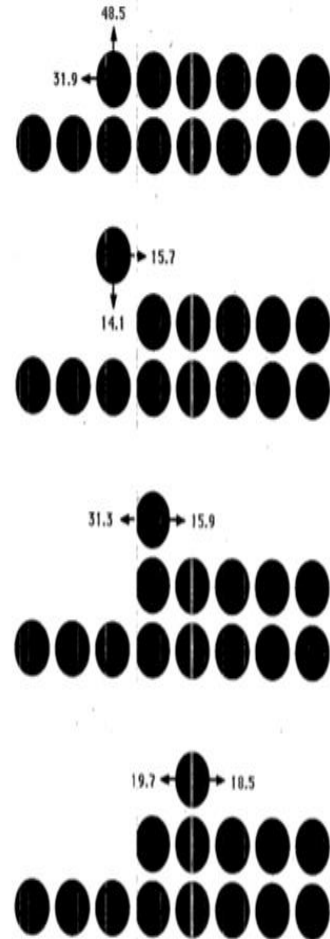


Some representative activation barriers (kcal/mol) from the rate catalog. All the catalog values for  $\nu_0$  [Eq. (16)] are between  $4.8 \times 10^{12}$  and  $1.2 \times 10^{13} \text{ sec}^{-1}$ .



each rate constant is computed from

$$k = \nu_0 \exp(-E_{\text{act}}/k_B T) \quad \text{Eq. (16)}$$



Mechanism for an atom to climb onto a fresh edge from a kink site. The maximum barrier is 48.5 kcal/mol.



**BBM : Break Bond Model**

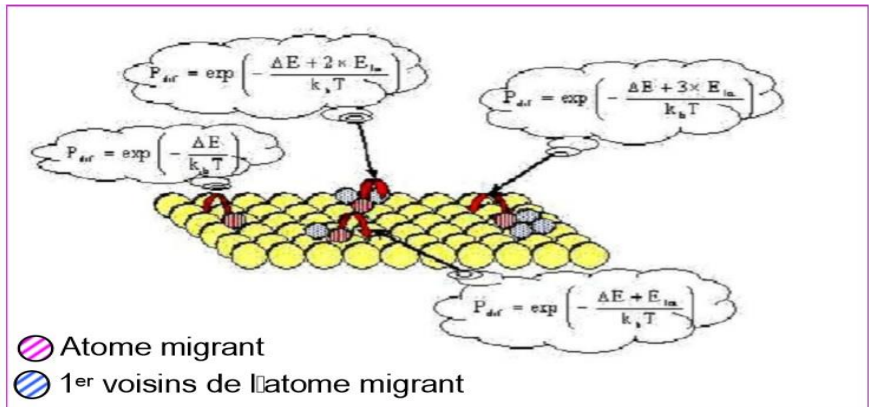
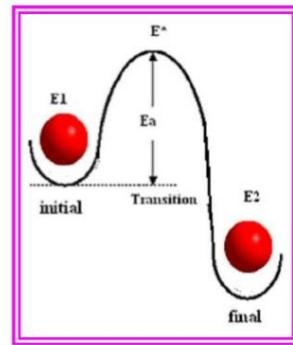
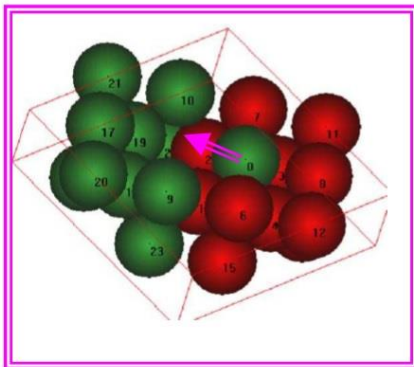


Figure : diffusion d'atomes et modèle de calcul en BBM  
**E<sub>d</sub>**: énergie de diffusion  
**ΔE** : variation d'énergie du cristal



# Annexe C

## //Fichier : SPARCCV1.cpp

// Programme du Simulateur Particulaire de la Croissance Cristalline à Basse Température -Version 1  
// Application à la simulation de l'interaction Gaz/Surface par Adsorption, via la méthode KMC  
// C++

```
#include "Mesdonnees.h" //Fichier contenant toutes les bibliothèques et constantes
```

```
//*****
```

```
//.....PROGRAMME PRINCIPAL.....
```

```
//*****
```

```
main()
```

```
{
```

```
//-----
```

```
//.....DECLARATION DES VARIABLES .....
```

```
//-----
```

```
int i, j; //...Variables pour les boucles 'for' ....
```

```
int SURF=L*C, S[L][C]; //...Variable pour la configuration géométrique du substrat....
```

```
double z, t[L][C]; //...Variables pour le calcul du temps 'loi de poisson'....
```

```
double log( double z ); //...Fonction 'log' pour le calcul du temps 'loi de poisson'....
```

```
int imin, jmin;
```

```
double tmin=0, tsim=0; // ...Declaration des temps de simulations
```

```
int eve=0, eveA=0;
```

```
double Tsimul[Evemax+1];
```

```
// En c++ les tableaux et vecteurs reservent un espace de mémoire entre 0 et Evemax-1
```

```
int n=0, lmini[Evemax+1], jmini[Evemax+1], even[Evemax+1],evenA[Evemax+1];
```

```
FILE *f1,*f2; //...Declaration de Fichier externe.....
```

```
//*****
```

```
// .....affichage de la dimension de la surface.....
```

```
//*****
```

```
printf("SURF=%d\n",SURF);
```

```
getch ();
```

```
//*****
```

```
//....Initialisation des vecteurs d'enregistrement des données pour analyse....
```

```
//*****
```

```
for (i=0; i <= Evemax; i++){
```

```
    Tsimul[i]=0.0;
```

```
    lmini[i]=0;
```

```
    jmini[i]=0;
```

```
    even[i]=0;
```

```
    evenA[i]=0;
```

```
}
```

```
//*****
```

```
//....Initialisation des sites du substrat à INIT=Zero - remplir le tableau
```

```
//*****
```

```

for (i=0; i <L; i++)
for (j=0; j <C; j++)
    S[i][j]=INIT;           //INIT=0

//*****
//*****
// Le temps affecté à chaque site de la surface (calendrier des atomes ou molecules dites "des arrivée " sur la
//surface de croissance & rappel de la formule du GNA entre 0 et 1
//*****
for(i=0; i <L; i++){
for(j=0; j <C; j++){
t[i][j]=INIT;           // Initialisation à zero
z = (double) rand()/RAND_MAX; // z=rand()/RAND_max
t[i][j]=-0.5*log(z);
    }
//*****
// recherche du minimum des temps "tmin" affecté aux sites (imin , jmin) de la surface
//*****
tmin =t[0][0];
for (i=0; i <L; i++){
for (j=0; j <C; j++){
if(t[i][j]<=tmin){
tmin= t[i][j];
imin=i;
jmin=j;
}}}

//*****
// nouvelle configuration de la surface
//*****
S[imin][jmin]=1+S[imin][jmin]; // la configuration du site sera modifiée

//*****
// calcul du temps d'occurrence (tsim)des evenements et le type,nombre des evenement
//*****
tsim=tsim+tmin;
eve=eve+1;
//*****
// type d'évènement
//*****
if(S[imin][jmin]==1) eveA=eveA+1;
//*****
// Enregistrement des données tmin , imin,jmin dans des tableau et vecteurs pour exploitation en excel
//*****
n=1;
Tsimul[n]=tsim;
Imini[n]=imin;
Jmini[n]=jmin;
even[n]=eve;
evenA[n]=eveA;

```

```

//-----
// plus de la 2eme boucle de calculs , configuration de surface et actualisation des temps (tmin, tsim)et du nombre
//des evenements (eve, eveA et eveR)
//-----
while(eve<=Evemax)
{
    //*****
    // recherche du minimum des temps "tmin" affecté aux sites (imin , jmin) de la surface
    //*****
    tmin=t[0][0];
    for (i=0; i < L; i++){
    for (j=0; j < C; j++){
        if(t[i][j]<=tmin){
            tmin= t[i][j];
            imin=i;
            jmin=j;
        }
    }
    //*****
    // nouvelle configuration de la surface
    //*****
    S[imin][jmin]=S[imin][jmin]+1;          / / la configuration du site sera modifiée si le site est vacant
    if(i==imin && j==jmin) S[imin][jmin]=1+S[imin][jmin];

//*****
// nouveau calendrier affecté à la surface avec un nouveau temps affecté au site (imin,jmin)
//*****
    z = (double) rand()/RAND_MAX;
    t[imin][jmin]=-0.5*log(z);

//*****
// Affichage du temps d'occurrence (tsim)des evenements et le type,nombre des evenement
//*****
    tsim=tsim+tmin;
    eve=eve+1;
    //*****
    // type d'evenement
    //*****
    if(S[imin][jmin]==1)eveA=eveA+1;

//*****
// Enregistrement des données tmin , imin,jmin dans des tableau et vecteurs pour exploitation en excel
//*****
    n=n+1;
    Tsimul[n]=tsim;
    lmini[n]=imin;
    Jmini[n]=jmin;
    even[n]=eve;
    evenA[n]=eveA;
}

```

```

}
//*****
//Ecriture des resultats dans un fichier .xls
//*****
f1 = fopen("resultat2.xls", "w"); //xls
if (f1 == NULL) printf("Impossible d'ouvrir le fichier en écriture !");
else {
    fprintf(f1, " n \tTsimul \tImini\tJmini\teve\teveA\n");
    for (i=1;i<=Evemax;i++){
        fprintf(f1, " %d \t%8f \t%d \t%d \t%d \t%d\n",i,Tsimul[i],Imini[i],Jmini[i],even[i],evenA[i]);
    }
}
fclose (f1);

//-----FIN 1-----
return 0;
}

//-----FIN 2-----

```

## // Fichier Mesdonnees.h

### //1ere partie du programme : Introduction des Bibliotheques

```

#include <stdio.h> /*bibliotheque contenant l'instruction printf , for ; int ... */
#include <conio.h> /*bibliotheque contenant l'instruction getch () */
#include <stdlib.h> /*bibliotheque contenant l'instruction rand et ran_max () */
#include <math.h> /*bibliotheque contenant l'instruction ln (logarithme) */

```

### // 2eme partie du programme : Introduction des constantes

```

const int L= 20; //5; //2;
const int C= 20; //5; //3;
const int INIT=0;
const float Timemax =200; //120.0; // secondes
const int Evemax=1000; //100; //500; //L*C;

```