

Un modèle de formation des cristaux de neige

Melissa Mansour, Julian Merle, Nicolas Ruche, Océane Trazie

Université Pierre et Marie Curie

ARE Dynamic 2017-2018

Résumé

Au cours de ce semestre, notre ambition a été de réussir à modéliser mathématiquement puis informatiquement des flocons de neiges. Nous nous sommes imposé une modélisation assez réaliste et proche du modèle physique afin de réussir à simuler la formation de flocons de différentes formes observables dans la nature.

Nous avons conçu un automate cellulaire sous forme de grille où à chaque cellule est attribué un niveau de vapeur. La mise à jour du modèle se base sur l'équation de diffusion, bien connue des physiciens.

Le modèle que nous avons choisi est celui de Reiter, le plus populaire et utilisé pour la modélisation de flocons car il permet d'obtenir des résultats suffisamment fidèles à la réalité.

[Melissa Mansour, relecteur : Nicolas Ruche]

"Si éclairant soient les grands textes, ils donnent moins de lumière que les premiers flocons de neige." - Christian Bobin, écrivain et poète français.

Table des matières

I	Présentation du modèle	6
1.	Modèle physique	6
a)	Pression de vapeur saturante	6
b)	Effet Bergeron	6
c)	Équation de diffusion	7
2.	Modèle mathématique	7
a)	Discrétisation de l'équation de diffusion	7
b)	Le modèle de Reiter	7
II	Contributions	9
1.	Grille hexagonale	9
a)	Module numpy	9
b)	Espace vectoriel et hexagrid	10
2.	Modèle	10
3.	Vue et Contrôleur	10
III	Interprétation des résultats	12
IV	Conclusion	16

Introduction

Les flocons de neige sont formés de petits cristaux de glace (entre 0.2 et 2 mm de diamètre) qui se forment au sein des nuages. En effet, trois conditions essentielles doivent être réunies pour permettre la formation d'un flocon : la présence de vapeur d'eau, une température plus basse que 0 ° C et la présence de minuscules poussières.

C'est au contact de ces dernières que la vapeur d'eau ambiante se condense, et si la température est suffisamment basse, prend la forme d'un microscopique germe de glace (de l'ordre de 10 μm) qui croît grâce à l'effet Bergeron jusqu'à donner naissance au flocon.

Une de nos problématiques majeures a été de trouver le support d'affichage adéquat pour les flocons que nous modéliserons. Nous nous sommes rendu compte qu'utiliser une grille hexagonale simplifierait bien largement les calculs et permettrait une meilleure représentation des flocons qu'avec une matrice carrée qui fut la première option nous venant à l'idée.

Notre équipe se compose de Melissa Mansour, Julian Merle-Remond, Nicolas Ruche et Océane Trazie.

Nous allons donc commencer par présenter le modèle de Reiter sur lequel nous avons basé notre travail. Nous verrons ensuite les différentes contributions de l'équipe à ce projet et finiront par exposer les principales observations comportementales du modèle que nous avons pu faire.

[Melissa Mansour, relecteur : Nicolas Ruche]

Chapitre I

Présentation du modèle

Le flocon de neige se forme au sein des nuages par condensation de vapeur sur une micro-poussière naturellement présente dans l'atmosphère. On observe dans la nature trois grandes formes de flocons représentables en 2 dimensions : les plaquettes, les dendrites étoilées et les dendrites à rayons (voir annexe [1]).

Le modèle repose sur trois phénomènes physiques importants : la pression de vapeur saturante, l'effet Bergeron et l'équation de diffusion.

[*Melissa Mansour, relecteur : Océane Trazie*]

1. Modèle physique

a) Pression de vapeur saturante

Dans notre cas, le germe s'apparente à un corps pur (composé uniquement d'eau).

La pression de vapeur saturante est la pression de la phase gazeuse de l'eau lorsqu'elle est en équilibre thermodynamique avec la phase solide de l'eau. Si la pression partielle (pression qui serait exercée par les molécules d'eau elles occupaient seules tout le volume offert) dépasse la pression de vapeur saturante, l'eau condense pour devenir solide.

Ainsi, auprès d'un point gelé, si suffisamment d'eau s'accumule autour de celui-ci, la pression partielle dépasse la pression de vapeur saturante, et la vapeur se solidifie sur ce point.

[*Nicolas Ruche, relecteur : Melissa Mansour*]

b) Effet Bergeron

Dans un milieu condensé où coexistent à température négative des cristaux de glace et des gouttelettes d'eau en état de surfusion, ces dernières se vaporisent, puis se condensent à l'état solide au contact des cristaux, du fait que la pression de vapeur saturante de la vapeur d'eau est moins élevée par rapport à la glace que par rapport à l'eau liquide (*météofrance.fr*).

Ceci nous indique qu'auprès d'un cristal de glace, la vapeur a plus tendance à se condenser.

[*Melissa Mansour, relecteur : Océane Trazie*]

c) Équation de diffusion

Les changements de concentration en vapeur d'eau suivent l'équation de diffusion :

$$\frac{\partial c(x, y, t)}{\partial t} = \frac{1}{12} \left(\frac{\partial^2 c(x, y, t)}{\partial x^2} + \frac{\partial^2 c(x, y, t)}{\partial y^2} \right)$$

Avec $c(x, y, t)$, la concentration de vapeur au point (x, y) à l'instant t . La constante $\frac{1}{12}$ est la constante de diffusion, on choisit ce nombre pour des simplifications qui vont s'opérer. Cela veut dire que plus l'écart entre la concentration en vapeur d'un point, et celle de ses voisins, est grand, plus celui-ci aura tendance à être comblé pour atteindre un équilibre (voir annexe [2]).

[Nicolas Ruche, relecteur : Melissa Mansour]

2. Modèle mathématique

a) Discrétisation de l'équation de diffusion

Chaque cellule hexagonale possède une concentration c en vapeur d'eau. L'équation :

$$\frac{\partial c(q, r, t)}{\partial t} = \frac{1}{12} \left(\frac{\partial^2 c(q, r, t)}{\partial q^2} + \frac{\partial^2 c(q, r, t)}{\partial r^2} \right) \quad (4)$$

peut se discrétiser comme suit, d'après la méthode des différences finies :

$$\frac{c(q, r, t+1) - c(q, r, t)}{t+1-t} = \frac{1}{12} (c(q+1, r, t) + c(q-1, r, t) + c(q, r+1, t) + c(q, r-1, t) + c(q-1, r+1, t) + c(q+1, r-1, t) - 6 \times c(q, r, t)) \quad (I.1)$$

$$+ c(q+1, r-1, t) - 6 \times c(q, r, t)) \quad (I.2)$$

On pose :

$$V(c) = c(q+1, r, t) + c(q-1, r, t) + c(q, r+1, t) + c(q, r-1, t) + c(q-1, r+1, t) + c(q+1, r-1, t) \quad (6)$$

(5) devient après simplification :

$$c(q, r, t+1) = \frac{1}{2} c(q, r, t) + \frac{1}{12} V(c) \quad (6)$$

[Nicolas Ruche, relecteur : Melissa Mansour]

b) Le modèle de Reiter

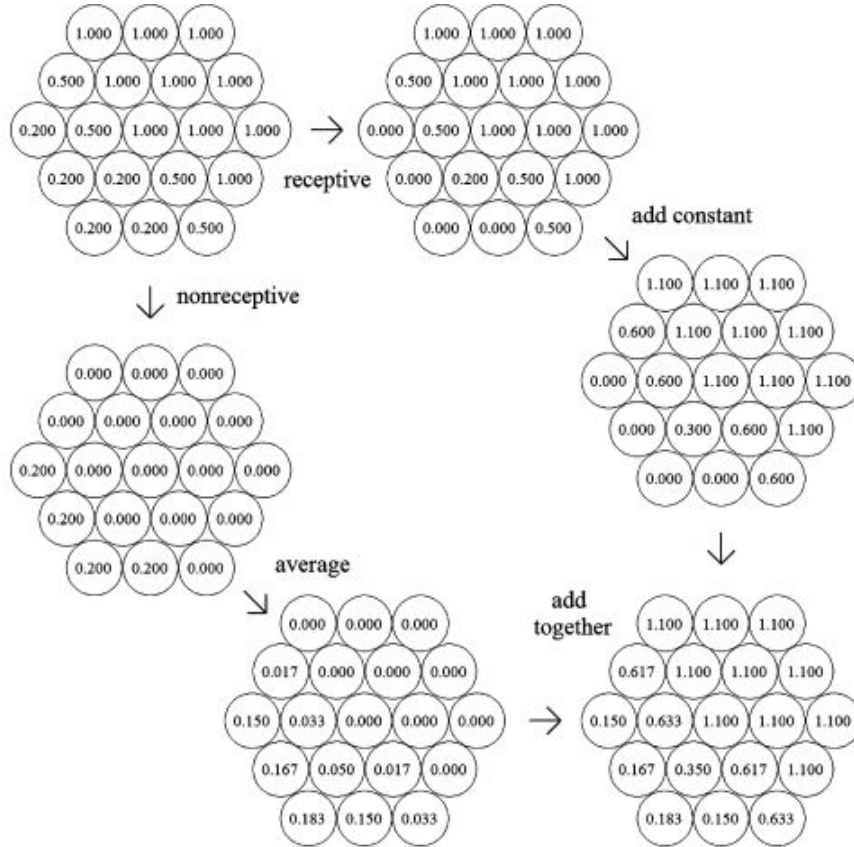
La concentration en vapeur d'eau est représentée par un nombre réel positif, qu'on appelle état. On initialise une grille hexagonale, dont chaque état vaut une constante β (comprise entre 0 et 1), sauf la cellule centrale, représentant le germe, que l'on met à 1.

On considère qu'une cellule est gelée si son état est 1 ou plus. On dit qu'une cellule est réceptive si elle a au moins une cellule voisine gelée.

On considère qu'une cellule est gelée si son état vaut 1 ou plus (la pression partielle a dépassé la pression de vapeur saturante). On dit qu'une cellule est réceptive si elle a au moins une cellule voisine gelée. On impose aux cellules se trouvant au bord de rester à la valeur de β afin que la valeur de la vapeur environnante est continuellement un impact sur la formation du flocon.

Pour simuler la diffusion de la vapeur : On applique l'équation (6) sur chacune des cellules (en prenant garde de ne pas les mettre à jour une par une mais toutes en même temps!), en comptant l'état des cellules réceptives égal à 0. (voir annexe [4])

D'après l'effet Bergeron, l'état des cellules réceptives doit augmenter, car celles-ci sont près d'un site glacé. Ainsi on ajoute une constante dite d'addition γ aux cellules qui étaient réceptives avant d'appliquer l'équation de diffusion. Cette constante d'addition simule l'échange de matière entre les cellules dont la frontière est bien évidemment imaginaire et ne sert qu'à la discrétisation du plan.



[Océane Trazie, relecteur : Julian Merle]

Chapitre II

Contributions

Pour notre programme, nous avons choisi d'utiliser un motif d'architecture logicielle destiné aux interfaces graphiques lancé en 1978 et très populaire pour les applications web : le Model-View-Controller. Notre programme est donc composé de 3 modules ayant chacun un rôle bien précis : le modèle, qui contient les données à afficher ; la vue, qui correspond à l'interface graphique visible par l'utilisateur et le contrôleur, qui traite les données de l'utilisateur, il dépend et interagit avec le modèle ainsi que la vue.

Aussi, nous avons comme précisé précédemment utilisé une méthode de grille hexagonale que nous avons nommé HexaGrid dans notre code et qui sera présentée plus en détail dans les parties qui suivent. [*Melissa Mansour, relecteur : Océane Trazie*]

1. Grille hexagonale

a) Module numpy

Numpy est un module python permettant de calculer rapidement des résultats mathématiques trop lourds pour être implémentés en python pur efficacement, ce premier étant en effet écrit en C. Ce module permet en outre de gérer facilement et rapidement les matrices de nombres.

Cette propriété intéressante le rendait parfaitement adapté à notre travail : en effet, la grille hexagonale peut être décrite par coordonnées en deux dimensions se prêtant particulièrement à ce style de stockage.

Le traitement des vecteurs par l'interface *numpy* aurait surement été très efficace mais la grille a été réalisée sans utiliser ce module bien que beaucoup de tests probants aient été conduits.

En effet, par manque d'efficacité dans la mise en œuvre du logiciel final, l'implémentation a été abandonnée en partie due à des problèmes de compatibilité et de portabilité des ressources. La programmation ayant été réalisé sur des machines Linux et Windows, la non-portabilité des codes C compilés de numpy entre systèmes et versions de python empêchaient une utilisation aisée de la librairie.

Il a donc été choisi d'optimiser le code python standard pour notre usage. (voir annexe [3])

[*Julian Merle, relecteur : Melissa Mansour*]

b) Espace vectoriel et hexagrid

La première idée consistait à paver le plan d'un maillage carré. Néanmoins, pour les flocons adoptant une forme hexagonale, les résultats n'étaient que passables et parfois difformes. (voir schéma [2]) Nous avons donc opté pour une grille hexagonale qui épouse mieux la forme naturellement hexagonale des flocons de neige qui est due aux forces d'attractions et de répulsions des molécules d'eau. (voir schéma [...])

La grille est donc construite telle qu'elle soit assimilable à un hyperplan H tel que :

$$H = \{(x, y, z) \in \mathbb{Z}^3 \mid x + y + z = 0\}$$

Ceci est aussi dû au manque de liberté des particules (seulement 4 direction), visible sur l'image. Le plan est donc représenté par un maillage hexagonal. Pour un hexagone centré en l'origine de rayon h , les 6 coins ont pour coordonnées cartésiennes :

$$\left\{ \left(h \times \cos \left(\frac{\pi}{6} \times i \right), h \times \sin \left(\frac{\pi}{6} \times i \right) \right), i \in \llbracket 0, 1, 2, 3, 4, 5 \rrbracket \right\} \quad (1)$$

Chaque hexagone est défini par un couple (q, r) . Voir l'image [3] pour comprendre le système de coordonnées. Pour chaque hexagone de rayon h , son centre en coordonnées cartésiennes est donné par :

$$h \times \begin{pmatrix} \sqrt{3} & \frac{\sqrt{3}}{2} \\ 0 & \frac{3}{2} \end{pmatrix} \begin{pmatrix} q \\ r \end{pmatrix} \quad (2)$$

En combinant (1) et (2), on trouve que les coins de chaque hexagone ont pour coordonnées :

$$\left\{ \left(h \times \left(\cos \left(\frac{\pi}{6} \times i \right) + \sqrt{3} \times q + \frac{\sqrt{3}}{2} \times r \right), h \times \left(\sin \left(\frac{\pi}{6} \times i \right) + \frac{3}{2} \times r \right) \right), i \in \llbracket 0, 1, 2, 3, 4, 5 \rrbracket \right\} \quad (3)$$

[*Julian Merle, relecteur : Nicolas Ruche*]

2. Modèle

Cette partie du programme gère l'état des cellules de la grille. Elle a été très fortement optimisée tout au long de la phase de finalisation du programme.

Pour pouvoir représenter les flocons, nous avons discrétisé le plan (voir méthode hexagrid), ainsi le temps. Ce dernier est représenté par un entier naturel t qui est incrémenté de 1 à chaque nouvelle itération.

De plus, pendant cette phase, nous avons réussi à diviser le temps d'exécution par unité de temps par environ 6.5 (temps qui diffère selon les configurations).

Le gain de temps s'explique par la forte diminution du nombre de boucles itérant sur l'ensemble des cellules (seules 2 des 5 initiales furent conservées) et l'optimisation des boucles restantes en limitant les appels de fonctions.

[*Julian Merle, relecteur : Melissa Mansour*]

3. Vue et Contrôleur

Pour réaliser la vue, nous avons utilisé le module de gestion de fenêtres *tkinter*, intégré à la bibliothèque standard de Python.

Ce dernier nous a permis d'avoir une interface graphique offrant une passerelle entre Python et la bibliothèque *Tk*. L'avantage du canvas que nous avons créé est qu'il nous permet de dessiner notre flocon au pixel près, en définissant la résolution de l'affichage.

Afin d'optimiser ce processus, la bibliothèque *threading* a été utilisée afin de permettre l'exécution de tâches de fond tout en gardant l'interface graphique en interaction avec l'utilisateur. Les fonctions implémentées en exécution parallèle à l'aide de ce module sont par exemple le dessin des cellules en continu à l'écran, le calcul de l'étape temporelle suivante et la progression automatique.

Cette gestion permet l'implémentation de fonctionnalités liant le calcul et l'interface comme un bouton d'arrêt pour de la tâche en cours (bouton auto/stop de la progression automatique).

De plus, pour simplifier le code, nous avons créé les classes suivantes :

- Hexacell : représente une cellule hexagonale, celle-ci stocke sa position et son état
- HexaMap : initialise la grille et offre des méthodes pour accéder aux voisins d'une cellule, vérifiant l'écriture et la lecture d'une cellule
- Model : stocke les constantes du modèle et est responsable de la mise à jour de la grille
- Controller : définit les opérations permises sur le modèle (initialisation et mise à jour de la grille) Point, Orientation et Layout : Sont responsables des changements de bases entre la base hexagonale et l'écran
- Window : Stocke l'interface Tkinter et affiche la grille

(voir annexe [3])

[*Melissa Mansour, relecteur : Julian Merle*]

Chapitre III

Interprétation des résultats

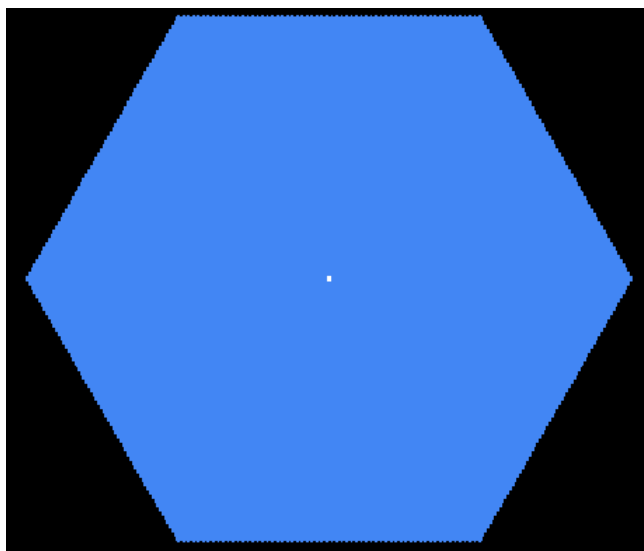
Nous rappelons que $0 \leq \beta \leq 0.95$ correspond au niveau de vapeur environnant et $0 \leq \gamma \leq 1$ à la constante d'addition d'eau.

Nous avons généré nos flocons sur une grille hexagonal d'environ 60×60 cellules.

Le nombre d'itérations avant d'obtenir le résultat final dépend des configurations, certaines convergeant plus rapidement vers leur état final.

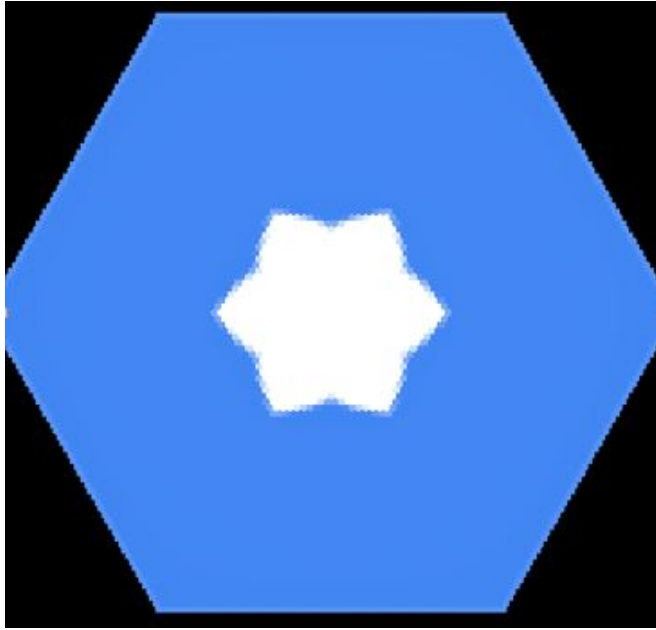
La partie claire correspond aux cellules dont la valeur est supérieure à 1 (soit à l'eau solide) et la partie plus foncée correspond aux cellules dont la valeur est inférieure à 1, avec le bleu correspond à un niveau 0 (eau liquide).

- $\beta = \gamma = 0$: Aucune formation n'apparaît. Ceci s'explique par le fait qu'il n'y ait ni humidité ni échange de matière d'une cellule à une autre : les conditions pour la formation d'un flocon ne sont pas réunies donc l'eau reste à l'état liquide et seule la cellule du milieu reste glacée.



$\beta=\gamma=0$, 500 itérations.

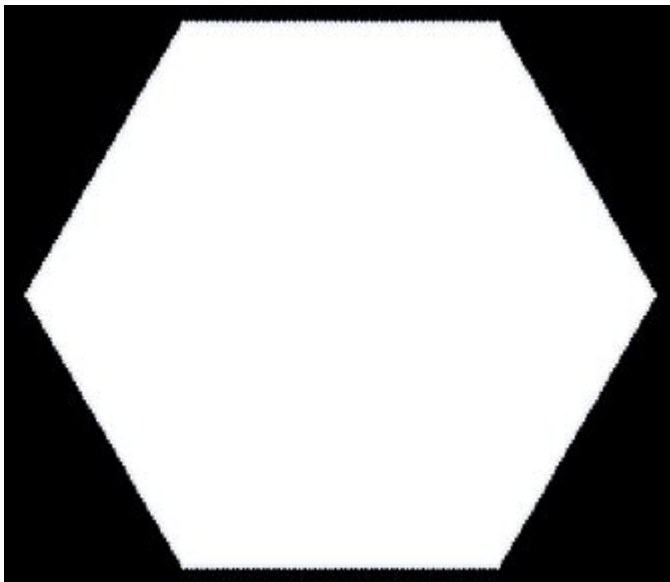
- $\beta = 0, \gamma > 0$: On obtient alors des plaquettes dont le diamètre augmente lorsque γ augmente. La croissance elle, est très lente à cause du fait que la vapeur provient uniquement de la condensation du germe, le niveau d'humidité étant trop faible pour que d'autres sites de glace se forment indépendamment.



$\beta=0.10, \gamma=0.0035, 2000$ itérations.

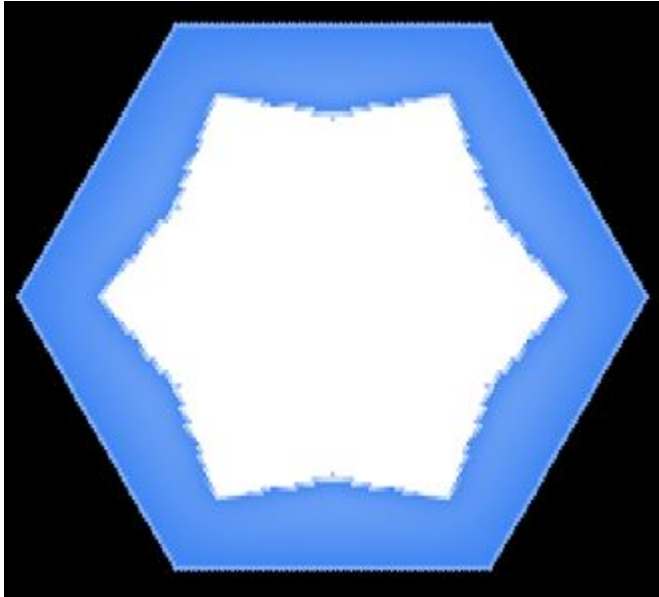
- $\beta > 0, \gamma$ très grand :

- $\gamma = 1$: On obtient uniquement des plaquettes (hexagones réguliers). Pour un même nombre d'itérations, le diamètre de plaquette est plus petit si la valeur de β est plus petite et inversement si la valeur de β est plus importante. Cependant, le diamètre à l'état final est maximum, et donc le même peu importe la valeur de β .



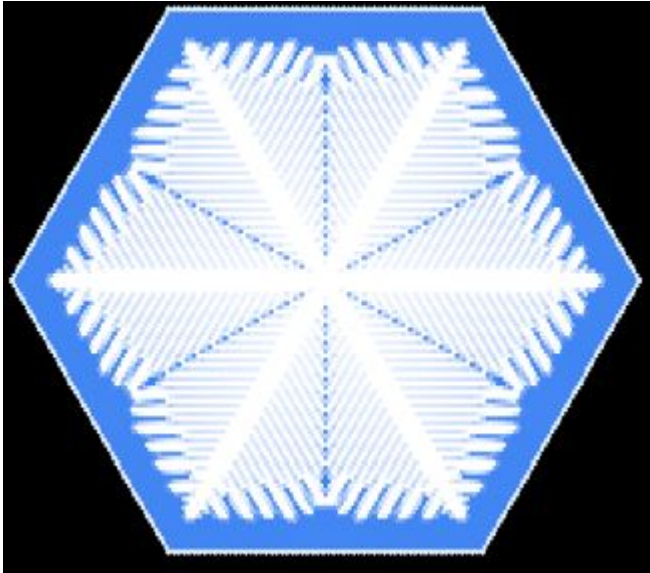
$\beta=0.90, \gamma=1.000, 150$ itérations.

- $\gamma < 1$: On obtient d'abord des plaquettes pour un β élevé, puis plus il diminue, plus le flocon tend plus vers une forme dendritique étoilée.



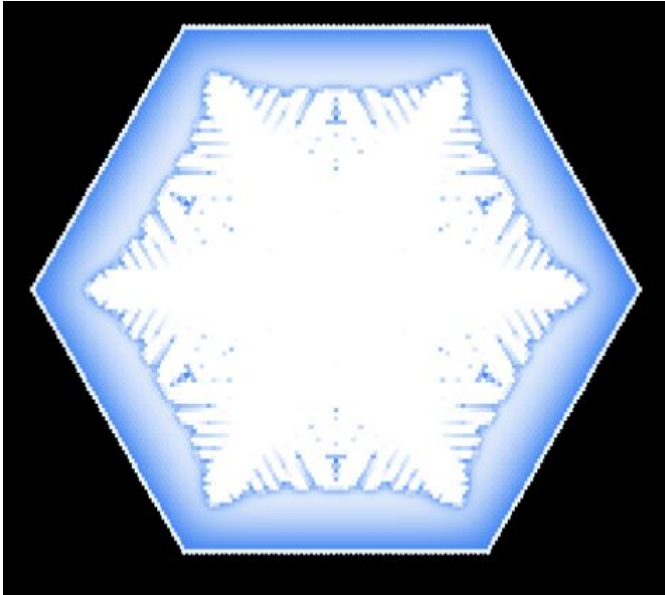
$\beta=0,30 ; \gamma=1$, 500 itérations.

- β, γ très petits : La formation du flocon est lente et minutieuse, les flocons obtenus sont alors essentiellement des dendrites à rayons.

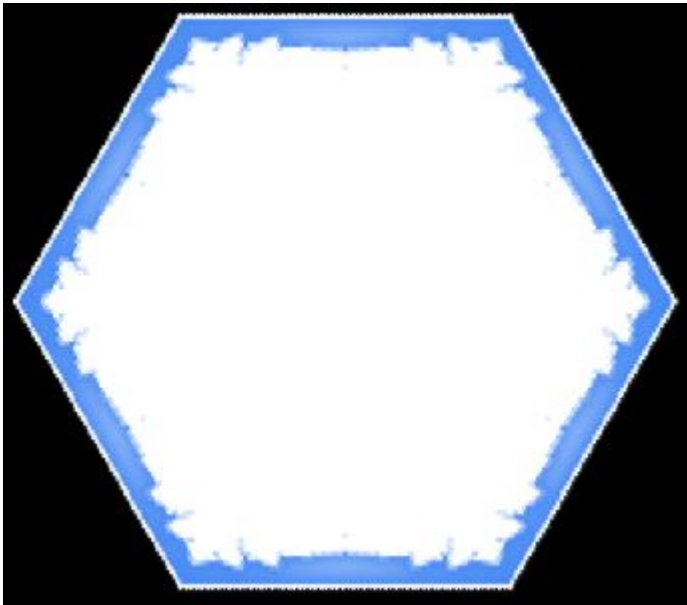


$\beta=0,40 ; \gamma=0.001$, 450 itérations.

- $\gamma > 0, \beta$ très grand : On obtient alors des formes abstraites de dendrites étoilées qui varient assez aléatoirement en fonction des valeurs de β et γ .
Du fait que le niveau vapeur environnante soit très élevé, la croissance est plutôt rapide. avec cette configuration, on obtient un nombre impressionnant de formes dendritiques différentes ce qui nous apparaît comme remarquablement réaliste.



$\beta=0,90$; $\gamma=0.95$, 150 itérations.



$\beta=0,95$; $\gamma=0.999$, 150 itérations.

[*Melissa Mansour, relecteur : Océane Trazie*]

Chapitre IV

Conclusion

Nous avons expliqué physiquement la pertinence du modèle, ainsi que codé une simulation optimisée du celui-ci à l'aide du multi-threading. Cette expérience nous a permis de mieux maîtriser l'outil git, qui a grandement facilité notre travail d'équipe.

Nous avons aussi pu appréhender les opérateurs différentiels tels que les gradients, divergents et laplaciens, qui nous ont permis de mieux comprendre l'équation de diffusion, élément-clé du modèle.

Enfin la discrétisation du plan nous a permis de travailler sur une grille peu habituelle, mais pourtant très utile.

Abstract

During this semester, our ambition was to succeed in mathematically and computationally modelize snow crystal growth. We imposed ourselves to use a fairly realistic model in order to approach the physical model and successfully simulate the formation of different snowflakes shapes that can be observed in nature.

Therefore, we designed a cellular automaton in the form of a hexagonal grid where a vapor level is assigned to each cell. The update of the model is based on the well known by physicists diffusion equation.

The model we chose is the popular Reiter's model, using a combined approach of mathematical analysis and numerical simulation.

Indeed our work combined to this model allowed us to obtain results that we judged sufficiently faithful to reality. [*Océane Trazie, relecteur : Julian Merle*]

Annexes



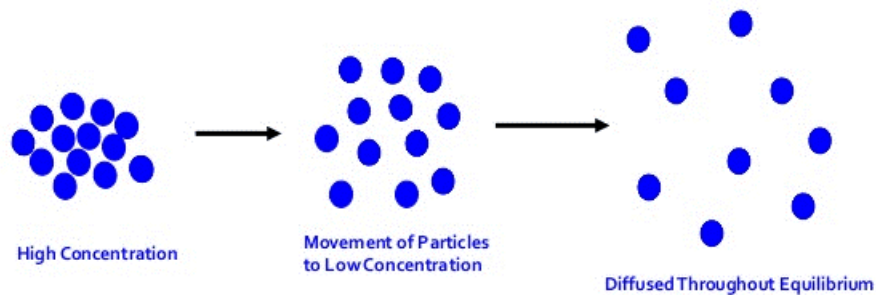
Types of Snowflakes ... SnowCrystals.com

Annexe 1

Diffusion

Particle net movement from;

a region of **HIGH** concentration to a region of **LOW** concentration



Annexe 2

```

class HexaMap:
    __slots__ = ("radius", "cells", "nbCellsWidth")

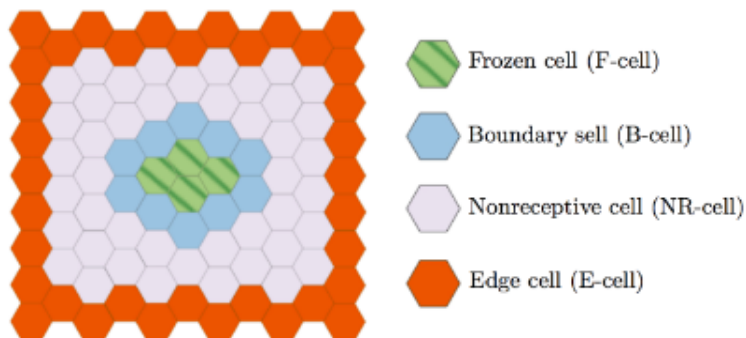
    def __init__(self, radius):
        self.radius = radius
        self.cells = dict()
        self.nbCellsWidth = 2 * self.radius + 1

        self._InitMap()

```

3.JPG

Annexe 3



Annexe 4