

Rapport de projet

TER N°2 : Simulations

Hoël ROQUINARC'H
Paul HERCOUËT

Tables des matières :

Tables des matières :	2
Contexte :	3
Explication de l'expérience :.....	3
Description du projet :	3
Les deux approches.....	3
L'approche population centrée.....	4
L'approche entité centrée.....	4
Implémentation :	5
Introduction.....	5
Structure de données.....	6
Lecture du fichier.....	6
Analyseur lexical.....	6
Analyseur syntaxique.....	7
Simulateur population centré :.....	9
Simulateur entité centrée :.....	10
Résultats.....	11
Utilisation :	11
Discussions :	12

Contexte :

Notre projet de TER consiste à simuler les réactions produites entre des molécules dans une vésicule dans l'objectif de tester les biomarqueurs correspondant à certaines maladies. Le but est de tester les biomarqueurs de manière booléenne afin de déterminer si une personne est atteinte d'une certaine maladie selon les taux des différents biomarqueurs. Cette technique consiste à mesurer la concentration d'un biomarqueur, si celle-ci est supérieure à un seuil fixé alors la personne peut être déclarée malade.

Explication de l'expérience :

Dans une vésicule, un récipient en forme de capsule mesurant de 50 à 100 microns, on met des enzymes, dans notre cas des protéines de 5 à 10 nm, ainsi que des produits chimiques, ici des métabolites de 1 nm. Certains métabolites vont passer à travers de la membrane de vésicule, par contre rien ne pourra en sortir. Sur la membrane, on place des grosses protéines (des protéines transmembranaires) qui sont composés de 3 domaines, ces protéines sont appelées des récepteurs, ils peuvent fixer et reconnaître certains biomarqueurs, modifier leur structure et ils agissent comme des enzymes des 2 côtés de la protéine, à l'intérieur et à l'extérieur.

A l'intérieur de la vésicule, il y a un réseau métabolique, c'est-à-dire un grand nombre de protéines et de métabolites. On essaye donc de synthétiser un produit qui change de couleur ou qui est bioluminescent en réaction à certains biomarqueurs.

L'enzyme transforme les substrats (les métabolites) en produit (nouvelle molécule avec la même composition chimique).

Description du projet :

Dans notre cas, on souhaite, à partir d'un document décrivant les molécules contenues dans la vésicule et leurs réactions, faire fonctionner notre projet afin de simuler l'expérience décrite ci-dessus. On souhaite donc avoir des molécules qui réagissent entre elles pendant une certaine durée et obtenir en résultat les courbes des concentrations de chaque type de molécules présentes à l'origine et des nouvelles apparues. Ce projet sera implémenté de deux façons différentes qui possèdent chacune leurs avantages. Celles-ci seront expliquées dans la prochaine partie.

Les deux approches

Notre projet prend en compte deux approches différentes, qui sont représentées par deux implémentations différentes pour ce même problème. La première approche est appelée l'approche population-centrée, et la seconde l'approche entité-centrée. Des explications pour les deux approches sont faites dans les parties qui suivent.

L'approche population centrée

L'approche population centrée est celle qui sera utilisée pour implémenter le premier simulateur, un simulateur population centrée, c'est-à-dire que pour ce simulateur, on ne prend pas en compte chaque molécule individuellement, mais on détermine la probabilité qu'un certain nombre de réaction, parmi les réactions possibles suivant les molécules présentes dans la vésicule, ait lieu pendant un certain intervalle de temps.

Cet intervalle de temps se nomme "pas de temps", il correspond au temps qui s'écoule entre chaque itération de l'algorithme du simulateur, autrement dit à chaque pas de temps on calcule le nombre de réactions qui peuvent se produire et on l'applique aux concentrations des molécules, i.e on diminue ou on augmente les concentrations des molécules qui sont utilisés dans les réactions. Les réactions peuvent être effectuées plusieurs fois suivant la durée sélectionnée pour le pas de temps, c'est pour cela que le pas de temps ne doit ni être trop long, ni être trop court.

Cette approche part du principe que les molécules sont toutes réparties de manière homogène dans la vésicule, même si cela ne représente pas les conditions réelles dans lesquelles les molécules ne sont pas toutes réparties de manière équitable dans le contenant.

L'approche entité centrée

L'approche entité centrée est, contrairement à l'approche précédente, une représentation plus fidèle de la réalité, on ne suppose plus que les molécules sont réparties de manière homogène dans la vésicule.

Pour cette approche, on en vient à considérer les molécules une à une, en prenant en compte leur vitesse et leur position ainsi que les collisions qu'elles effectuent pendant leur déplacement. Cela nous permet d'obtenir des résultats plus fiables vis-à-vis de la réalité mais cela apporte l'inconvénient de prendre plus de temps que l'approche précédente.

Nous calculons les déplacements en trois dimensions dans la vésicule pour chaque molécule puis les collisions possibles avec d'autres molécules lors de ce déplacement. Plus de détails seront donnés dans la partie implémentation.

Implémentation :

Introduction

La partie implémentation de notre projet est séparée en plusieurs parties distinctes, la première partie étant nécessaire aux deux parties suivantes pour leur fonctionnement. Les trois parties sont la lecture de fichier, le simulateur population-centrée et le simulateur entité-centrée.

Ce projet a besoin d'un fichier d'entrée pour fonctionner, dans lequel l'on trouve différentes informations telles que les concentrations des différents types de molécules ou bien les différentes réactions possibles. En voici un exemple simple :

```
// Exemple de cascade de deux reactions enzymatiques

[E1] = 0.005 uM;
[E2] = 0.008 uM;
[s] = 2 uM;

// les concentrations sont tres faibles pour ne pas avoir trop de
// copies de molecules (moins que 1100) pour que le simulateur
// entite-centre ne soit pas trop lent

duree = 1000;

E1 : s -> inter | 50 uM - 240;
E2 : inter -> p | 0.2 mM - 100;

diametre (s) = 1;      // 1 nano-metre
diametre (inter) = 1;  // 1 nano-metre
diametre (p) = 1;      // 1 nano-metre

vitesse (E1) = 0.1;
vitesse (E2) = 0.1;
```

La première partie de l'implémentation, la lecture de fichier, est la partie qui prend ce fichier en compte et qui en extrait toutes les informations pour les envoyer vers les autres parties. Dans cette lecture de fichier, on trouve deux sous-parties, l'analyseur lexical et l'analyseur syntaxique. Ces deux analyseurs permettent de déterminer si le fichier en entrée correspond bien au format nécessaire au bon fonctionnement du projet et récupèrent les données du fichier pour les stocker de manière interne au projet.

Les deux prochaines parties de l'implémentation correspondent toutes deux à une des approches discutées précédemment, l'une implémente donc un simulateur population-centrée se référant à l'approche du même nom et la seconde implémente l'autre approche, c'est-à-dire l'approche entité-centrée avec un simulateur entité-centrée.

La seconde partie est donc l'implémentation d'un simulateur population-centrée, il suit l'approche homonyme. De même pour la troisième partie qui suit l'approche entité-centrée expliquée plus haut.

Structure de données

Pour nous faciliter la tâche, nous avons eu recours à des structures de données (struct en C++) pour modéliser différents composants importants du projet.

La première structure est l'espèce moléculaire, elle a un nom, un diamètre, une vitesse et un nombre de copies, elle représente les caractéristiques d'une molécule spécifique. Le nombre de copies est utile pour le second simulateur.

La seconde structure représente une réaction, elle a un ou plusieurs substrats (molécules utilisées pour faire la réaction), un ou plusieurs produits (résultats de la réaction) et une probabilité de se produire quand les conditions sont réunies.

La troisième structure représente une molécule, elle a un type et des coordonnées en trois dimensions. Cette structure est utilisée dans l'approche entité-centrée.

La dernière structure que nous avons créée représente un cube, il a un identifiant, des coordonnées en trois dimensions et une liste de molécules. Plus d'explications sur cette structure et son utilisation sont données dans la partie Discussions.

Lecture du fichier

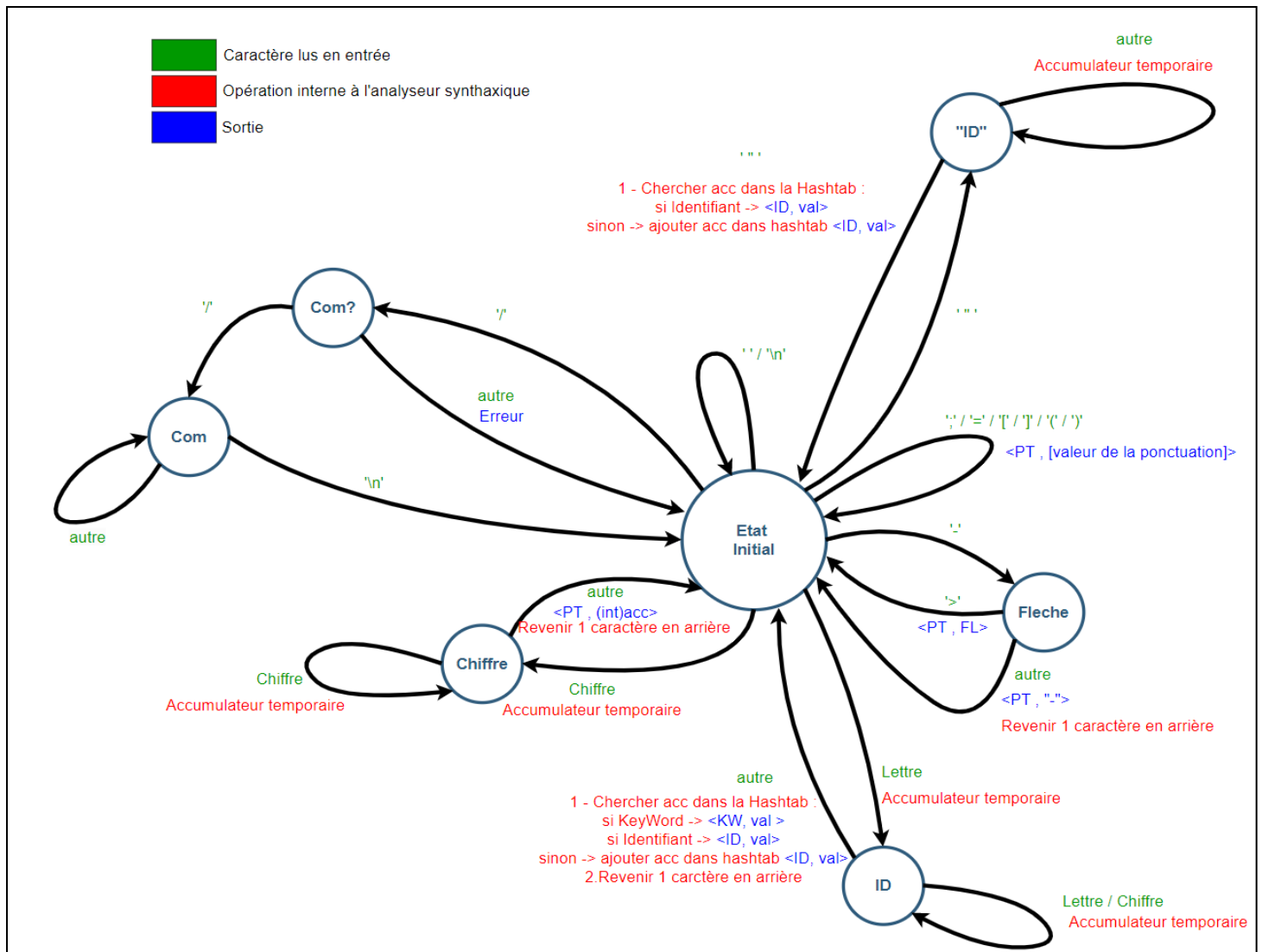
La première étape de notre projet est de lire le fichier qui est donné en entrée du programme et de le décoder afin d'avoir toutes les informations nécessaires à l'exécution de nos algorithmes. Il nous faudra, dans un premier temps, faire l'inventaire de toutes les unités lexicales disponibles puis, définir toutes les expressions acceptables.

Analyseur lexical

On peut séparer les informations données par le document en 4 grandes classes :

- **Les ponctuations** regroupant tous les symboles ne donnant pas une information et qui seront acceptés par notre programme ce qui inclut les parenthèses, les crochets, quelques symboles de ponctuation classiques, ainsi que les symboles mathématiques +, -, et =.
- **Les Nombres**
- **Les Noms** qui indiquent les espèces moléculaire présentes
- **Les Mots clés** qui permettent de définir la caractéristique impliquée

L'analyseur lexical lit donc les caractères en entrée un à un puis sépare les différentes unités lexicales ainsi :

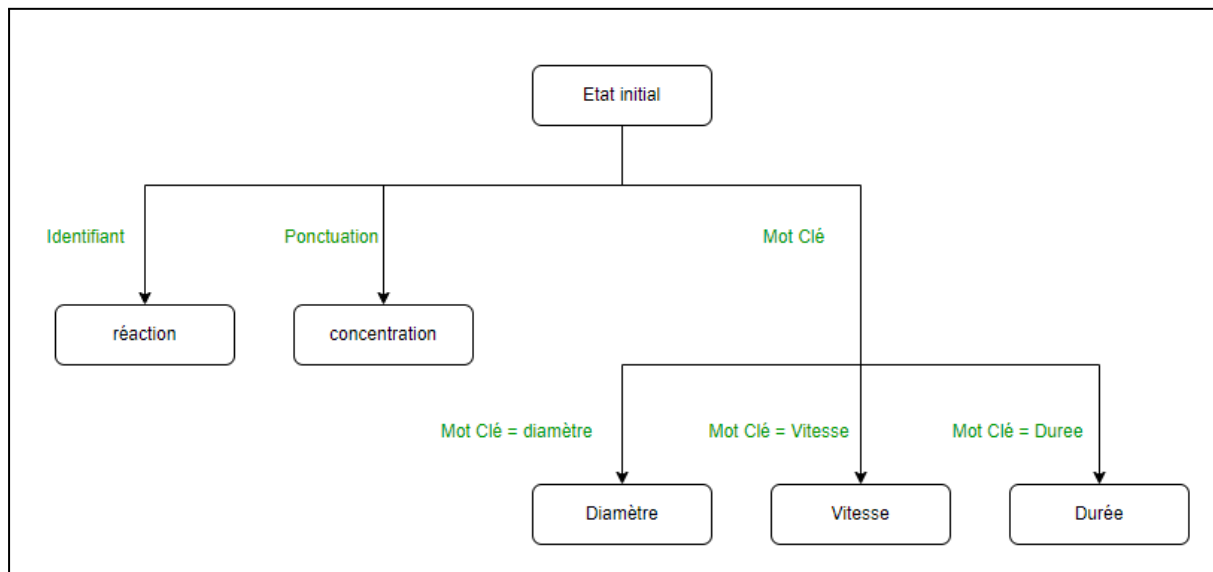


Analyseur syntaxique

L'objectif de l'analyseur syntaxique est de déterminer laquelle des expressions autorisées est utilisée. Les expressions acceptées sont :

- Les durées : "duree = 10;"
- Les diamètres : "diametre (s) = 1;"
- Les vitesses : "vitesse (E2) = 1;"
- Les concentrations : "[E1] = 0.1 uM;"
- Les réactions : "E1 : s -> inter | 50 mM - 240;"

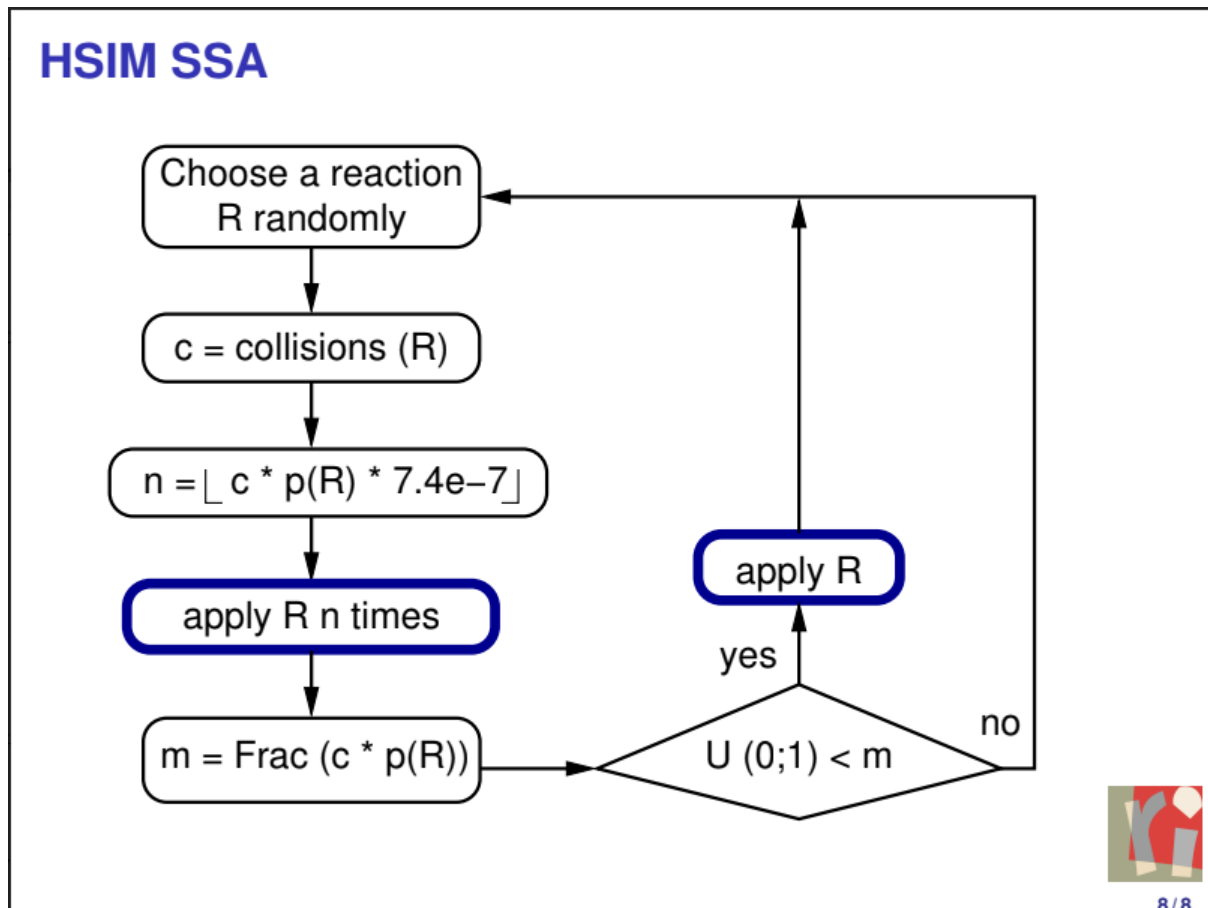
Le premier objectif est donc de savoir laquelle de ces expressions nous sommes en train de lire. Pour cela nous regarderons la première unité lexicale :



Ensuite, le second objectif est de vérifier que l'expression est bien écrite, nous prenons toutes les unités lexicales une à une jusqu'à trouver un “;” et nous vérifions qu'elles soient correctes et dans le bon ordre puis si l'expression est correcte, nous appliquons les caractéristiques de l'expression.

Simulateur population centré :

L'implémentation du simulateur population-centrée se base sur une version simplifiée de l'algorithme de Gillespie, un algorithme de simulation stochastique. Voici un schéma représentant l'algorithme implémenté :



Voyons maintenant le fonctionnement de cet algorithme:

Il commence par sélectionner aléatoirement une réaction R dans la liste des réactions possibles, puis il calcule le nombre de fois où cette réaction peut s'effectuer dans un certain laps de temps, on obtient donc un nombre réel qui correspond au nombre de fois que la réaction doit être appliquée. On applique n fois la réaction, n correspondant à la partie entière du nombre réel calculé précédemment, avec la partie réelle restante m, on tire un nombre aléatoire, si le nombre aléatoire est inférieur à m alors on applique une fois de plus la réaction. On enlève cette réaction de la liste puis on recommence l'algorithme avec une nouvelle réaction jusqu'à ce que la liste soit vide, à ce moment-là, on passe au pas de temps suivant et on recommence avec la liste complète.

Toutes les 100 millisecondes, on récupère les concentrations de chaque type de molécules dans un fichier csv, ce qui nous permettra de pouvoir suivre l'évolution des différentes espèces moléculaires au cours du temps.

Simulateur entité centrée :

⚠ Cette partie du projet est implémentée mais non fonctionnelle, nous avons encore des bugs que nous n'arrivons pas à résoudre.

L'implémentation du simulateur entité-centrée récupère les données extraites du fichier grâce aux différents analyseurs. Nous avons donc accès aux concentrations des différents types de molécules, à leur taille et à leur vitesse. À partir des concentrations et de la formule ci-dessous on peut passer d'une concentration à un nombre de molécules et inversement.

```
// concentration vers le nombre de copies (volume de 1 $\mu$ ³)

#define uM2cn(c) (int)(c*602.252)           // c en  $\mu$ M
#define mM2cn(c) uM2cn(c*1000.)           // c en mM

// nombre de copies vers concentration (volume de 1 $\mu$ ³)

#define cn2uM(n) (((float)n)/602.252)      // résultat en  $\mu$ M
```

Les molécules sont représentées par la structure Molécule présentée précédemment, elles sont stockées dans une liste les regroupant toutes.

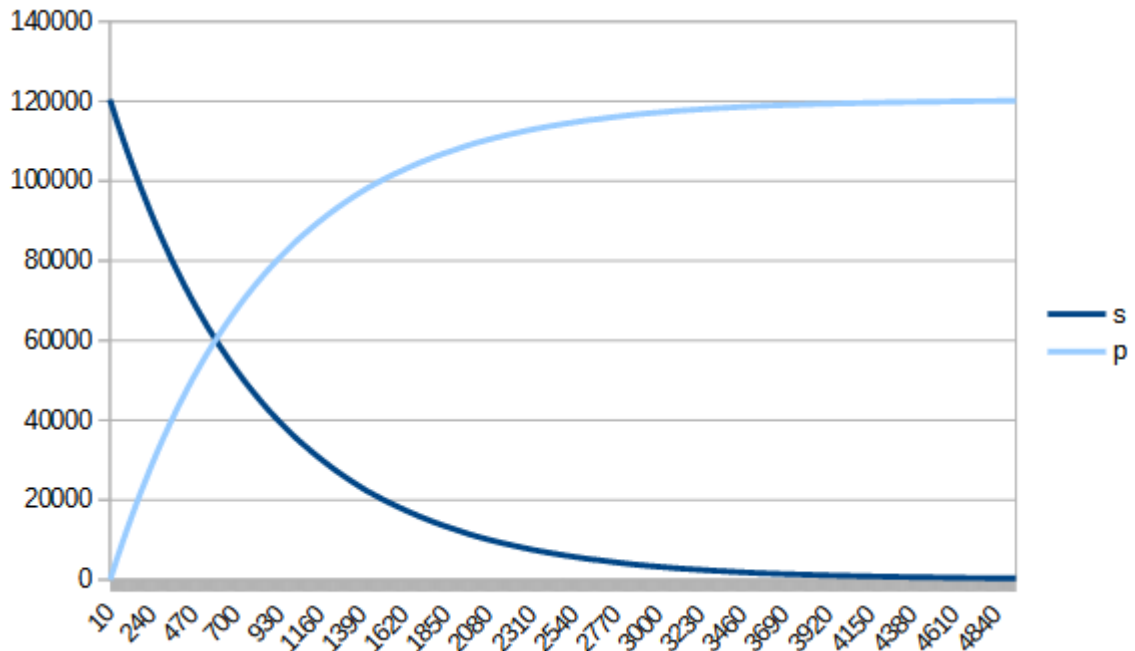
Pour faire fonctionner le simulateur, au premier pas de temps, on parcourt cette liste dans un sens, puis on calcule le déplacement des molécules, pour se déplacer une molécule tire un nombre aléatoire qui sera sa direction puis avance dans cette même direction de la distance qui correspond à la vitesse de ce type de molécule, ainsi que les collisions, si une collision est arrivée alors on calcule si cette collision a donné lieu à une réaction. Si tel est le cas alors on transforme les molécules impactées suivant la réaction et on les modifie dans la liste (deux molécules impliquées peuvent être collées et n'en former plus qu'une ou bien elles peuvent en synthétiser une autre). Toutes les molécules nouvellement créées sont ajoutées à la fin de la liste.

Une fois toute la liste parcourue, et donc après avoir fait se déplacer toutes les molécules, on passe au prochain pas de temps, on répète l'algorithme mais cette fois en partant de l'autre côté de la liste.

On continue ainsi en alternant le côté de la liste duquel on commence à chaque pas de temps jusqu'à ce qu'on atteigne le temps donné dans le fichier en entrée.

Résultats

Pour le simulateur population centrée, on possède un suivi des relevés des concentrations des différents types de molécules, on peut donc tracer des graphiques, montrant l'évolution des concentrations grâce à un logiciel tel que excel par exemple. C'est une façon graphique de constater comment s'est déroulée l'expérience. Voici un graphique montrant les résultats d'une expérience avec ce simulateur :



Pour le second simulateur, ayant accès à la position de toutes les molécules à chaque pas de temps, nous aurions pu faire une représentation en trois dimensions de la vésicule qui s'actualise à chaque pas de temps, avec des points de couleurs différents pour représenter les types de molécules mais nous n'avons pas eu assez de temps pour cela.

Utilisation :

Pour utiliser les simulateurs, il faut taper les commandes suivantes dans la console :

```
g++ Projet.cpp -o exe  
./exe test.txt
```

où test.txt est le fichier contenant les informations de l'expérience qu'il faut donner en entrée.

Afin de choisir le simulateur il faut décommenter les lignes dans le fichier Projet.cpp :

- 826 pour le simulateur entité centrée
- 831 pour le simulateur population centrée

⚠ il reste des bugs qui empêche le simulateur entité-centrée de fonctionner

Discussions :

Tout d'abord concernant les simulateurs, on peut dire que le simulateur est moins précis que celui entité-centrée mais il a l'avantage d'être plus rapide et plus économe en ressource. Pour ce qui est du simulateur entité-centré, bien que moins rapide et plus gourmand, il possède l'avantage d'être beaucoup plus représentatif de la réalité. Ces deux simulateurs sont intéressants dans leur domaine et ne doivent pas être utilisés pour les mêmes raisons, le premier permettant uniquement de rendre des approximations alors que le deuxième essaye de se rapprocher le plus possible d'une situation réelle.

Le second point que nous souhaitons aborder est en rapport avec le second simulateur, le simulateur entité-centré, en effet nous avons des pistes concernant de potentielles améliorations que nous pourrions lui apporter. L'amélioration majeure concerne la gestion des molécules et comment on interagit avec elles. Pour le moment, quand on souhaite accéder à une molécule particulière, on doit parcourir la liste entière jusqu'à trouver la molécule voulue. L'amélioration consisterait à séparer notre vésicule en plusieurs petits carrés et à tenir à jour une liste des molécules présentes dans chaque petit carré, de sorte à ce que lors d'un déplacement d'une molécule, nous n'ayons besoin que de prendre en compte le carré de départ et celui d'arrivée de la molécule, ce qui réduirait grandement le temps nécessaire au calcul des déplacements et des collisions engendrées par cette même molécule.

Un autre point à aborder est la présentation graphique de l'expérience. Comme dit précédemment dans la partie Résultats, nous avons pour objectif de faire une représentation 3D d'une vésicule dans laquelle nous pourrions voir les molécules évoluer, malheureusement nous avons rencontré des difficultés au niveau de l'implémentation de ce simulateur et donc nous n'avons pas eu assez de temps pour tenter d'implémenter cette interface graphique.