TP de Simulation

Simulation stochastique d'une population de lapins

BARBESANGE Benjamin & GARÇON Benoît

novembre 2015

Encadré par David Hill

Table des matières

Présentation du TP

Remerciements

Nous tenons à remercier C. Mazel pour les informations qu'il a pu nous apporter sur les différentes réflexions concernant les lois de probabilités. Nous remercions également M. Hill pour son encadrement lors des séances de TP et l'aide qu'il a pu nous apporter.

Objectif du TP

Ce TP s'inscrit dans le cadre des cours de simulation de $2^{\text{ème}}$ année. Le but est de créer un modèle de simulation stochastique de population de lapins un peu plus réaliste que les précédents modèles effectués avec la fonction de Fibonacci.

Nous allons essayer de produire une simulation dont les performances permettent d'obtenir des simulations sur un, deux, trois, quatre ans puis d'atteindre un nombre de lapins de l'ordre du milliard en un temps raisonnable, voire beaucoup plus.

Ce TP sera aussi l'occasion d'utiliser des outils de développement comme la génération automatique de documentation avec Doxygen, un Makefile propre et aboutit ainsi qu'une mesure des performances à l'aide du profiler gprof.

Analyse des données

Avant de débuter la simulation de la population de lapins, il faut se renseigner afin d'avoir une simulation aussi réaliste et performante que possible.

Reproduction des lapins

Généralement, les femelles matures peuvent faire entre 4 et 8 portées par an avec plus de chance d'en effectuer entre 5 et 7. Chacune de ces portées donne entre 3 et 6 lapereaux.

Afin de compléter l'énoncé, nous avons effectué quelques recherches sur le lapin de garenne sauvage [1] [2]. Celles-ci ont montré que ces lapins avaient une période de reproduction commençant à la fin de l'hiver et s'étalant jusqu'à la fin de l'été, les naissances étant négligeables durant la période automne-hiver (perte de libido « voire des testicules » des mâles). Ceci nous a donc mené à proposer une distribution du nombre de portées de moyenne 6 avec une plus forte probabilité d'avoir 5, 6 ou 7 portées :

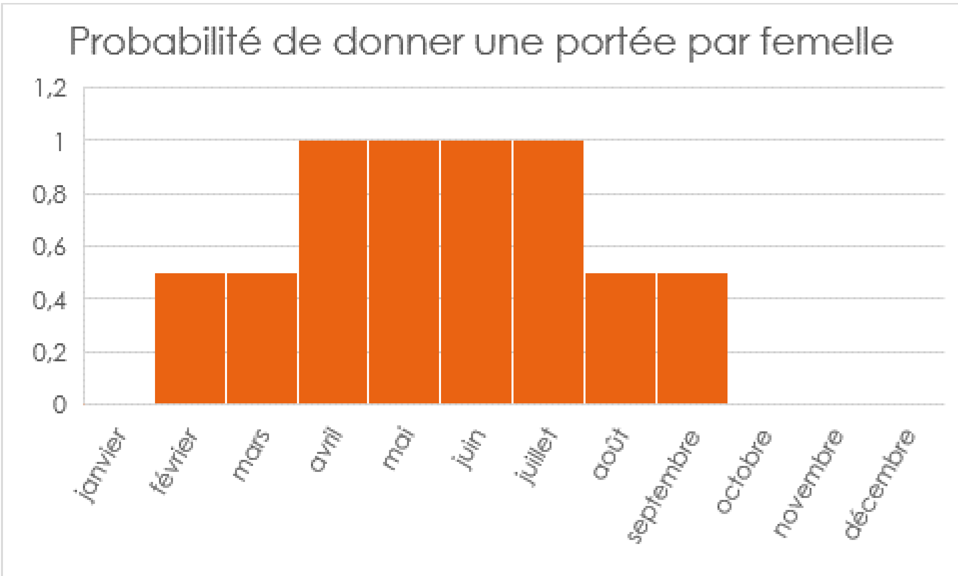


Figure 1 - Probabilité de donner une portée par femelle

Ainsi nous avons ici une probabilité certaine d'avoir une portée chaque mois du cœur de la saison de reproduction, une probabilité nulle en dehors et une probabilité intermédiaire (0,5) sur l'entame et la fin de saison. Ceci nous garantit donc un nombre de portées minimal de 4 et maximal de 8.

En effet, si nous posons Po la variable aléatoire suivant notre loi du nombre de portées par an par femelle, on remarque que Po = 4 + Binomiale(4, ½), ce qui nous donne $P(Po = k) = \frac{1}{16} C_4^k$.

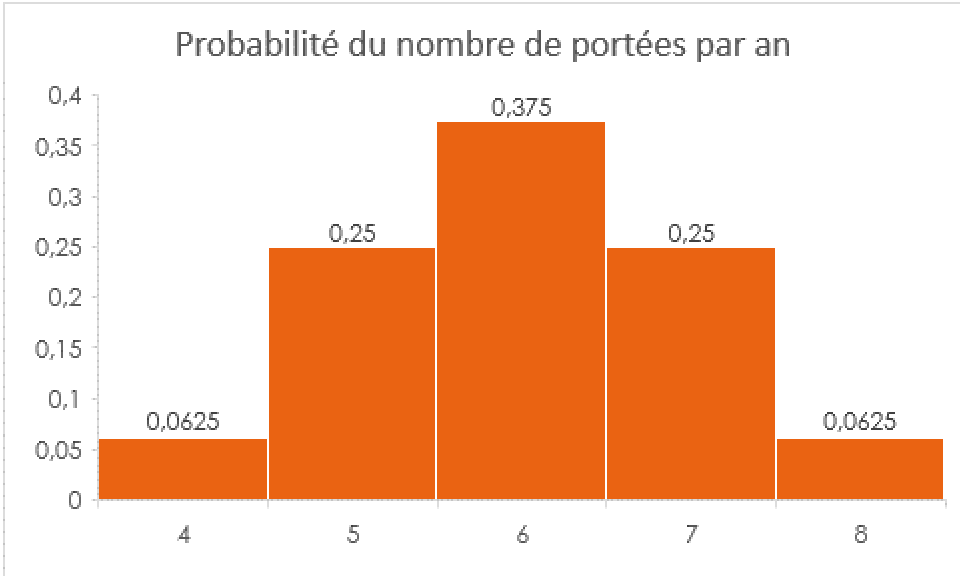


Figure 2 - Probabilité du nombre de portées par an

Comme on peut le voir sur le graphique si dessus l'énoncé est bien respecté avec notre distribution, même si toutefois le choix d'une probabilité certaine peut être contestable sur le cœur de la saison. Mais comme les lapins semblent se reproduire en permanence durant la saison de reproduction ce choix n'est pas non plus aberrant.

Naissance des lapins

La période de gestation de la lapine se trouve entre 28 et 33 jours ; nous simplifierons et prendrons un mois comme période de gestation. En effet cette variation est négligeable à notre échelle mensuelle.

Une portée peut, comme il est mentionné plus haut, être composée de 3 à 6 lapereaux. Il n'y a pas vraiment de causes précises hors mis le poids des lapins, nous avons donc choisi de traiter tous les cas de façon équiprobable.

Pour la répartition mâle/femelle des nouveau-nés, on supposera qu'il y a autant de chances d'avoir un lapereau mâle que de chances d'avoir un lapereau femelle.

Concernant la maturité sexuelle, elle est atteinte entre 5 et 8 mois selon l'énoncé. Selon nos sources [1], celle-ci tend vers une année, de plus il semble que les jeunes lapins attendent pour la plupart la saison des amours suivant celle de leur naissance pour commencer à se reproduire, avant celle-ci ils restent dans le terrier familial. Pour simplifier la simulation, nos lapins sont donc capables de se reproduire dès lors qu'ils entrent dans l'année suivant leur naissance. Cette approximation n'a pas d'importance puisque parmi les plus vieux lapins d'une conscription ne seront pas matures avant la fin de la saison des amours.

Vieillissement et mortalité

L'espérance de vie d'un lapin peut aller au maximum jusqu'à 15 années pour les plus coriaces. C'est pourquoi dans notre simulation nous ne gèrerons les lapins que jusqu'à leurs 15 ans : anniversaire au terme duquel la mort frappera obligatoirement.

De plus, il est notable que les lapins n'ont pas le même taux de survie lorsqu'ils sont jeunes et lorsqu'ils sont plus âgés. Nous avons effectué la répartition suivante pour les taux de survie :

Age en année	Taux de survie en
0 à 9 mois	20
10 à 12 mois	50
2	50
3	50
4	50
5	50
6	50
7	50
8	50
9	50
10	50
11	40
12	30
13	20
14	10
15	0

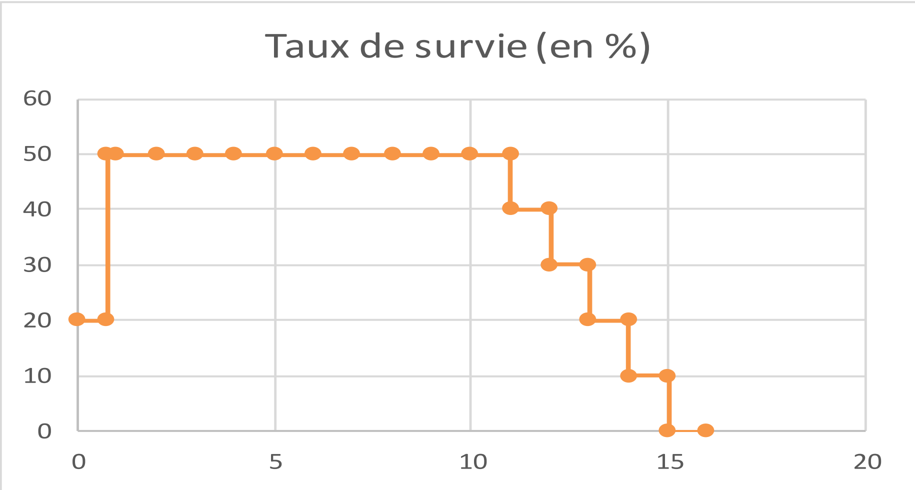


Figure 3 - Taux de survie des lapins en fonction de leur âge

Solution retenue

Organisation du code

Notre programme de simulation s'organise autour de 5 fichiers :

- main.cpp : code principal de la simulation
- ClasseLapins.cpp : représentation de lapins en classe d'âge
- LapinManager.cpp : code de gestion de toutes les classes de lapins et de la simulation
- Student.cpp : classe utile au calculs statistiques

Les fichiers Student.cpp/, ClasseLapins.cpp et LapinManager.cpp sont accompagnés de leur header C++.

Il existe deux versions du code toutes deux prêtes à compiler :

- Une première version compilable grâce à la commande **make GameOfRabbitsl** qui utilise des unsigned long long pour compter les lapins et exécutable grâce à **./GameOfRabbitsl** ;
- Une seconde version compilable grâce à la commande **make GameOfRabbits** ou simplement **make** qui utilise des long double pour compter les lapins et exécutable grâce à **./GameOfRabbits**.

Ceci est dû au fait que la version long long ne nous permet pas de poursuivre de longues simulations, alors que long double le permet au détriment de la précision. Dans la suite de ce rapport nous utiliseront la version unsigned long long pour les simulations de précision (fichiers finissant par l).

Vous pourrez trouver la documentation complète des fichiers en ouvrant le fichier *index.html*. Les fichiers du programme seront également trouvables et sont accompagnés d'un Makefile permettant de recompiler le programme sur votre machine.

Simulation

Dans cette partie, nous détaillerons un peu plus les procédés utilisés pour effectuer la simulation.

Contextualisation de notre simulation

Nous avons décidé de fixer le début de la simulation à un seul couple de lapins d'un an, ceci nous permet de coller à peu près à ce qui s'est produit en Australie et d'ainsi pouvoir mettre en parallèle un phénomène réel avec notre simulation.

Afin d'obtenir de bonnes performances, nous avons utilisées certaines approximations de lois de probabilités qui sont adaptées aux grandes populations, s'éloignant ainsi d'une étude individu-centré. Toutefois notre population initiale n'est pas une "grande population". C'est pourquoi nous avons fait le choix d'une simulation à deux vitesses :

- Tant que nos populations de lapins sont inférieures à 100 individus nous utilisons des lois dites individu-centrées ;
- Au-delà nous utilisons des lois approximant les lois individuelles (théorème central limite …) selon les conseils de C. Mazel.

Sur une étude statistique, on entend souvent par "grande population" une population d'effectif supérieur à 30. Nous sommes donc avec notre seuil de 100 dans un bon cadre d'utilisation.

Vieillissement de population

En ce qui concerne le vieillissement de la population, si la population de la classe n'est pas trop importante (inférieure à 100), nous faisons un tirage selon une loi de Bernoulli ayant pour paramètre le taux de survie pour la classe. Nous le faisons pour chaque individu de la classe. Si ce tirage réussit, l'individu reste en vie, sinon il meurt. Nous n'utilisons pas ici de loi Binomiale pour bien insister sur l'aspect unitaire des tirages.

Dans le cas où nous avons beaucoup d'individus dans une classe (plus de 100), nous utilisons une loi normale d'espérance :

$$e = \ taux\ de\ survie\ de\ la\ classe * nombre\ d'individus\ de\ la\ classe$$

Et d'écart type :

$$\sigma = \sqrt{taux\ de\ survie * nombre\ d'individus * (1 - taux\ de\ survie)}$$

Grace au cours de probabilité de première année, nous avons pu constater qu'une loi Binomiale avec une grande population convergeait vers une loi Normale. C'est pourquoi ici, pour une grande population, nous avons décidé de choisir une loi Normale de paramètres (n*p, n*p*q).

Naissance de lapereaux

La naissance des lapereaux se répartie sur les 8 mois de février à septembre. La probabilité de reproduction des femelles pour les deux premiers mois ainsi que les deux derniers mois de cette période est de 0.5. Sur les mois restant, la probabilité de reproduction est de 1 car le minimum de portées par ans est de 4.

Nous pouvons interpréter les mois sans reproduction comme étant logique puisque nous entrons dans une période hivernale, ce qui implique que nous aurons seulement des décès et aucune naissance.

Il faut noter que nous supposons qu'il y a toujours assez de mâles pour féconder les femelles qui en ont besoin. Toutefois pour éviter que des populations constituées uniquement de femelles ne se reproduisent nous vérifions qu'il existe au moins un mâle. Cette vérification est très importante lors des premiers mois de simulation.

Lorsque la population de la classe est inférieure à 100, nous faisons une épreuve de Bernoulli pour chaque femelle des classes matures avec pour paramètre la probabilité d'avoir une portée. Ensuite si nous avons obtenu un succès, nous effectuons un tirage d'une loi uniforme discret entre 3 et 6. Cette loi uniforme U est décrite ainsi :

$$E[U_{[\![3;6]\!]}] = \frac{3+4+5+6}{4} = \frac{9}{2} \quad et \quad V[U_{[\![3;6]\!]}] = E[{U_{[\![3;6]\!]}}^2] - E[U_{[\![3;6]\!]}]^2 = \frac{5}{4}$$

Nous avons aussi une loi Normale pour une classe dont l'effectif est supérieur à 100. L'espérance est :

$$e = \frac{9}{2} * probabilité\ de\ portée * nombre\ de\ femelles$$

Et l'écart type utilisé est :

$$\sqrt{\frac{5}{4} * probabilité\ de\ portée * nombre\ de\ femelles}$$

Nous approximons donc notre somme de loi uniforme, dont nous avons calculer les paramètres précédemment, par une loi Normale d'espérance N*E[U] et de variance N*V[U]. A ceci nous multiplions la probabilité d'avoir une portée.

La détermination du sexe s'effectue en utilisant les mêmes lois dans le mêmes conditions (taille de la population de la classe) que dans le cas des naissances. Les paramètres de la loi Normale dans ce cas est, pour l'espérance :

$$e = 0.5$$

L'écart type :

$$n * p^2 = \sqrt{0.25 * nombre\ de\ naissances}$$

Le paramètre de la loi de Bernoulli dans ce cas est 0.5 car nous estimons qu'il y a autant de chances d'obtenir un lapin mâle qu'un lapin femelle.

Utilisation du programme

Pour lancer la simulation avec des paramètres personnalisés pour le nombre d'années à simuler et le nombre de réplications, il faut lancer le programme dans le terminal et spécifier dans cet ordre : le nombre d'années à simuler, puis le nombre de réplications à effectuer. Dans le cas où rien n'est spécifie, nous simulons simplement 10 années en une seule fois.

Nous pouvons avoir également différents modes d'affichage. Pour lancer la simulation et obtenir les affichages à l'écran, il faut lancer le programme dans la console et ajouter l'option '-v'. On peut également utiliser l'option '-f' afin d'avoir les résultats dans le fichier *lap.out* qui sera créé dans le répertoire du projet. Ces deux options sont cumulables.

Structure du fichier de sortie

En utilisant l'option –f du programme, un fichier de sortie *lap.out* est généré. Ce fichier contient les résultats de toutes les simulations lancées. Une ligne se compose des données suivantes :

[9/2007]	**43526**	**4553**	**5277**	**21888**	**21638**

Les champs contiennent :

- La date à laquelle on a recueillies les données. On va trouver le mois ainsi que l'année. Notons que l'année de début de simulation est 2000.
- La population actuelle à la date indiquée.
- Le nombre de naissances
- Le nombre de morts
- Le nombre de mâles
- Le nombre de femelles

Chaque donnée est séparée par une tabulation. Ainsi si on ouvre directement ce fichier avec un tableur, les données seront disposées correctement et on peut tracer des graphes.

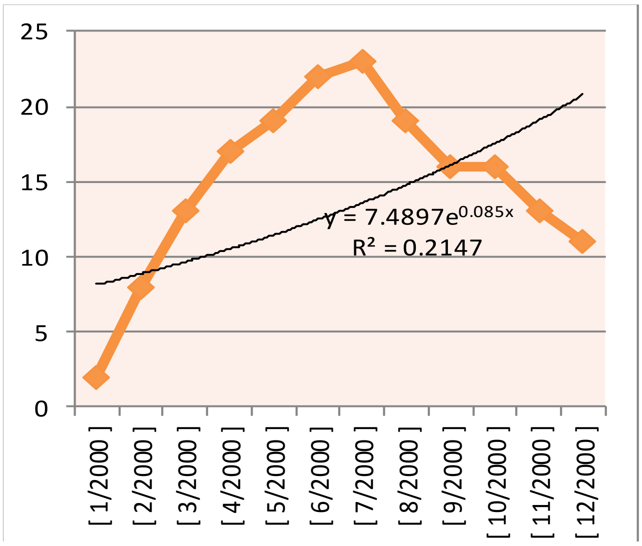
Analyse des résultats

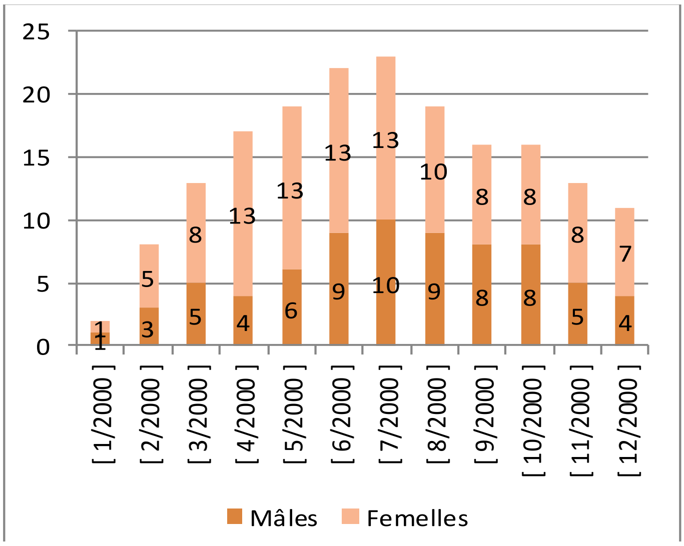
Le fait d'avoir implémenté la sauvegarde des données dans un fichier permet trace d'exécution du programme mois par mois. Ceci nous permet d'avoir des informations, comme le nombre de mâles, de femelles, la population courante ou encore le nombre de naissances. Il est possible.

Lors de ces simulations, étant donné que nous débutons avec un mâle et une femelle d'un an, il arrive que ceux deux lapins meurent rapidement sans avoir eu le temps de se reproduire. La population de lapins reste donc à 0.

Toutes nos simulations sont répétés 5000 fois afin d'affiner au mieux les résultats.

Simulation sur 1 an





Nous observons que pour une simulation d'un an, la population ne semble pas suivre une loi exponentielle. On peut remarquer cependant que la proportion mâles/femelles reste constante au fil de la simulation.

Simulation sur 20 ans

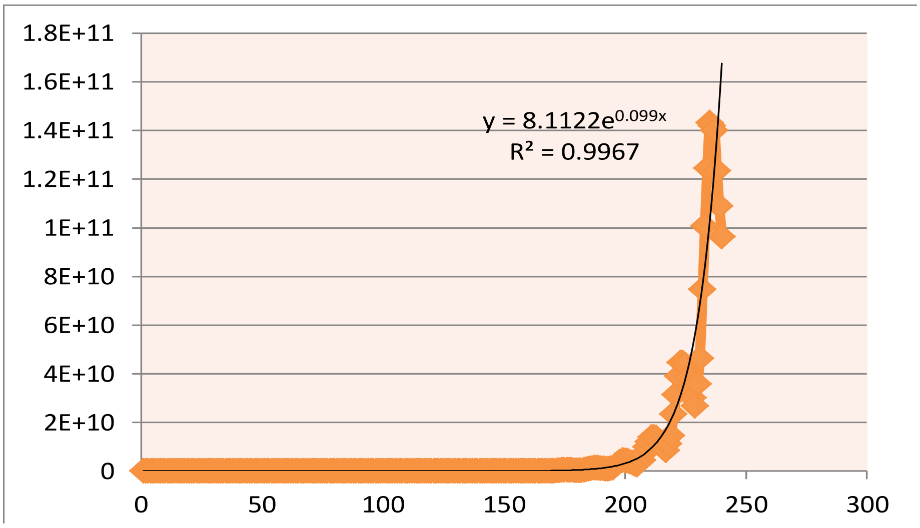


Figure 5 - Répartition de la population sur 20 ans

Cette fois ci, nous observons que la population croît vraiment selon une loi exponentielle. Ceci est cohérent, car plus il y a de lapins, plus on obtiendra de lapereaux et plus la population augmentera.

Comparaison du nombre de réplications

Simulation sur 20 ans et 50 réplications

Nous procédons à une simulation sur 20 ans, avec 50 réplications. La commande exécutée est la suivante :

```
./GameOfRabbitsI 20 50
```

Le programme nous retourne la population finale de chaque simulation. Ainsi, en entrant ces valeurs dans un tableur, nous pouvons calculer la moyenne et l'intervalle de confiance relatifs à ces données.

Données
3.5289E+10
4.3713E+10
3.7295E+10
1.0590E+11
1.3844E+11
1.9020E+10
2.7945E+10
8.2069E+10
1.5534E+11
4.4852E+10
7.6744E+09
0.0000E+00
0.0000E+00

Moyenne
6.2659E+10

Intervalle confiance
min
4.7237E+10
max
7.8081E+10

Taille echantillon
50

Figure 6 - Intervalle de confiance sur 50 réplications

En lançant le programme avec la commande *time*, on peut obtenir le temps d'exécution du programme qui est ici de 1.640s.

Nous pouvons remarquer que l'intervalle de confiance est plutôt large. Nous allons donc effectuer plus de réplications.

Simulation sur 20 ans et 5000 réplications

La commande exécutée est la suivante :

```
/GameOfRabbitsI 20 5000
```

Afin de teste les performances du programme, nous avons lancé une simulation de 20 ans avec 5000 réplications. Nous avons pu obtenir ces résultats en 2 minutes et 5 secondes. Nous pouvons donc établir une meilleure approximation des résultats.

Données
3.5289E+10
4.3713E+10
3.7295E+10
1.0590E+11
1.3844E+11
1.9020E+10
2.7945E+10
8.2069E+10
1.5534E+11
4.4852E+10
7.6744E+09
0.0000E+00
0.0000E+00

Moyenne
6.2910E+10

Intervalle confiance
min
6.1496E+10
max
6.4324E+10

Taille echantillon
5 000

Figure 7 - Intervalle de confiance avec 5 000 réplications

Avec plus de réplications, nous observons que l'intervalle de confiance est largement réduit.

Analyse des performances

Pour analyser les performances, nous avons exécutées la simulation sur un nombre d'années de plus en plus grand. Il en résulte le graphique suivant :

Années	Temps (s
10	0.000
20	0.016
30	0.012
40	0.024
50	0.056
60	0.060
70	0.084
80	0.108
90	0.116
100	0.152
200	0.192
300	0.304
400	0.332
500	0.460
600	0.596

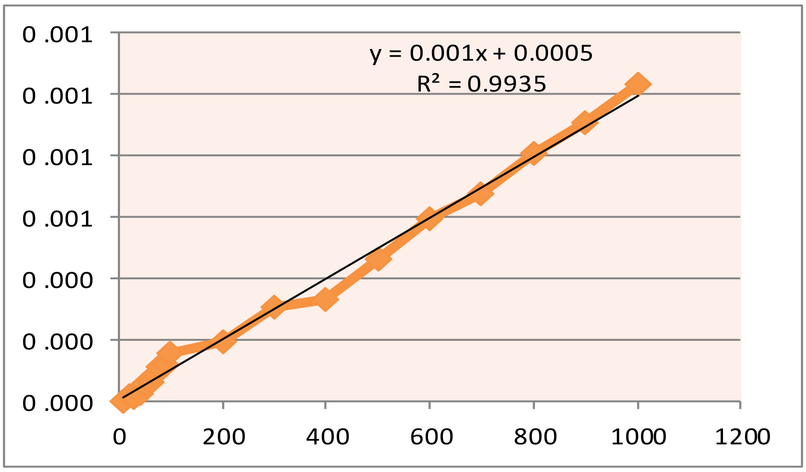


Figure 8 - Temps d'exécution en fonction du nombre d'années

D'après ce graphique, on peut voir que le temps de calcul dans les premières années augmente de plus en plus et n'est pas linéaire. Ceci est dû au fait que la population n'est pas très élevée au début et que l'on simule individuellement chaque lapin.

Une fois que la population est assez importante et que l'on simule avec la loi normale, on observe que le temps d'exécution devient linéaire.

Ce simulations sont effectuées sur nos machines personnelles, on peut donc espérer avoir des temps de calcul largement réduits sur les serveurs de l'ISIMA.

Conclusion

Grâce à ce TP, nous avons pu avoir une meilleure approche de la simulation. Nous avons bien compris que l'analyse est la partie dominante qui va guider nos choix par rapport à l'orientation de la solution.

Dans notre cas, nous voulions pouvoir simuler une très longue période, c'est pourquoi nous avons simplifié certaines données comme le temps de gestation des femelles ou encore le temps de maturité des lapereaux.

Il en résulte donc ici un programme dont les performances sont bonnes, comme ont pu le montrer les résultats. Après avoir effectuées des comparaisons avec d'autres groupes ayant des simulations individu-centrées, il est apparu que nos données sont proches des leurs.

De plus, ce TP a permis de mieux nous familiariser avec les fonctions de base en C++ permettant d'utiliser des lois ou encore le générateur de nombres aléatoires Mersenne Twister.

Références

Aucune source spécifiée dans le document actif.

[1] http://www.margueritecie.com/garenne.php Dernière mise à jour : 18/11/2015

[2] https://fr.wikipedia.org/wiki/Oryctolagus_cuniculus