



INSTITUT SUPÉRIEUR
D'INFORMATIQUE, DE MODÉLISATION
ET DE LEURS APPLICATIONS

TP de Simulation

Simulation de Monte Carlo et Calcul d'Intervalles de Confiance

Marion CARRIER et Aurélie DHENRY



année scolaire 2010/2011
Encadrant : David HILL

Table des matières

Table des figures

Liste des tableaux

Introduction	1
1 Calcul de π par la méthode de Monte Carlo	2
1.1 Méthode utilisée	2
1.2 Tests de l'importance des différents paramètres	3
1.2.1 Importance du nombre de tirages effectués	3
1.2.2 Importance du nombre de réplifications effectuées	4
2 Simulation de croissance de population	6
2.1 Théorie d'évolution de la population	6
2.2 Résultats	7
3 Simulation de reproduction de lapins	9
3.1 Théorie d'évolution de la population	9
3.2 Description des paramètres grâce à des lois de probabilités	9
3.3 Résultats	10
3.3.1 Performances	10
3.3.2 Evolution sur 3, 4 et 5 ans	11
3.3.3 Coefficient d'accroissement de la population	12
3.3.4 Taille de la population après 5 ans	13
Conclusion	15
Références	
Ouvrages consultés	
Annexes	

Table des figures

1.1	Estimation de la valeur de π par une simulation de Monte Carlo.	2
2.1	Reproduction théorique des lapins suivant une simulation simple.	6
3.1	Répartition théorique de la reproduction des lapins.	10
3.2	Reproduction théorique de l'âge de décès des lapins.	10
3.3	Temps d'exécution en fonction du nombre de lapins générés sur le serveur Omer de l'Isima.	11
3.4	Evolution de la taille de la population en fonction des mois, simulée sur différentes durées.	11
3.5	Ratio d'accroissement de la population en fonction du mois.	12
3.6	Probabilités des tailles de population que l'on peut obtenir en 5 ans.	13



Liste des tableaux

1.1	Variation de la valeur de π obtenue en fonction du nombre de tirages. . . .	3
1.2	Variation de la précision des résultats en fonction du nombre de réplifications.	4
2.1	Valeurs de la suite de Fibonacci.	7
3.1	Ratio d'accroissement de la population par année.	13



Introduction

Lors du précédent TP, nous avons pu prendre conscience qu'une seule simulation ou qu'un nombre relativement faible d'expériences ne suffisait pas à donner des résultats certains. Nous avons pu approcher concrètement la notion de moyenne, qui est essentielle pour l'expression de résultats cohérents.

Ce TP nous permet d'appliquer à des cas concrets de simulation, l'évolution d'une population de lapins, les lois implémentées précédemment. Il a également pour but de nous faire comprendre que la moyenne seule n'apporte pas assez de précision quant à l'efficacité d'une simulation, et qu'il faut en plus tenir compte des notions d'écart-type, et d'intervalles de confiance.



1 Calcul de π par la méthode de Monte Carlo

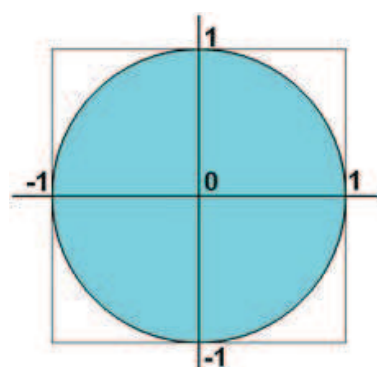
1.1 Méthode utilisée

On appelle méthode de Monte Carlo toute méthode de simulation discrète utilisant des procédés stochastiques, c'est-à-dire qui utilise un générateur de nombres aléatoires. Ici nous allons utiliser l'une de ces méthodes pour calculer la valeur de π . L'expérience est similaire à celle de l'aiguille de Buffon (voir [3]).

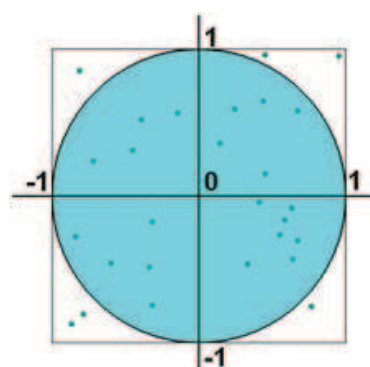
La méthode utilisée est simple : on définit un carré de côté 2, soit d'aire 4, et on définit également le cercle inscrit dans ce carré, de diamètre 2, comme le montre la figure 1.1(a). Le carré et le cercle sont centrés en 0 sur un repère orthonormal, pour des raisons algorithmiques. Le calcul de π correspondra, dans ce cas, exactement à l'aire du cercle. Pour le déterminer, la technique à utiliser est la suivante : nous allons tirer aléatoirement des nombres appartenant au carré, et nous allons regarder s'ils appartiennent aussi au cercle (figure 1.1(b)). Ainsi, le rapport :

$$\frac{\text{Nb tirages dans le cercle}}{\text{Nb tirages dans le carré}}$$

donne une valeur approchée de π , la précision de l'estimation dépendant bien évidemment du nombre de tirages effectués.



(a) Univers d'intérêt.



(b) Tirages aléatoires dans l'univers des possibles.

FIGURE 1.1 – Estimation de la valeur de π par une simulation de Monte Carlo. *Source personnelle.*

Pour cette simulation, nous avons appliqué l'algorithme donné dans [2] :

```

Initialiser le générateur de nombre pseudo-aléatoires
n = 0, nb_expériences = 0

Pour i = 1.. nb_expériences, faire
    Tirer un point dans le carré  $[-1, 1] * [-1, 1]$ , soit :
    x = 2 * (Tirage d'un nombre aléatoire dans  $[0, 1]$ ) - 1
    y = 2 * (Tirage d'un nombre aléatoire dans  $[0, 1]$ ) - 1
    Si le point tiré est sur le disque (ie racine carrée de  $x^2 + y^2 \leq 1$ )
        Alors incrémenter n
    Fin si
Fin pour

Retourner  $4 * n / \text{nb\_expériences}$ 

```

1.2 Tests de l'importance des différents paramètres

Pour la simulation, nous avons laissé le choix à l'utilisateur du nombre de réplifications de ce calcul à faire, et du nombre de tirages à effectuer. De cette façon, nous pouvons obtenir une moyenne plus ou moins fiable. Mais pour obtenir des résultats intéressants, nous avons rajouté des fonctions nous permettant de connaître l'écart-type correspondant à la moyenne calculée, et donc des intervalles de confiance. Ici nous n'avons considéré que des intervalles de confiance à 95%.

1.2.1 Importance du nombre de tirages effectués

Lorsque l'on fait varier uniquement le nombre de tirages à effectuer (et donc que le nombre de réplifications et de répétitions de l'expérience restent fixes), nous avons obtenu les résultats donnés dans le tableau 1.1.

Nombre de tirages	Nombre de réplifications	Valeur moyenne de π (sur 50 expériences)	Ecart par rapport à $\pi = 3.14159$	Ecart-type moyen (sur 50 expériences)	Rayon d'intervalle moyen (sur 50 expériences)	Intervalle à 95% moyen (déduit du rayon moyen)
100	10	3.14096	0.00063	0.16042	0.10374	[3.03722, 3.24470]
1 000	10	3.13792	0.00367	0.0485	0.03067	[3.10725, 3.16860]
10 000	10	3.13675	0.00484	0.01577	0.01000	[3.12677, 3.14672]
100 000	10	3.13798	0.00361	0.00491	0.00311	[3.13487, 3.14110]

Tab. 1.1 – Variation de la valeur de π obtenue en fonction du nombre de tirages. *Source personnelle.*

Comme on peut le voir, la moyenne à elle seule n'est pas suffisamment pertinente : nous pourrions déduire de ces résultats que nous obtenons des meilleurs résultats avec peu de tirages. Cependant, l'écart-type est primordial : si la moyenne est meilleure avec 100 tirages qu'avec 100 000, l'écart-type quant à lui nous montre bien que l'amplitude des calculs est beaucoup trop importante pour 100 tirages : nous pouvons tirer des valeurs entre 2.98054 et 3.30138, alors qu'avec 100 000 tirages nous pouvons tirer des valeurs entre 3.13307 et 3.14289.

1.2.2 Importance du nombre de réplifications effectuées

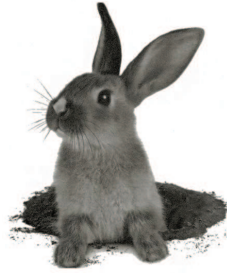
Nous nous sommes également intéressées à l'influence du nombre de réplifications sur le calcul de π , et les résultats obtenus sont ceux du tableau 1.2.

Nombre de réplifications	Nombre de tirages	Valeur moyenne de π (sur 50 expériences)	Ecart par rapport à $\pi = 3.14159$	Ecart-type moyen (sur 50 expériences)	Rayon d'intervalle moyen (sur 50 expériences)	Intervalle à 95% moyen (déduit du rayon moyen)
10	10 000	3.13675	0.00484	0.01577	0.01000	[3.12677, 3.14672]
15	10 000	3.13795	0.00364	0.01622	0.00838	[3.12957, 3.14632]
20	10 000	3.13627	0.00532	0.01594	0.00713	[3.12914, 3.14340]
25	10 000	3.13809	0.00350	0.01629	0.00652	[3.13158, 3.14461]
30	10 000	3.13649	0.00510	0.01631	0.00596	[3.13054, 3.14245]
40	10 000	3.13803	0.00356	0.01665	0.00526	[3.13280, 3.14329]
80	10 000	3.13834	0.00325	0.01643	0.00367	[3.13467, 3.14202]
120	10 000	3.13754	0.00405	0.01653	0.00151	[3.13603, 3.13905]

Tab. 1.2 – Variation de la précision des résultats en fonction du nombre de réplifications. *Source personnelle.*

Comme on peut le voir, les valeurs du nombre de réplifications de calculs n'influent pas sur la moyenne de π , mais plutôt sur le rayon moyen d'intervalle : en effet, plus le nombre de réplifications augmentent, et plus le rayon a tendance à être petit, ce qui réduit considérablement la taille de l'intervalle de confiance. Cela peut être une bonne chose, dans le sens où 95% des cas appartiennent à un petit intervalle.

Cependant, comme nous pouvons le voir sur les 50 simulations à 120 répliques, la valeur théorique de π peut ne pas appartenir à l'intervalle de confiance, ce qui pose problème. Il vaut donc mieux avoir un nombre moyen de répliques, quitte à ne pas avoir de résultats extrêmement précis, mais qui au moins sont cohérents avec la théorie.



2 Simulation de croissance de population

2.1 Théorie d'évolution de la population

Pour cet exercice, nous allons simuler une reproduction de lapins. Cette simulation sera basée sur des critères extrêmement simples, et absolument irréalistes.

En effet, nous allons considérer les lapins comme des couples, qui peuvent être soit jeunes, soit adultes. Un couple de jeunes lapins met une saison à devenir adulte, et peuvent alors commencer à copuler. La gestation dure une saison après quoi ils donnent naissance à un nouveau couple de jeunes lapins. De plus, on considérera que les lapins ne meurent jamais, et se reproduisent à l'infini. La simulation doit alors suivre la figure 2.1.

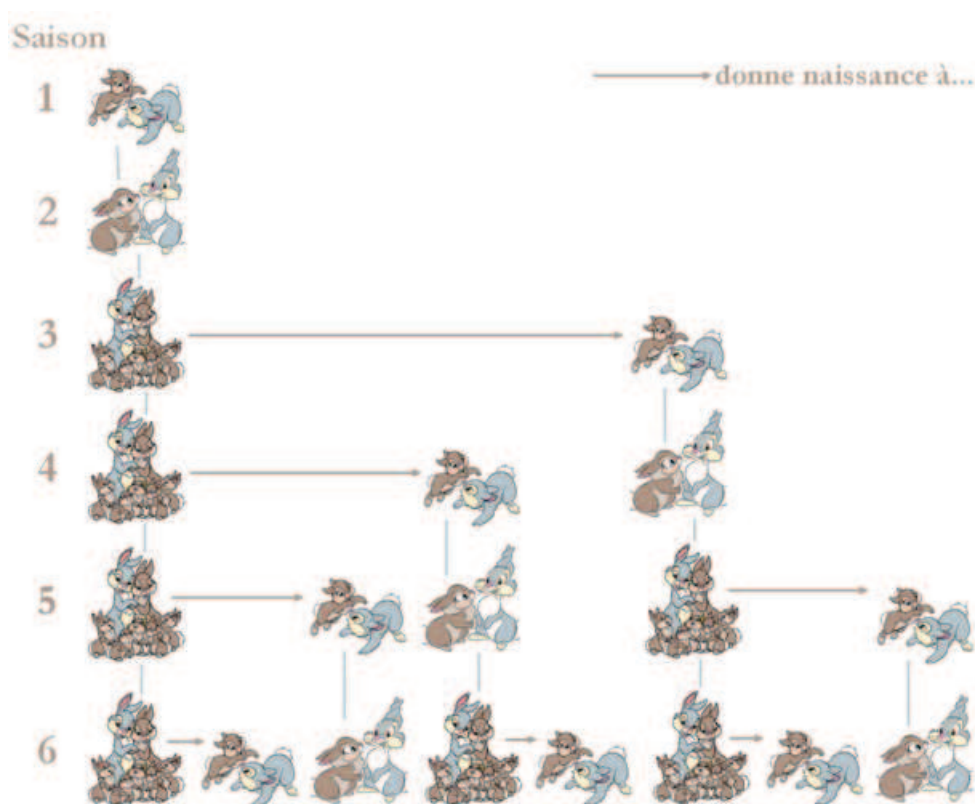


FIGURE 2.1 – Reproduction théorique des lapins suivant une simulation simple. *Schéma créé à partir des images disponibles sur [1].*

Comme on peut s'en douter, il n'y a aucune place pour l'aléatoire dans cette simulation. Le nombre de lapins existant par saison doit suivre les valeurs de la suite de Fibonacci, rapellée dans le tableau 2.1.

Itération (ou saison)	Nombre de lapins
1	1
2	1
3	2
4	3
5	5
6	8
7	13
8	21
9	34
10	55
20	6 765
30	832 040
40	102 334 155

Tab. 2.1 – Valeurs de la suite de Fibonacci. *Source personnelle.*

2.2 Résultats

Nous avons exécuté notre programme, en utilisant des unsigned long long int pour pouvoir simuler le plus grand nombre de saisons, et donc de reproduction de lapins possibles (voir code en annexe [2]). De cette façon, nous avons ainsi pu obtenir des résultats pour les 93 premières saisons :

```
Saisir le nombre de saisons : 93
Nombre de couples de lapinous a la saison 1 : 1
Nombre de couples de lapinous a la saison 2 : 1
Nombre de couples de lapinous a la saison 3 : 2
Nombre de couples de lapinous a la saison 4 : 3
Nombre de couples de lapinous a la saison 5 : 5
Nombre de couples de lapinous a la saison 6 : 8
Nombre de couples de lapinous a la saison 7 : 13
Nombre de couples de lapinous a la saison 8 : 21
Nombre de couples de lapinous a la saison 9 : 34
Nombre de couples de lapinous a la saison 10 : 55
...
Nombre de couples de lapinous a la saison 20 : 6765
Nombre de couples de lapinous a la saison 21 : 10946
...
Nombre de couples de lapinous a la saison 40 : 102334155
Nombre de couples de lapinous a la saison 41 : 165580141
...
Nombre de couples de lapinous a la saison 60 : 1548008755920
Nombre de couples de lapinous a la saison 61 : 2504730781961
...
```

Nombre de couples de lapinoux a la saison 80 : 23416728348467685
Nombre de couples de lapinoux a la saison 81 : 37889062373143906
...
Nombre de couples de lapinoux a la saison 91 : 4660046610375530309
Nombre de couples de lapinoux a la saison 92 : 7540113804746346429
Nombre de couples de lapinoux a la saison 93 : 12200160415121876738

Nous pouvons voir que cette simulation n'a rien de réaliste et est une introduction à une simulation plus réaliste que nous allons développer en troisième partie.



3 Simulation de reproduction de lapins

3.1 Théorie d'évolution de la population

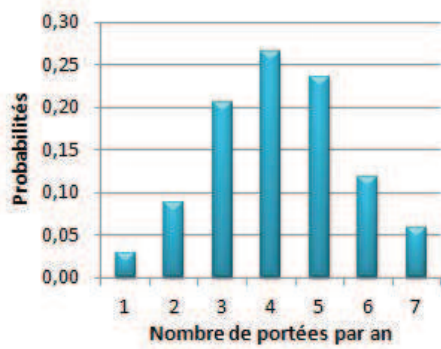
Dans cet exercice, nous allons également considérer une population de lapins, mais de façon plus réaliste.

Tout d'abord, nous n'allons plus considérer les lapins comme des couples, puisque cette espèce n'est pas monogame. En effet, une femelle accepte des mâles pendant la moitié de sa période de gestation et chaque mâle peut féconder plusieurs femelles le même jour, l'accouplement étant extrêmement rapide (entre 10 et 15 secondes). Nous considérons donc qu'un mâle peut féconder 3 femelles par jour, soit 90 par mois. De plus, si nous étudions l'évolution de la population par mois, nous pouvons estimer que la durée de gestation des femelles est d'un mois. Il faut également tenir compte de la maturité des lapins : les lapines deviennent adultes (et donc aptes à se reproduire) à l'âge de quatre mois, et les lapins mâles à l'âge de trois mois.

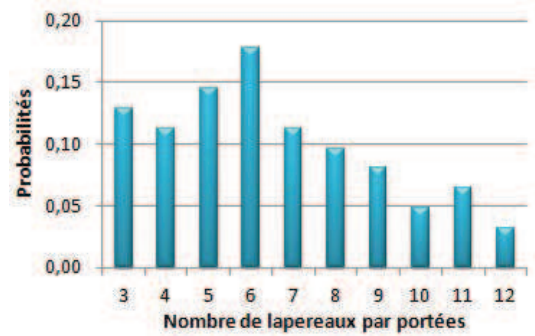
3.2 Description des paramètres grâce à des lois de probabilités

Pour gérer le nombre de portées qu'une femelle peut avoir dans l'année (entre un et sept) nous avons recours à des tirages probabilistes, se basant sur les méthodes implémentées et testées pour le TP précédent (cf. histogramme 3.1(a)). Ce nombre de portées est tiré tous les douze mois de la lapine pour l'année à venir. De même, nous avons estimé le nombre de lapereaux par portées entre trois et douze suivant l'histogramme 3.1(b), et nous avons tiré aléatoirement ce nombre pour chaque portée.

Nous avons également estimé la durée moyenne de vie d'un lapin suivant une répartition par histogramme (figure 3.2). Pour cette simulation, nous avons tiré l'âge de mort du lapin dès sa naissance. Ainsi, dès qu'il arrivera à l'âge en question, il mourra. On peut remarquer que les chances de mourir augmentent assez rapidement : en effet, sur le campus des Cézeaux, les lapins qui vivent plus de 5 ans se font rares. On estime donc qu'à partir de 5 ans, la probabilité qu'un lapin vive encore un an diminue de plus en plus. De même, on estime que le taux de mortalité infantile est assez élevée, cela étant dû à l'inconscience des lapereaux lorsqu'ils découvrent le campus, et au fait qu'ils sont sans expérience lorsqu'ils se retrouvent face à une voiture. Ce taux de mortalité baisse par la suite, car on estime qu'ils sont plus conscients du danger, et donc plus aptes à l'éviter.



(a) Histogramme du nombre de gestation par femelle et par an.



(b) Histogramme du nombre de petits d'une femelle par portée.

FIGURE 3.1 – Répartition théorique de la reproduction des lapins. *Source personnelle.*

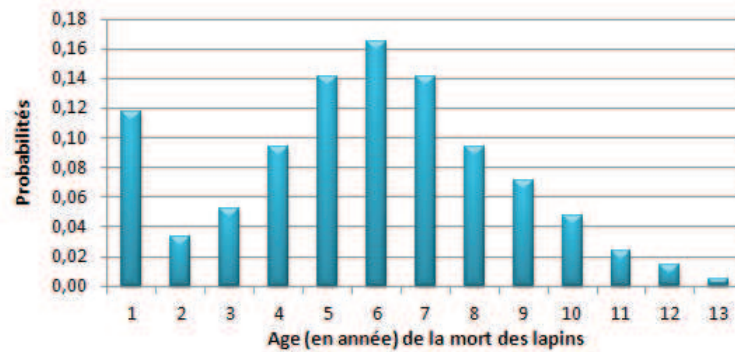


FIGURE 3.2 – Reproduction théorique de l'âge de décès des lapins. *Source personnelle.*

3.3 Résultats

3.3.1 Performances

Nous avons testé les performances de notre simulateur en fonction du nombre de lapins manipulés durant l'itération. Les résultats pour 4 exécutions effectuées sur le serveur Omer de l'Isima sont reportés dans la figure 3.3.

On constate que le temps d'exécution dépend linéairement du nombre de lapins manipulés, sur Omer comme sur une machine Windows (voir résultats dans l'annexe [6]).

Nous avons étudié l'implémentation que nous avons fait du modèle, nous allons maintenant analyser le modèle lui même et les résultats qu'il propose.

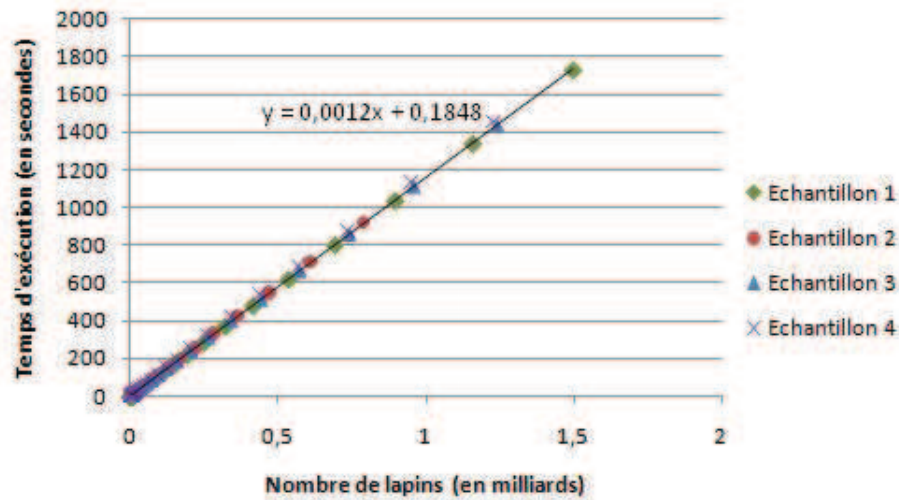


FIGURE 3.3 – Temps d'exécution en fonction du nombre de lapins générés sur le serveur Omer de l'Isima. *Source personnelle.*

3.3.2 Evolution sur 3, 4 et 5 ans

Nous avons tout d'abord simulé la reproduction de la population sur différentes longueurs de temps. Sur les graphiques de la figure 3.4 sont présentés des résultats obtenus pour des simulations sur 3, 4 et 5 ans.

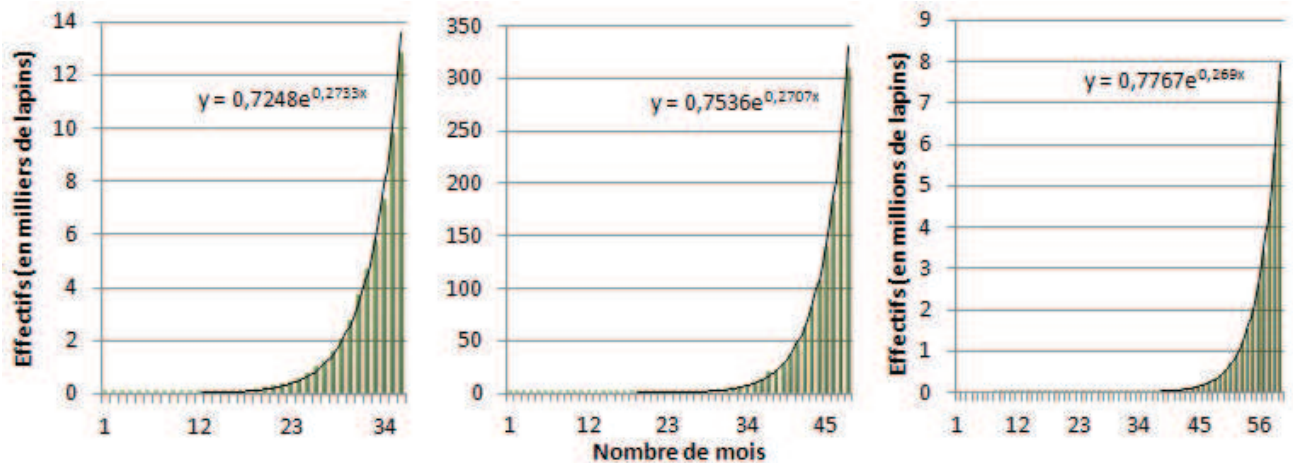


FIGURE 3.4 – Evolution de la taille de la population en fonction des mois, simulée sur différentes durées. *Source personnelle.*

Il apparaît sur ces graphiques que, quelque soit le temps d'étude, la population obtenue suit une croissance exponentielle. Ce résultat est parfaitement logique : plus il y a de lapins adultes, plus on obtient de lapereaux.

On constate également que les facteurs de croissance (paramètres de l'équation modélisant l'évolution de la population) sont presque toujours les identiques, que l'on étudie

sur 3, 4 ou 5 ans la population. Ainsi, la durée d'étude ne modifie en rien le comportement de la population. Cette propriété est essentielle pour la validité du modèle.

3.3.3 Coefficient d'accroissement de la population

Puisque la population croît de manière exponentielle, nous nous sommes intéressées au taux d'accroissement de la population. Nous avons calculé le coefficient qui permet de passer du nombre de lapins en un mois au nombre obtenu le mois suivant pour plusieurs échantillons sur 60 mois. Les valeurs obtenues sont en annexe [7] et nous ont permis d'obtenir le graphique 3.5.

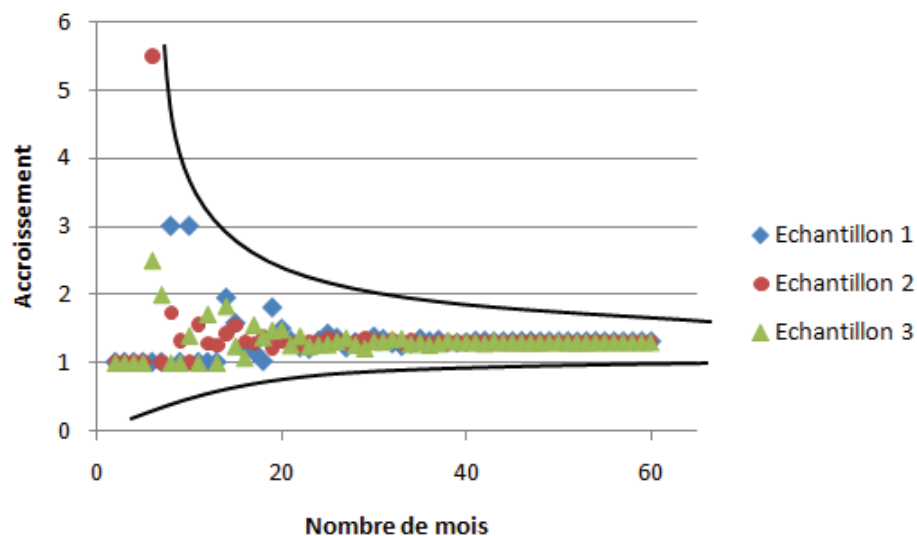


FIGURE 3.5 – Ratio d'accroissement de la population en fonction du mois. *Les valeurs utilisées sont celles de l'annexe [6]. Source personnelle.*

On constate que le ratio d'accroissement de la population est réparti autour de environ 1,30. Plus le mois d'intérêt est éloigné du commencement de l'étude, et donc le nombre de lapins important, plus les variations du taux d'accroissement autour de environ 1,30 sont faibles.

Nous avons donc déterminé précisément le coefficient qui permet de passer d'une itération d'évolution de la population à celle de l'année suivante (cf tableau 3.1). Nous avons utilisé des itérations suffisamment éloignées du départ de la simulation puisque nous avons vu que les premières ont un taux d'accroissement très variables.

Sur le tableau 3.1, la moyenne du ratio est de 21,9652, la variance de 0,0096 et l'écart-type de 0,0980 soit 0,4461%.

Selon notre simulation, la population augmente d'entre 21,8672 et 22,0631% chaque année à partir de la 3ème année. Il est donc possible de prédire l'évolution de la population de lapins d'une année sur l'autre de manière vague en multipliant le nombre de lapins par ce coefficient d'accroissement lorsque l'on dispose d'une population importante.

Nombre d'années	Simulation 1		Simulation 2		Simulation 3		Simulation 4		Simulation 5		Simulation 6		Simulation 7	
	Nombre de lapins	Ratio	Nombre de lapins	Ratio	Nombre de lapins	Ratio	Nombre de lapins	Ratio	Nombre de lapins	Ratio	Nombre de lapins	Ratio	Nombre de lapins	Ratio
3	17 978	-	22 093	-	10 734	-	9 956	-	9169	-	3 859	-	11 419	-
4	394 991	21,9708	481 710	21,8037	236 893	22,0694	219 388	22,0358	199 045	21,7085	84 096	21,7922	250 025	21,8955
5	8 683 677	21,9845	10 609 516	22,0247	5 241 054	22,1241	4 801 687	21,8867	4 395 764	22,0843	1 848 437	21,9801	5 493 040	21,9700
6	190 809 032	21,9733	233 493 117	22,0079	115 170 122	21,9746	105 532 945	21,9783	96 772 034	22,0148	40 559 866	21,9428	120 987 434	22,0256
7	-	-	-	-	-	-	-	-	-	-	891 754 229	21,9861	-	-

Tab. 3.1 – Ratio d'accroissement de la population par année. *Certaines cases sont non renseignées car la simulation ne donne pas ces informations. Source personnelle.*

3.3.4 Taille de la population après 5 ans

Nous avons étudié les proportions que peut prendre une population de lapins au bout de 5 ans. Des probabilités calculées sur 615 simulations nous permettent d'obtenir les résultats présentés dans le graphique 3.6.

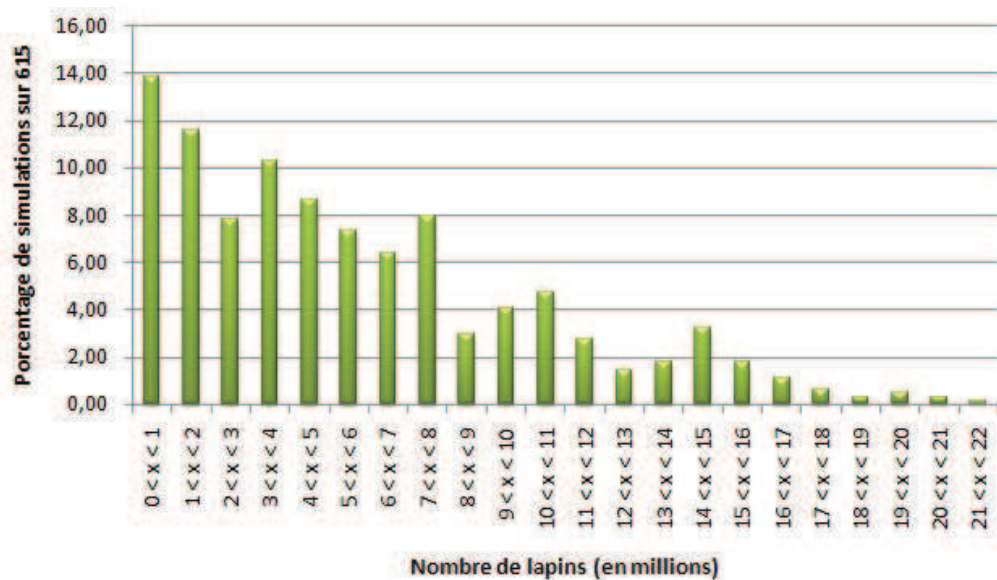


FIGURE 3.6 – Probabilités des tailles de population que l'on peut obtenir en 5 ans. *Source personnelle.*

Nous concluons qu'il est très probable d'avoir moins de 10 millions de lapins en 5 ans (80,91%), avec une probabilité importante d'avoir moins de 5 millions de lapins (52,20%). Même si nous avons observé quelques résultats particulièrement importants (20 ou 30 millions de lapins), ces cas relèvent de l'exception (moins d'1% de probabilité).

Conclusion

Le calcul de π grâce à la Méthode de Monte Carlo nous a permis de comprendre l'importance du nombre d'itérations pour avoir un résultat précis. Cette précision est visible par l'intermédiaire de la moyenne, mais surtout des écart-types et des intervalles de confiance. En effet, sans des écart-types et des intervalles de confiance qui mesurent la proximité des valeurs par rapport à la moyenne, celle-ci n'apporte aucune information significative sur la précision.

Pour l'exercice de simulation sur les lapins, nous avons pu voir qu'un modèle déterministe comme Fibonacci ne peut simuler convenablement l'évolution d'un système complexe, tel qu'une population de lapins. Modéliser un événement qui contient une part de hasard nécessite l'emploi d'outils stochastiques pour être réaliste.

Nous avons pu établir un modèle général correct et qui semble correspondre aux résultats de nos camarades. Cependant, pour être fidèle à un cas réel, il lui manque la prise en compte de nombreux paramètres environnementaux (terrain, nature et quantité de la nourriture, des prédateurs, maladies et des concurrents...) ainsi que certains paramètres intrinsèques, qui peuvent varier suivant les espèces. Ainsi, notre modèle peut fournir des résultats très écartés de résultats observés en réalité : par exemple, une population de 24 lapins de garenne introduits en Australie en 1874 a proliféré pour donner 30 millions de lapins en 50 ans alors que nous atteignons généralement ce nombre d'animaux en moins de 6 ans avec notre modèle.



Références bibliographiques

- [1] Chezclio. Bunnies de walt disney [en ligne], 2008. Consultée le 21 novembre 2010. [http ://chezclio.centerblog.net/rub-Bunnies-de-Walt-Disney.html](http://chezclio.centerblog.net/rub-Bunnies-de-Walt-Disney.html).
- [2] Benny Hill. Introduction à la simulation, application à la simulation aléatoire à événements discrets, 2009. Support de cours.
- [3] Wikipédia. Aiguille de buffon [en ligne], 2010. Consultée le 14 novembre 2010. [http ://www.statelem.com/aiguille_de_buffon.php](http://www.statelem.com/aiguille_de_buffon.php).

Ouvrages consultés

- [4] Anonyme. Le nombre d'or [en ligne], 2003. Consultée le 21 novembre 2010. [http ://maths.amateurs.fr/index.php?page=nombreor](http://maths.amateurs.fr/index.php?page=nombreor).
- [5] J.P. Cussonneau. Introduction aux méthodes de monte carlo [en ligne], 2008. Consultée le 20 novembre 2010. [http ://www-subatech.in2p3.fr/incade/RIA/m2p-ria/MethodeMC.pdf](http://www-subatech.in2p3.fr/incade/RIA/m2p-ria/MethodeMC.pdf).
- [6] Wikipédia. Australie, section faune et flore [en ligne], 2010. Consultée le 15 décembre 2010. [http ://fr.wikipedia.org/wiki/Australie](http://fr.wikipedia.org/wiki/Australie).



ANNEXES

Table des annexes

<u>Annexe 1</u> : Signature des fonctions utilisées pour le calcul de π par une méthode de Monte Carlo.....	II
<u>Annexe 2</u> : Code source en langage C de la simulation d'une reproduction de lapins suivant la suite de Fibonacci	II
<u>Annexe 3</u> : Signature des fonctions utilisées pour initialiser les probabilités pour la reproduction de lapins (partie 3)	III
<u>Annexe 4</u> : Source du fichier Lapin.hpp, code de la classe Lapin, en langage C++	III
<u>Annexe 5</u> : Source C++ de la méthode vieillir	IV
<u>Annexe 6</u> : Tableau et graphique des temps d'exécution sous Windows, à raison de 12 simulations par intervalle de nombre de lapins	V
<u>Annexe 7</u> : Tableau des ratios d'accroissement d'un mois sur l'autre.....	VI

Annexe 1 : Signature des fonctions utilisées pour le calcul de π par une méthode de Monte Carlo

```
double simuSimple (int A, int B) ;
/* Précondition :      A et B sont deux entiers positifs
   Postcondition :     retourne un réel compris entre A et B */

double monteCarlo(int nbExperiences) ;
/* Précondition :      nbExperiences est un entier positif
   Postcondition :     retourne un réel issu du calcul de PI par une
méthode de Monte Carlo, selon nbExperiences itérations */

double estimVariance ( double * PI, double moy, int nbIter ) ;
/* Précondition :      PI est un tableau de réels, préalablement calculés
grâce à monteCarlo, moy est un réel, nbIter est un entier positif
   Postcondition :     renvoie la variance calculée entre les réels
contenus dans PI et moy */

double rayonConfiance ( double var, int nbIter, double quantile ) ;
/* Précondition :      var et quantile sont deux réels, nbIter est un
entier positif
   Postcondition :     retourne le rayon de confiance associé : quantile *
sqrt(var/nbIter); */

void intervalle ( double *borneInf, double *borneSup, double rayon, double
moy) ;
/* Précondition :      rayon et moy sont deux réels, borneInf et borneSup
sont deux pointeurs de réels
   Postcondition :     modifie borneInf et borneSup pour qu'ils
contiennent les bornes inférieure et supérieure de l'intervalle de
confiance calculé à partir de rayon et de moy */
```

Annexe 2 : Code source en langage C de la simulation d'une reproduction de lapins suivant la suite de Fibonacci

```
/* Précondition :      nbSaisons est un entier positif
   Postcondition :     retourne le nombre de lapins au bout de nbSaisons
saison, soit la valeur de la suite de Fibonacci à l'itération nbSaisons, et
affiche le nombre de lapins à chaque itération */
unsigned long long int simuLapinou (int nbSaisons)
{
    unsigned long long int nbAdultes = 0, nbAdos = 0, nbBebes = 1;
    unsigned long long int total = nbAdultes + nbAdos + nbBebes;
    int i;
    printf("Nombre de couples de lapinous a la saison 1 : %d\n", total);

    for ( i = 1 ; i < nbSaisons ; i++ )
    {
        nbAdultes += nbAdos;
        nbAdos = nbBebes;
        nbBebes = nbAdultes;
        total = nbAdultes + nbAdos + nbBebes;
        printf("Nombre de couples de lapinous a la saison %d : %llu\n",
i+1, total);
    }

    return total;
}
```

Annexe 3 : Signature des fonctions utilisées pour initialiser les probabilités pour la reproduction de lapins (partie 3)

```
double simuSimple (int A, int B);
/* Précondition :      A et B sont deux entiers positifs
   Postcondition :     retourne un réel compris entre A et B */

void creationHist (int *T, int size, double *hist);
/* Précondition :      T est un tableau d'entiers de taille size, hist est
un tableau de double
   Postcondition :     remplit hist suivant les valeurs de T, histogramme
normalisé sur 1 */

int simulationHist (double *hist, int size);
/* Précondition :      hist est un tableau de réels de taille size
   Poscondition :      renvoie une valeur, tirée aléatoirement suivant
l'histogramme hist */

void initProbas (int * TNbBebes, int * TMortalite, int * TNbPorteesAn);
/* Précondition :      TNbBebes, TMortalite et TNbPorteesAn sont des
tableaux d'entiers
   Postcondition :     remplit les tableaux avec des valeurs définies
préalablement */
```

Annexe 4 : Source du fichier Lapin.hpp, code de la classe Lapin, en langage C++

```
class Lapin
{
private :
    int    _age;                // sera compté en mois
    bool   _maturite;           // 3 mois pour les mâles, 4 pour les femelles
    int    _sexe;               // 0 pour les femelles, 1 pour les mâles
    int    _nbPorteesAn;        // sera positif uniquement pour les femelles
    arrivées à maturité
    bool   _enceinte;           // uniquement pour les femelles. La période de
    gestation est fixée à un mois
    int    _mort;               // donne l'age de la mort du lapin

    static int NbFemelles;
    static int NbMales;
    static int NbMalesMatures;

public :
    Lapin (double * histMortalite); // histMortalite servira à déterminer
    l'âge de la mort du lapin
    Lapin (int inSexe, double * histMortalite); // servira pour créer les
    deux premiers lapins de l'histoire des Cézeaux
    ~Lapin ();

    int getNbLapins () ; // retourne le nombre total de lapins
    int vieillir (double * histNbPorteesAn, double * histNbBebes);
    /* Précondition : histNbPorteesAn et histNbBebes sont des tableaux de
    réels. Postcondition : vieillit le lapin d'un mois, et renvoie : -1 s'il
    meurt, 0 s'il n'a pas engendré d'autres lapins, le nombre de lapins
    enfantés sinon */
    int mettreBas (double * histNbBebes); /* Précondition : histNbBebes
    est un tableau de réels. Postcondition : retourne le nombre de bébés qu'une
    femelle a engendré, suivant les probabilités de histNbBebes */
};
```

Annexe 5 : Source C++ de la méthode vieillir

```
int Lapin::vieillir (double * histNbPorteesAn, double * histNbBebes)
{
    int res = 0;
    _age ++;

    /* Changements de maturité */
    if( _sexe == 0 && (_age == 4 || _age % 12 == 0) ) // Femelle
    {
        _maturite = true;
        _nbPorteesAn = simulationHist(histNbPorteesAn, 7) + 1;
    }
    else if( _sexe == 1 && _age == 3) // Male
    {
        _maturite = true;
        NbMalesMatures ++;
    }

    /* Possibilité de mourir */
    if ( _age == _mort )
    {
        res = -1;
        if( _sexe == 0)
            NbFemelles --;
        else
        {
            NbMales --;
            if( _maturite == true )
                NbMalesMatures --;
        }
    }

    /* Possibilité d'avoir des bébés */
    if( _sexe == 0 && _enceinte == true)
        res = mettreBas( histNbBebes );

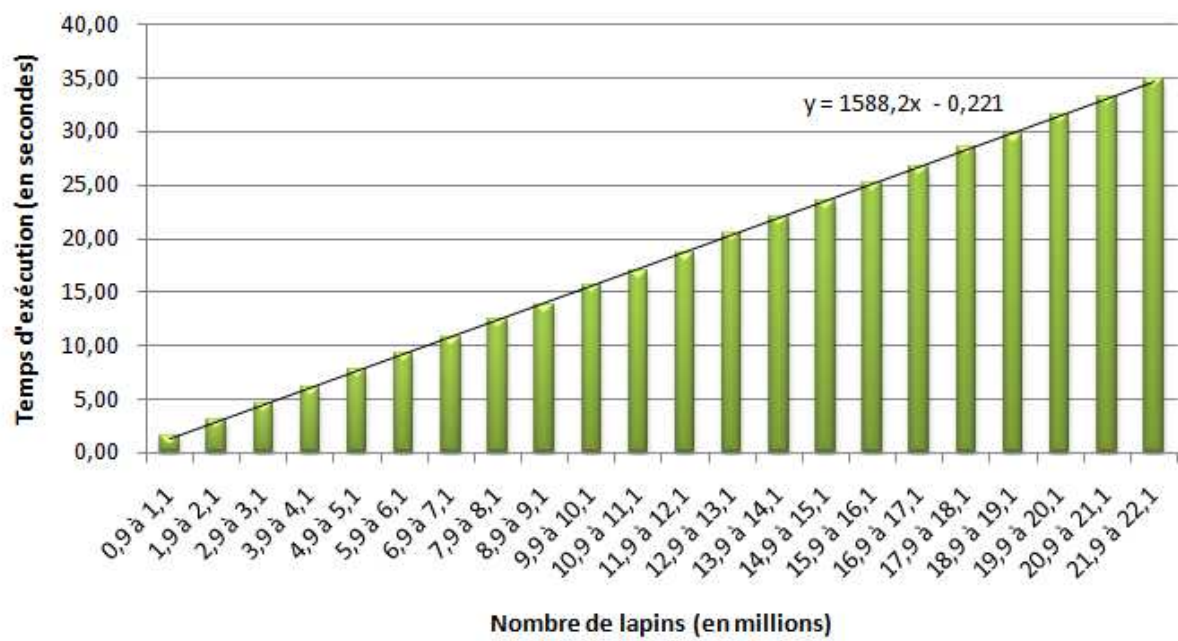
    /* Possibilité de tomber enceinte */
    if( _sexe == 0 && _maturite == true) // Si la femelle qui peut avoir
des bébés
    {
        if ( ( (float)NbFemelles/NbMalesMatures ) < 90)
            /* On considère qu'un mâle mature peut féconder jusqu'à 3
femelles par jour, soit 90 par mois */
        {
            int proba = rand () % 12 + 1;
            if ( proba <= _nbPorteesAn )
                _enceinte = true;
        }
    }

    return res;
}
```


Annexe 6 : Tableau des temps d'exécution sous Windows, à raison de 12 simulations par intervalle de nombre de lapins

Nombre de lapins (en millions)	Essai 1 (en ms)	Essai 2 (en ms)	Essai 3 (en ms)	Essai 4 (en ms)	Essai 5 (en ms)	Essai 6 (en ms)	Essai 7 (en ms)	Essai 8 (en ms)	Essai 9 (en ms)	Essai 10 (en ms)	Essai 11 (en ms)	Essai 12 (en ms)	Moyenne	Ecart- type (en %)
0,9 à 1,1	1 590	1 543	1 451	1 467	1 463	1 497	1 450	1 517	1 466	1 626	1 650	1 623	1 528,58	4,93
1,9 à 2,1	3 121	3 198	3 166	3 121	3 167	2 946	3 180	3 213	3 028	3 182	3 182	3 011	3 126,25	2,74
2,9 à 3,1	4 554	4 588	4 522	4 587	4 714	4 541	4 631	4 617	4 588	4 713	4 728	4 461	4 603,67	1,79
3,9 à 4,1	6 226	6 380	6 118	6 205	6 256	6 209	9 086	6 116	6 179	5 961	5 912	6 177	6 152,08	2,06
4,9 à 5,1	7 693	7 676	7 629	7 766	7 783	7 692	7 566	7 989	7 582	7 740	7 568	7 675	7 696,58	1,53
5,9 à 6,1	9 219	9 349	9 331	9 565	9 389	9 437	9 232	9 313	9 126	9 424	9 329	9 378	9 341	1,23
6,9 à 7,1	10 684	10 714	10 980	10 671	10 823	10 813	10 622	10 749	10 826	10 699	10 735	10 621	10 744,75	0,95
7,9 à 8,1	12 416	12 543	12 280	12 340	12 590	12 341	12 372	12 606	12 528	12 466	12 307	12 483	12 439,33	0,90
8,9 à 9,1	13 850	13 808	13 994	13 927	13 885	14 068	13 836	13 947	13 913	13 665	13 933	13 787	13 884,42	0,76
9,9 à 10,1	15 553	16 054	16 054	15 535	15 771	15 281	15 866	15 803	15 805	15 555	15 745	15 698	15 709,33	1,14
10,9 à 11,1	16 907	17 348	17 348	17 427	17 006	16 864	16 820	16 835	16 864	16 963	17 010	17 214	17 037,58	1,22
11,9 à 12,1	18 551	18 782	18 752	18 799	18 410	18 548	18 876	18 692	18 648	18 721	18 698	18 740	18 684,75	0,69
12,9 à 13,1	20 248	20 489	20 440	20 843	20 200	20 561	20 532	20 920	20 541	20 880	20 522	20 560	20 561,33	1,10
13,9 à 14,1	21 826	21 701	21 995	21 935	22 448	21 961	22 075	21 734	21 920	22 153	22 005	21 840	21 966,08	0,91
14,9 à 15,1	23 320	23 371	23 698	23 557	23 431	23 584	23 335	23 456	23 688	23 345	23 470	23 564	23 484,92	0,57
15,9 à 16,1	25 100	25 178	25 444	25 116	25 145	25 289	25 458	25 015	25 454	25 269	25 240	25 220	25 244	0,58
16,9 à 17,1	26 682	26 633	26 814	26 941	26 647	26 784	26 589	26 888	26 987	26 547	26 688	26 574	26 731,17	0,55
17,9 à 18,1	28 361	28 253	28 581	28 393	28 385	28 667	28 393	28 465	28 669	28 666	28 528	28 788	28 512,42	0,57
18,9 à 19,1	29 642	29 765	29 566	29 880	29 740	29 688	29 547	29 756	29 777	29 851	29 710	29 655	29 714,75	0,34
19,9 à 20,1	31 499	31 903	31 358	31 449	31 568	31 644	31 958	31 677	31 354	31 599	31 488	31 589	31 590,5	0,60
20,9 à 21,1	33 024	33 461	33 259	32 810	33 370	33 341	33 224	33 212	33 585	33 014	32 890	33 125	33 192,92	0,70
21,9 à 22,1	34 761	35 004	35 163	34 728	34 988	35 069	35 247	35 068	34 772	34 998	35 020	35 121	34 994,92	0,47

Graphique associé au tableau précédent :



Annexe 7 : Tableau des ratios d'accroissement d'un mois sur l'autre

Nombre de mois	Simulation 1		Simulation 2		Simulation 3	
	Nombre de lapins	Ratio	Nombre de lapins	Ratio	Nombre de lapins	Ratio
1	2		2		2	
2	2	1	2	1	2	1
3	2	1	2	1	2	1
4	2	1	2	1	2	1
5	2	1	2	1	2	1
6	2	1	11	5,5	5	2,5
7	2	1	11	1	10	2
8	6	3	19	1,727272727	10	1
9	6	1	25	1,315789474	10	1
10	18	3	25	1	14	1,4
11	18	1	39	1,56	14	1
12	18	1	50	1,282051282	24	1,71428571
13	18	1	63	1,26	24	1
14	35	1,944444444	90	1,428571429	44	1,83333333
15	55	1,571428571	139	1,544444444	55	1,25
16	65	1,181818182	181	1,302158273	59	1,07272727
17	75	1,153846154	231	1,276243094	92	1,55932203
18	76	1,013333333	318	1,376623377	127	1,38043478
19	137	1,802631579	388	1,220125786	188	1,48031496
20	204	1,489051095	506	1,304123711	280	1,4893617
21	268	1,31372549	646	1,276679842	353	1,26071429
22	326	1,21641791	802	1,241486068	492	1,39376771
23	389	1,193251534	1048	1,306733167	614	1,24796748
24	511	1,313624679	1364	1,301526718	777	1,26547231
25	724	1,416829746	1831	1,342375367	990	1,27413127
26	982	1,356353591	2441	1,333151283	1305	1,31818182
27	1192	1,213849287	3164	1,296190086	1785	1,36781609
28	1543	1,294463087	4143	1,309418458	2265	1,26890756
29	1992	1,290991575	5572	1,344919141	2752	1,21501104
30	2735	1,372991968	7447	1,336503948	3615	1,31359012
31	3654	1,336014625	9505	1,276352894	4743	1,3120332
32	4655	1,27394636	12687	1,334771173	6402	1,34977862
33	5750	1,235230934	16414	1,293765272	8685	1,35660731
34	7305	1,270434783	21760	1,325697575	11111	1,27933218
35	9804	1,342094456	28601	1,314384191	14521	1,30690307
36	12808	1,306405549	36847	1,288311597	18376	1,26547758
37	16967	1,324718926	47735	1,29549217	23621	1,28542664
38	21885	1,289856781	61753	1,293662931	31445	1,33123068
39	28103	1,284121544	80783	1,308163166	40973	1,30300525
40	36296	1,291534712	105889	1,310783209	53469	1,30498133
41	47618	1,311935199	138071	1,303922032	69788	1,30520489
42	62495	1,312423873	179395	1,29929529	89631	1,28433255
43	81503	1,304152332	234278	1,305933833	118221	1,31897446

44	106401	1,305485688	305937	1,305871657	153894	1,30174842
45	138753	1,304057293	398827	1,303624602	201283	1,30793273
46	181400	1,30735912	520523	1,305134808	262209	1,30268825
47	237224	1,307739802	677894	1,302332462	340600	1,2989638
48	309471	1,304551816	883054	1,302643186	443554	1,30227246
49	404174	1,306015749	1153668	1,30645238	578682	1,30464836
50	526855	1,303535111	1502158	1,302071306	757207	1,30850277
51	687609	1,305120005	1956453	1,30242824	985335	1,30127561
52	894917	1,301491109	2550429	1,3035984	1283302	1,30240172
53	1167773	1,304895314	3325720	1,303984545	1670969	1,30208556
54	1526269	1,30699117	4340370	1,30509183	2179588	1,30438566
55	1987419	1,302142021	5656321	1,303188668	2843230	1,30448048
56	2592554	1,304482849	7373885	1,303653912	3707301	1,30390471
57	3380415	1,303893767	9606073	1,302715326	4833757	1,303848
58	4405240	1,30316544	12524866	1,303848722	6300628	1,30346395
59	5753015	1,305948144	16328389	1,30367774	8215803	1,30396573
60	7497029	1,303147828	21285210	1,303570732	10707629	1,30329671