

# Simulation de Monte Carlo et Calcul d'intervalles de confiance

Maxime Escourbiac Jean-Christophe Septier

---

## Objectif

l'objectif de ce tp est d'effectuer plusieurs simulations afin de manipuler divers outils statistiques comme les moyennes, variances et intervalles de confiances.

Pour la partie programmation, la partie I portera sur une simulation de Monte Carlo. Tandis que la dernière partie porte plus sur un mini-projet avec élaboration d'un modèle, choix des générateurs aléatoires.

---

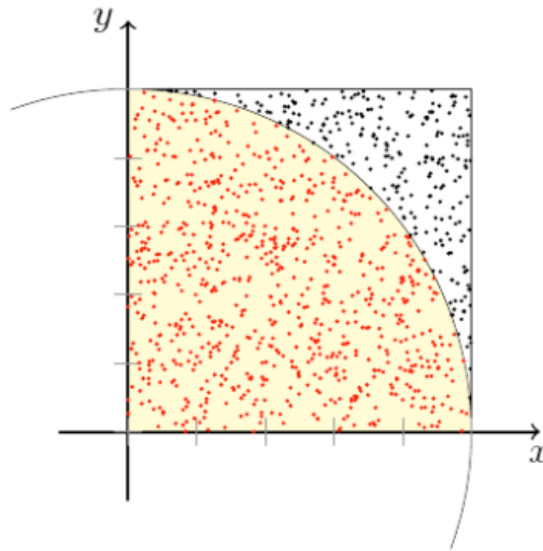
## Plan

- I. Calcul de PI avec une simulation de Monte Carlo
  - Principe de la simulation
  - Choix de l'implémentation
  - Résultats obtenues
- II. Simulation de croissance de lapins. Modèle de Fibonnaci
- III. Simulation de croissance de lapins. Modèle plus réaliste.
  - Etude du sujet
  - Elaboration des générateurs
  - Construction du diagramme UML
  - Choix sur l'implémentation
  - Etude des résultats
- IV. Conclusion

## I Calcul de PI avec une simulation de Monte Carlo

### 1°) Principe de simulation:

Représentation d'une réalisation d'une expérience:



source image : <http://learntofish.files.wordpress.com/>

On sait que l'aire d'un disque de rayon 1 est égale à  $\pi$  donc l'aire d'un  $1/4$  de cercle égale à  $\pi/4$ . On définit un carré de côté 1 on effectue 10000 tirs aléatoires dans le carré et on comptabilise le nombre de tir dans le  $1/4$  de cercle et on obtient la relation suivante:

$$\pi/4 = \text{nombre sur tir dans le cercle} / \text{nombre de tir total}$$

### 2°) Choix de l'implémentation:

Langage choisi: C

Choix du générateur: Mersenne Twister version C pour avoir un générateur de qualité.

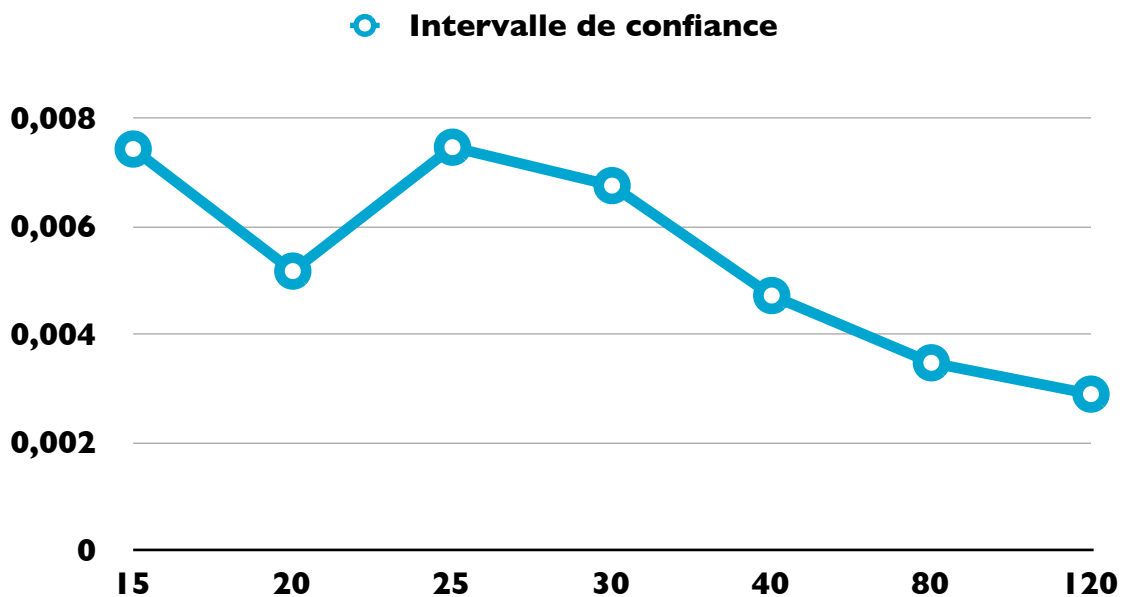
Nombre de tirs par boucle 10000.

### 3°) Résultats:

Les résultats seront exprimés sous forme de tableau

Nombre de réplifications	15	20	25	30	40	80	120
Moyenne	3.139920	3.141110	3.144176	3.144120	3.141850	3.138560	3.141833
Intervalle de confiance	0,007433	0,005173	0,007467	0,006756	0,004720	0,003475	0,002895

variations de l'intervalle de confiance selon le nombre de réplifications:



### Conclusion:

On remarque que quelque soit le nombre de réplifications, la moyenne ne semble pas varier, par contre on peut remarquer que l'intervalle de confiance diminue en fonction du nombre de réplifications

## II Simulation de croissance de lapins. Modèle de Fibonacci.

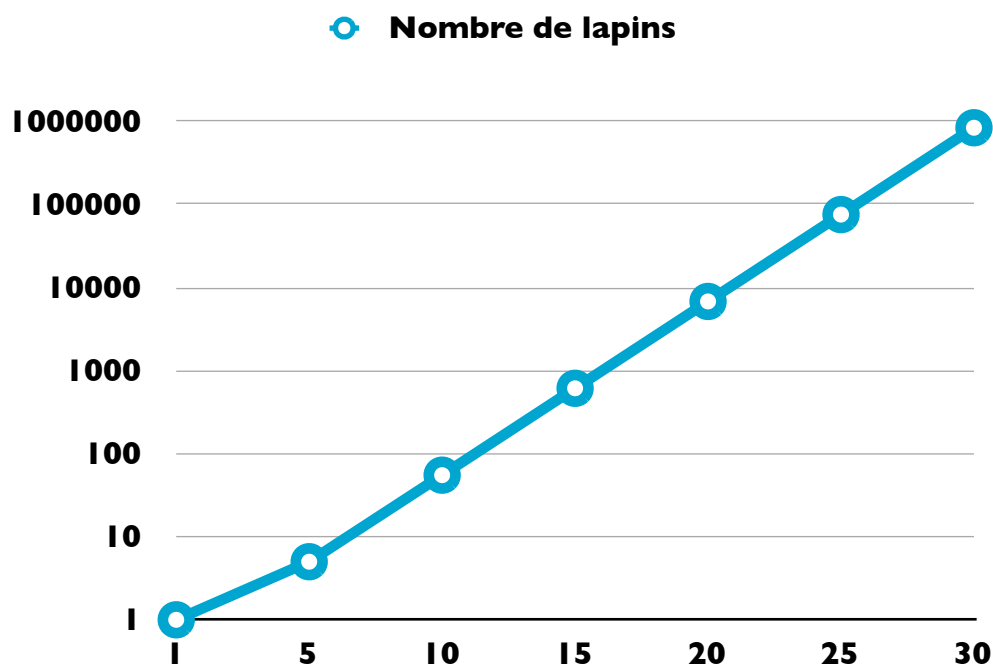
Ce programme calcule le  $i$  élément de la suite de fibonacci:

La suite de Fibonacci est définie comme ceci:

$$U_{n+2} = U_{n+1} + U_n \quad \text{avec } U_0 = U_1 = 1$$

Evolution du nombre de lapin en fonction du nombre de saison:

n	1	5	10	15	20	25	30
$U_n$	1	5	55	610	6765	75025	832040



### III Simulation de croissance de lapins. Modèle plus réaliste.

#### 1°) Etude du sujet:

On relève les données du problème, et on formule les hypothèses correspondantes:

Données du problèmes	Hypothèses associés
Gestation: 28 à 33 jours	Gestation : 1 mois
Portées/an : 1 à 7 -> 3,4,5 plus souvent	Aucune simplification
Lapins/portée -> 3 à 12	On favorise les portées moyennes
Majorité sexuelle: 3mois pour le mâle 4 mois pour la femelle approximativement	On fixe la majorité sexuelle à 3 mois pour le mâle et 4 mois pour la femelle.
Mortalité autour de 9 ans	On prendra en compte la mortalité infantile dans le générateur.

#### 2°) Elaboration des générateurs

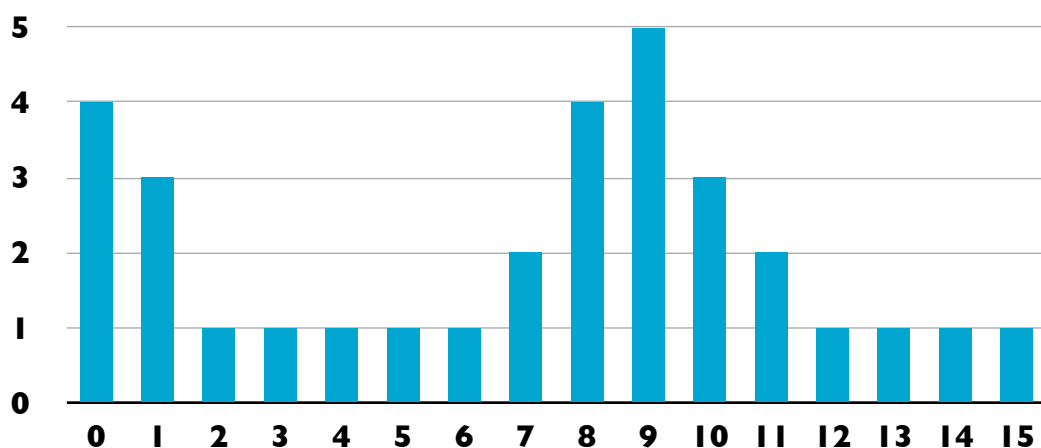
##### - Durée de vie des lapins :

On ne fait pas de distinctions entre les sexes, pour leur durée de vie. On rajoute par rapport au sujet une certaine mortalité infantile (entre 0 et 2ans) . Donc on ne peut plus modéliser la durée de vie des lapins avec une loi normale.

On a décidé alors d'utiliser une modélisation par diagramme pour la durée de vie en année ( densité donnée par l'histogramme ci-dessous) et pour modéliser le mois de mort du lapin on utilise un générateur équiprobable en 0 et 11.

$$\text{Durée de vie} = \text{Durée de vie en année} \times 12 + \text{Mois de mort}$$

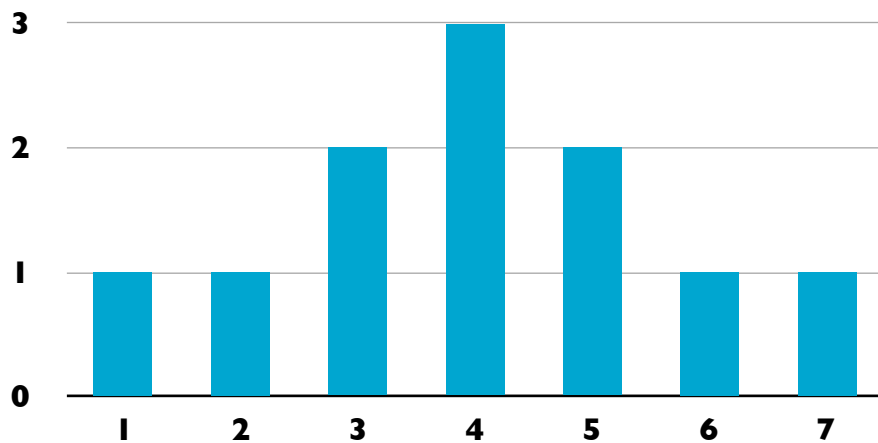
Diagramme de densité de la durée de vie en année des lapins:



- Nombre de portée par an:

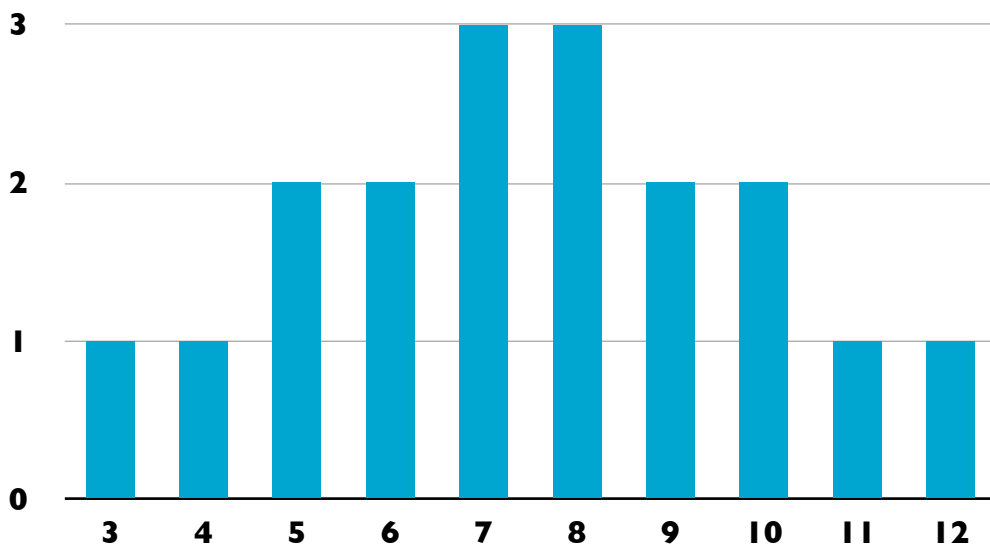
On reprend l'énoncé, avec une possibilité d'avoir entre 1 et 7 portée par an, avec une plus grande chance pour en avoir 3,4,5.

Donc on a l'histogramme suivant:

-Nombre de lapins par portée:

On nous indique sur le sujet un nombre de lapins par portée entre 3 et 12 sans nous préciser la répartition, on a pris la liberté de fixer une densité telle qu'on favorise les portées de 7,8 lapins

Donc on a l'histogramme suivant:

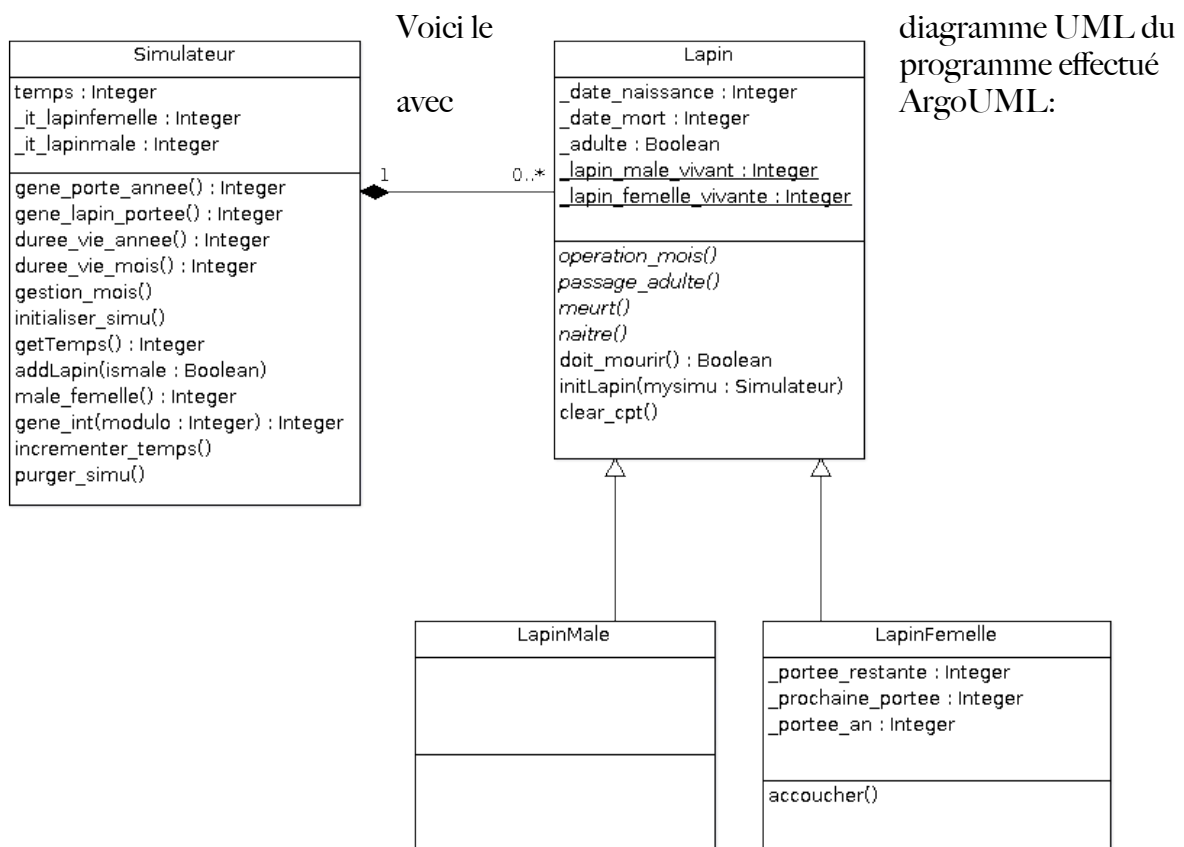


### - Implémentation des générateurs

Tous les générateurs utilisés, (à part le générateur qui définit le sexe du lapin à sa naissance qui est simplement un Random équiprobable entre 0 et 1, problème de pile et face), sont basés sur des histogrammes donc on a décidé de les implémenter comme ceci:

- On fixe un tableau d'entier image de l'histogramme (Ex: si un événement X à une probabilité de  $\frac{4}{15}$  d'être tiré alors on met 4 X dans le tableau d'entier)
- Pour réaliser le tirage, on tire un nombre aléatoire et on retourne la valeur qui lui est associé ans le tableau.
- Les générateurs sont basés sur le générateur MERSENNE TWISTER.

### 3°) Construction du diagramme UML:



On remarque que le Lapin Mâle ne possède aucun attribut propre (Dans la simulation on ne prend en compte que sa présence pour accoupler les femelles).

#### 4°) Choix sur l'implémentation:

Choix du pas de temps pour la simulation : 1 mois car la simulation demande par forcément une précision à la journée et aussi un choix par critère de performance.

Le stockage en mémoire des lapins s'effectue à l'aide d'un pool de 10 millions de lapins mâle et 10 millions de lapins femelle. Dans la classe Simulateur, on a une list (STL) pour stocker les pointeurs des lapins vivants(mâles et femelles).

Le choix du pool est due à une amélioration des performances considérable ainsi qu'une sécurité au niveau des fuites de mémoires.

Screenshot de Valgrind après une exécution du programme.

```
==3921==
==3921== HEAP SUMMARY:
==3921==   in use at exit: 0 bytes in 0 blocks
==3921==   total heap usage: 45,393,416 allocs, 45,393,416 frees, 544,720,992 bytes allocated
==3921==
==3921== All heap blocks were freed -- no leaks are possible
==3921==
==3921== For counts of detected and suppressed errors, rerun with: -v
==3921== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 19 from 8)
escourbi@escourbi-desktop:/media/travail/zz2/simulation/tp3/dossier/tp3_simu_lapin_v2/source$
```

#### 5°) Etude des résultats:

Pour l'exécution du programme on a réalisé une série de 1000 réplifications d'une simulation sur X mois avec 5 lapins de chaque sexe née au temps 0.

On obtient les résultats suivant:

<b>X (Mois)</b>	12	24	30	36	42	48
<b>Moyenne</b>	41,759	415,549	1249,88	4037,7	13158,6	42509
<b>Ecart-type</b>	14,2694	166,278	487,954	1570,89	5235,45	18558.1
<b>Rayon de l'intervalle de confiance</b>	0,884429	10,306	30,2437	97,3645	324,497	1150.25

#### Conclusion:

Le grand écart-type peut s'expliquer avec l'étendue des possibilités pour le nombre de lapins que peut produire une lapine pendant une année.

On remarque aussi que les résultats pour une réplification sont grandement conditionné par la productivité des 5 lapines qui sont présentes au début de la simulation.

Et pour conclure sur le rayon de l'intervalle de confiance, il augmente en fonction de du nombre de mois de simulation, ce qui est logique car plus on essaye d'aller loin sur la durée plus il est difficile de connaître le nombre de lapins qu'il y aura au temps X.



## IV Conclusion

Le but de ce TP était d'aborder les méthodes de Monte Carlo avec le calcul de PI et surtout d'apprendre à analyser des exécutions de simulations enfin d'en tirer des éléments statistiques et de pouvoir conclure sur la validité et la crédibilité des résultats.

Pour finir, un mot sur la performance du programme de simulation de Lapin, on arrive au maximum de capacités de stockage de Lapin (20millions) en 2s-3s.