

TP

# Simulation réaliste d'une population de lapins



CHASSAGNOL Rémi  
ZZ2 - F2 - promo24

*Professeur:* HILL David

DECEMBER 18, 2022

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Simulation avec Fibonacci</b>	<b>2</b>
<b>3</b>	<b>Simulation réaliste</b>	<b>3</b>
3.1	Calcul du nombre de lapins . . . . .	4
3.1.1	Structure de données . . . . .	4
3.1.2	Les survivants sur une année . . . . .	4
3.1.3	Calcul des naissances . . . . .	5
3.1.4	Calcul du nombre de lapins sur une année . . . . .	6
3.2	Simulation sur n années . . . . .	7
3.3	Résultats . . . . .	7
<b>4</b>	<b>Conclusion</b>	<b>9</b>

## 1 Introduction

Le but de ce tp est de proposer une simulation plus réaliste d'une population de lapins sur un certain nombre d'années. On commencera par une modèle très basique créé par Fibonacci. Ensuite nous étudierons un modèle plus réaliste où les lapins auront une probabilité de survie, un nombre de portées par ans et de petits par portées. Pour ce faire nous utiliserons le PRNG **Mersenne Twister**.

## 2 Simulation avec Fibonacci

Tout d'abord commençons par étudier le modèle de Fibonacci. La suite de Fibonacci est très simple, elle suit le principe suivant:

- $x_0 = x_1 = 1$
- $x_n = x_{n-1} + x_{n-2}$

On part donc avec 2 petits lapins ( $x_0$  et  $x_1$ ) et on simule le nombre lapins sur **n** mois selon la suite décrite plus tôt. Le code de la fonctions est le suivant:

```
void modelFib(int n) {  
    int i;  
    int fn, fn1 = 1, fn2 = 1;  
  
    for (i = 2; i <= n; ++i) {  
        fn = fn1 + fn2;  
        fn2 = fn1;  
        fn1 = fn;  
        printf("mois %d: %d lapins.\n", i, fn);  
    }  
}
```

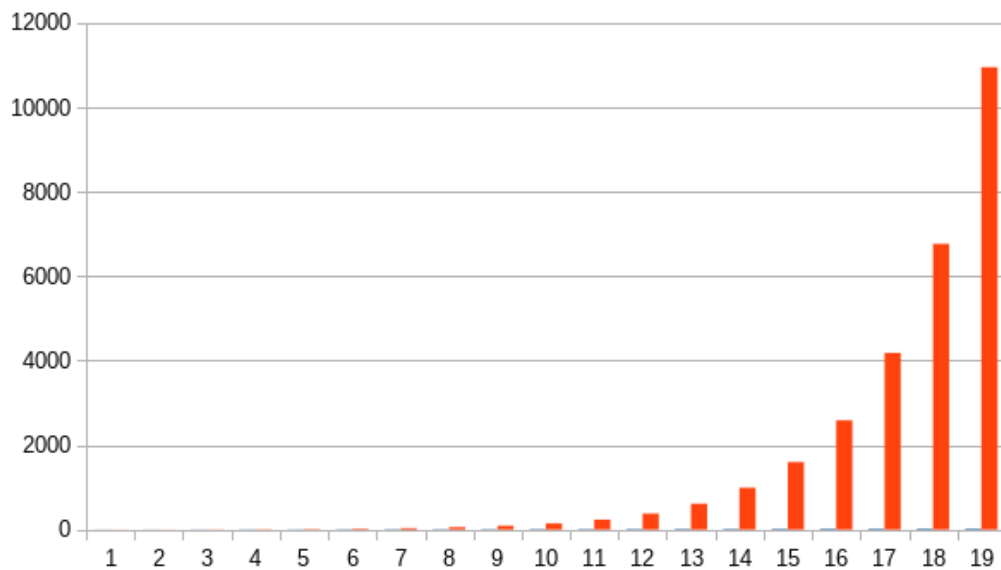


Figure 1: Résultats sur 20 ans

Cette simulation n'est vraiment pas réaliste car tous les lapins produisent le même nombre de petits. De plus on ne prend pas en compte l'âge ou le sexe des lapins et les lapins ont une durée de vie illimitée. C'est pour cela que le nombre de lapins sur  $n$  mois augmente extrêmement vite.

### 3 Simulation réaliste

Intéressons nous maintenant à une simulation plus réaliste. Ici nous prendrons en compte les données suivantes pour calculer le nombre de lapins sur  $n$  années:

- Nombre de portées pour une femelle sur une année: 4 à 8 (calculé selon une loi normale)
- Nombre de petits par portée: 3 à 6 (suivant loi uniforme)
  - 50% de chance pour que le petit soit un mâle ou une femelle.
- Maturité sexuelle: après 1 an
- Survivabilité:
  - Pour les petits: 35%
  - Pour les adultes de moins de 10 ans: 60%
  - Pour les adultes de plus de 10 ans: diminution de 10% par an
  - Un lapin vit au maximum 15 ans (années 0 jusqu'à 14)

À noter qu'ici le pas est l'année, mais on pourrait complexifier la simulation en modifiant les données et en calculant au mois près pour plus de précisions. De plus, ici les lapins ne peuvent mourir que de vieillesse mais on pourrait ajouter d'autres raisons comme par exemple des prédateurs, des accidents naturels (feu, ...) ou encore la probabilité qu'une épidémie se déclare durant une année.

### 3.1 Calcul du nombre de lapins

#### 3.1.1 Structure de données

Pour cette simulation nous utiliserons un tableau de taille **n** qui contiendra les populations de lapins sur les **n** années simulées. Les données sur la population seront stockées dans la structure suivante:

```
typedef struct data {
    unsigned long int nbF[15]; // nombre de femelles de 0 14 ans
    unsigned long int nbM[15]; // nombre de m les de 0 14 ans
    unsigned long int f; // nombre de femelles
    unsigned long int m; // nombre de m les
} data_t;
```

- **nbF**: contient les femelles âgées de 0 (petits) à 14 ans
- **nbM**: contient les mâles âgés de 0 à 14 ans
- **f**: nombre total de femelles adultes
- **m**: nombre total de mâles adultes

#### 3.1.2 Les survivants sur une année

Ici nous avons une fonction qui prend en paramètre **nbLapins** lapins âgés de **age** à l'année **N** et calcule le nombre de survivants pour l'année **N + 1**. Les lapins peuvent être des mâles ou des femelles et la probabilité de survie suit les règles énoncées plus tôt.

```
int nbSurvivants(int age, unsigned long int nbLapins) {
    unsigned long int nbSurvivants = 0;
    unsigned long int i;
    double probaSurvie = 0.6;

    if (age >= 10) // la proba de survie diminue de 10%/ans au dela de 10 ans
        probaSurvie = 0.6 - 0.1 * (age - 10);
    else if (age == 0) // proba de survie pour les petits
        probaSurvie = 0.35;

    for (i = 0; i < nbLapins; ++i) {
        if (genrand_reall() < probaSurvie) {
            nbSurvivants++;
        }
    }

    return nbSurvivants;
}
```

### 3.1.3 Calcul des naissances

Pour le calcul des naissances nous utiliserons deux fonctions. La première fonction permet de calculer le nombre de portées pour une femelle sur une année. Comme dit précédemment, une femelle peut avoir entre 4 et 8 portées par an et le nombre de portées suit une loi normale. Le code est le suivant:

```
int calculNbPortees() {
    int nbPortees = 8;
    int r = genrand_reall();

    if (r < 0.1) {
        nbPortees = 4;
    } else if (r < 0.3) {
        nbPortees = 5;
    } else if (r < 0.7) {
        nbPortees = 6;
    } else if (r < 0.9) {
        nbPortees = 7;
    }

    return nbPortees;
}
```

Ici pour gagner du temps de calcul, la loi normale est codée en dur et suit les valeurs suivantes:

nombre de petits	4	5	6	7	8
probabilité en %	10	20	40	20	10

Ensuite nous avons une fonction qui permet de calculer le nombre de naissances. On commence par calculer le nombre de lapins qui peuvent se reproduire, on considérera qu'il faut au moins un mâle pour deux femelles. Ensuite, pour chaque lapin apte à se reproduire, on calcule le nombre de portées. Puis pour chaque portée, on calcule le nombre de petits (compris entre 3 et 6 suivant une loi uniforme): `nbPetits = genrand_int32()%4 + 3`. Enfin, pour chaque petit, on calcule le sexe (50/50) puis on ajoute le nouveau petit au nombre de petits mâles ou au nombre de petites femelles.

```
void calculNaissance(data_t * populations, int annee) {
    unsigned long i;
    int j, k;
    int nbPortees, nbPetits;
    int estMale;
    unsigned long nbLapins = // on suppose qu'il faut minimum 1 mal pour 2 femelles
        minLapins(populations[annee - 1].f, 2 * populations[annee - 1].m);

    for (i = 0; i < nbLapins; ++i) {
        nbPortees = calculNbPortees();

        for (j = 0; j < nbPortees; ++j) {
            nbPetits = genrand_int32()%4 + 3;
            for (k = 0; k < nbPetits; ++k) {
                estMale = genrand_int32()%2;
                if (estMale) {
                    populations[annee].nbM[0]++;
                } else {
                    populations[annee].nbF[0]++;
                }
            }
        }
    }
}
```

À la fin, `populations[annee].nbF[0]` contient le nombre de petites femelles sur l'année et `populations[annee].nbM[0]` le nombre de petits mâles.

### 3.1.4 Calcul du nombre de lapins sur une année

Pour calculer le nombre de lapins sur une nouvelle année, nous utiliserons la fonction suivante. Le principe est très simple, on commence par faire vieillir les lapins, pour ce faire, on calcule le nombre de survivants chez les mâles et les femelles de l'année précédente (avec la fonction `nbSurvivants`). Le calcul est fait sur les lapins de l'année  $N$  âgés de  $Y$  ans et la valeur calculée est stockée dans la case  $Y + 1$  des lapins de l'année  $N + 1$ . De ce fait, tous les lapins survivants vieillissent de 1 an et les lapins âgés de plus de 15 ans meurent. Ensuite on met le nombre de petits à 0. Une fois fait, on met à jour le nombre de mâles et de femelles adultes.

```
void calculNouvelleAnnee(data_t *populations, int annee) {
    int i;
    int nombreF = 0;
    int nombreM = 0;

    // calcul des survivants chez les adultes
    for (i = 14; i >= 1; --i) {
        populations[annee].nbM[i] = nbSurvivants(i - 1, populations[annee - 1].nbM[i - 1]);
        populations[annee].nbF[i] = nbSurvivants(i - 1, populations[annee - 1].nbF[i - 1]);
    }
    // les petits deviennent adultes
    populations[annee].nbF[0] = 0;
    populations[annee].nbM[0] = 0;

    // calcul du nombre de mal et de femelles adultes pour pouvoir calculer les naissances
    for (i = 1; i < 14; ++i) {
        nombreF += populations[annee].nbF[i];
        nombreM += populations[annee].nbM[i];
    }
    populations[annee].f = nombreF;
    populations[annee].m = nombreM;

    // calcul des naissances
    calculNaissance(populations, annee);
}
```

Enfin, on utilise la fonction décrite précédemment pour calculer le nombre de naissances.

### 3.2 Simulation sur n années

Enfin pour la simulation sur n années, on utilise un tableau d'années dans lequel on stocke les données sur la population durant chaque année. Ces données sont calculées avec la fonction `calculNouvelleAnnee`.

```
data_t* computeNbLapins(int n) {  
    int i;  
    data_t* populations;  
    populations = initPopulations(n);  
  
    for (i = 1; i < n; ++i) {  
        calculNouvelleAnnee(populations, i);  
    }  
  
    return populations;  
}
```

La fonction commence par initialiser le tableau des populations avec la fonction `initPopulations`. Cette fonction alloue un tableau dynamique et initialise la première année avec des nombres de mâles et de femelles adultes de départ (constantes pré-processeur). Ensuite, on remplit le tableau avec `calculNouvelleAnnee`. À la fin de l'exécution, le tableau `populations` contient toutes les statistiques sur la population de chacune des années.

### 3.3 Résultats

Voici quelques statistiques sur la simulation. Dans les 2 premiers graphiques, on peut constater que l'évolution des lapins est **exponentielle**. De plus le nombre de mâles est **équivalent** au nombre de femelles, ce qui est logique compte tenu du fait que la probabilité qu'un petit soit un mâle ou une femelles est de 50%.

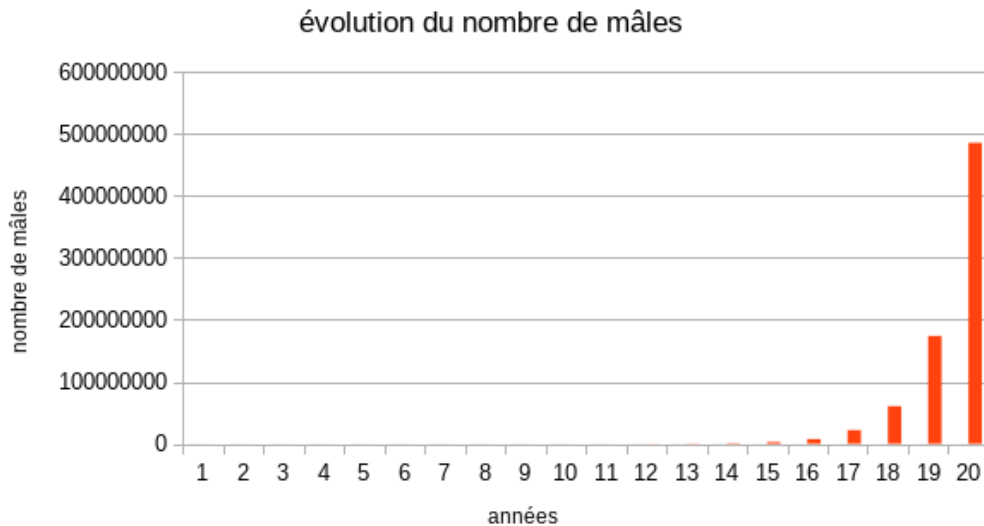


Figure 2: Évolution du nombre de mâles.



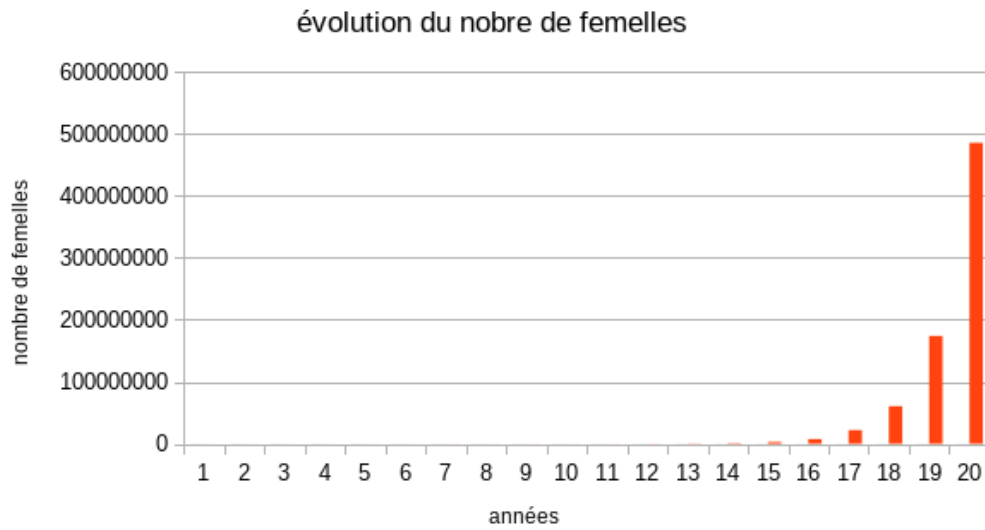


Figure 3: Évolution du nombre de femelles.

Dans ce dernier graphe, on peut observer le nombre de mâles et de femelles en fonction de leur âge. On constate un immense nombre de petits et on peut nettement voir le fait que très peu de petits survivent puisqu'il y a un très gros écart entre le nombre de petits et le nombre de lapins âgés de 1 ans.

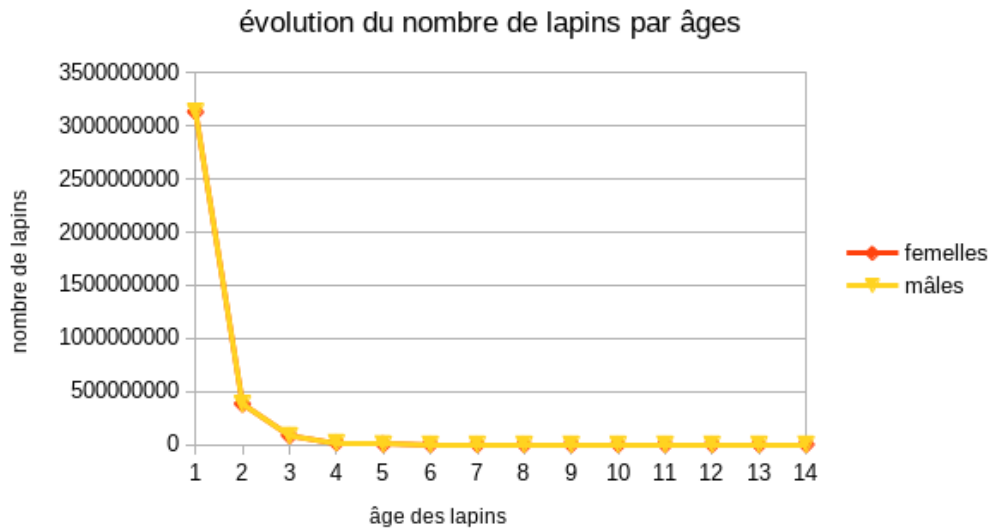


Figure 4: Évolution du nombre de lapins par âges la dernière année.

Enfin, ce dernier graphe nous montre aussi que le nombre de lapins survivants décroît extrêmement vite ce qui pourrait sembler illogique par rapport au fait que la population croît de façon exponentielle. On peut donc en conclure que la majorité des lapins sont toujours des jeunes et que la croissance de la population des lapins est uniquement due à leur capacité à se reproduire plutôt qu'à leur longévité.

## 4 Conclusion

Dans ce tp, on a pu observer que le modèle basique proposé par Fibonacci était peu réaliste et on a pu proposer un modèle plus complexe permettant de calculer beaucoup plus précisément le nombre de naissances et de morts. Cependant, bien que notre simulation soit plus réaliste, elle reste très éloignée de la réalité. Une simulation n'est jamais vraiment représentative de la réalité, mais elle peut s'en approcher, le modèle simulé peut toujours être complexifié pour gagner en précision. Par exemple nous aurions pu prendre comme pas le mois, ou même le jour. Nous aurions aussi pu rajouter d'autres événements causant la mort des lapins ou altérant leur capacité à se reproduire comme des épidémies, les mois de l'année (des morts de froid en hiver, ou de soif en été en cas de sécheresse).

Il faut tout de même noter le temps de calcul extrêmement long pour ce modèle, aussi simple soit-il. La limitation technique est à prendre en compte lorsqu'on décide de créer une simulation et il faut savoir garder sa simulation épurée de tout événement inintéressant qui pourrait engendrer des calculs inutiles. Il faut se concentrer sur ce que l'on souhaite observer.