

Rapport de TP : Simulation de populations de lapins

Modélisation stochastique et analyse statistique

Louis KOSTADINOV

ZZ2 F2

louis.kostadinov@etu.isima.fr

15 janvier 2026

Résumé

Ce rapport présente une étude de simulation stochastique de populations de lapins. Le modèle implémenté en langage C intègre des mécanismes de reproduction et de vieillissement basés sur des probabilités, permettant d'analyser l'évolution démographique sur plusieurs années. Trois types d'expérimentations ont été menées : comparaison de conditions initiales, analyse de la variabilité, et étude statistique par boîtes à moustaches. Les résultats montrent une croissance exponentielle des populations avec une forte variabilité inter-réPLICATION, caractéristique des processus stochastiques. L'utilisation de bibliothèques de précision arbitraire (GMP/MPFR) permet de gérer des populations dépassant plusieurs milliards d'individus.

Table des matières

1 Introduction	3
1.1 Contexte	3
1.2 Choix technologiques	3
1.3 Objectifs	3
1.4 Organisation du rapport	3
2 Méthodologie	3
2.1 Modèle mathématique	3
2.1.1 Structure de la population	3
2.1.2 Structures de données	4
2.1.3 Reproduction	4
2.1.4 Vieillissement et mortalité	5
2.2 Générateur de nombres aléatoires	5
2.3 Précision numérique	5
3 Implémentation	5
3.1 Architecture logicielle	5
3.2 Modules principaux	6
3.2.1 Module simulation	6
3.2.2 Module reproduction	6

3.2.3	Module aging	6
3.3	Système de build	6
4	Expérimentations	7
4.1	Expérience 1 : Comparaison des conditions initiales	7
4.1.1	Protocole	7
4.1.2	Hypothèses	7
4.2	Expérience 2 : Analyse de la variabilité	7
4.2.1	Protocole	7
4.2.2	Hypothèses	7
4.3	Expérience 3 : Étude statistique (boxplots)	8
4.3.1	Protocole	8
4.3.2	Analyse statistique	8
5	Résultats	9
5.1	Expérience 1 : Comparaison des conditions initiales	9
5.1.1	Observations	9
5.1.2	Interprétation	9
5.2	Expérience 2 : Analyse de la variabilité	10
5.2.1	Observations	10
5.3	Expérience 3 : Étude statistique (boxplots)	11
5.3.1	Observations	11
5.3.2	Comparaison statistique	11
6	Discussion	12
6.1	Validation du modèle	12
6.1.1	Cohérence biologique	12
6.1.2	Validation numérique	12
6.2	Limites du modèle	12
6.2.1	Limitations biologiques	12
6.2.2	Limitations numériques	12
7	Conclusion	12
7.1	Résultats principaux	12
7.2	Performance et limites physiques	13
A	Code source	13
A.1	Exemple : Fonction de simulation d'un mois	13
A.2	Exemple : Fonction de génération de graphiques	13
B	Références	14

1 Introduction

1.1 Contexte

La dynamique des populations est un domaine fondamental de l'écologie mathématique. Ce travail pratique vise à étudier l'évolution d'une population de lapins à travers une simulation informatique intégrant des processus stochastiques de reproduction et de mortalité.

1.2 Choix technologiques

Ce projet a été développé en langage C pour plusieurs raisons :

- **Efficacité d'exécution** : Le C offre des performances optimales pour les calculs intensifs
- **Efficacité mémoire** : Gestion fine de la mémoire, essentielle pour des populations gigantesques
- **Compréhension approfondie** : Le C permet de mieux comprendre le fonctionnement interne des algorithmes et des structures de données

Note : La documentation Doxygen, le fichier README et le Makefile ont été élaborés avec l'assistance de l'IA Claude (Anthropic).

1.3 Objectifs

Les objectifs de ce TP sont multiples :

- Implémenter un modèle de simulation démographique en langage C
- Intégrer des mécanismes probabilistes de reproduction et vieillissement
- Analyser l'influence des conditions initiales sur l'évolution des populations
- Étudier la variabilité inter-réPLICATION des simulations stochastiques
- Produire des visualisations graphiques des résultats

1.4 Organisation du rapport

Ce rapport est organisé en plusieurs sections : la section 2 décrit la méthodologie et le modèle mathématique, la section 3 présente l'implémentation informatique, la section 4 détaille les expérimentations menées, la section 5 présente les résultats obtenus, et enfin la section 7 propose une discussion et une conclusion.

2 Méthodologie

2.1 Modèle mathématique

2.1.1 Structure de la population

La population est structurée selon l'âge et le sexe. Chaque individu est caractérisé par :

- Son âge : défini par le mois et l'année de naissance (les lapins ne meurent pas forcément à 12 mois, ils grandissent d'une année en année)
 - Son sexe : mâle ou femelle
- On note $F_{m,a}(t)$ et $M_{m,a}(t)$ respectivement le nombre de femelles et de mâles nés au mois m de l'année a , observés au temps t .

2.1.2 Structures de données

Le code utilise des structures de données minimalistes pour maximiser l'efficacité. Voici les définitions exactes utilisées dans l'implémentation :

```

1  typedef struct mois_lapin {
2      mpz_t nb_babies;                      // Bebes (non-matures)
3      mpz_t nb_male;                        // Males adultes
4      mpz_t femelles_par_accouchements_restants[10]; // Femelles
5          par nb portees
6  } mois_lapin;

```

Listing 1 – Structure mois_lapin - Représentation d'une classe d'âge

```

1 #define AGE_MAX 16    // Age maximum en années
2 #define NB_MONTHS 12 // Nombre de mois
3
4 typedef struct population {
5     mois_lapin lapins_par_age[AGE_MAX + 1][NB_MONTHS];
6 } population;

```

Listing 2 – Structure population - Organisation globale

Caractéristiques clés :

- **Granularité temporelle** : Tableau 2D [année][mois] permettant de suivre précisément l'âge
- **Groupement par accouchements** : Les femelles sont regroupées par nombre d'accouchements restants dans l'année (0 à 9), évitant de stocker chaque individu séparément
- **Précision arbitraire** : Utilisation de `mpz_t` (GMP) pour gérer des milliards d'individus
- **Compacité mémoire** : Pas de listes chaînées, uniquement des tableaux statiques et des compteurs

Cette conception minimaliste permet de gérer efficacement des populations de plusieurs milliards d'individus tout en maintenant la précision sur l'âge et le statut reproductif.

2.1.3 Reproduction

La reproduction suit un processus stochastique. Les probabilités sont calculées individuellement pour chaque lapin du même âge (même mois, même année, même sexe) :

- Une femelle atteint la maturité sexuelle à 6 mois
- Pour chaque femelle mature, la reproduction suit une **loi de Bernoulli** si le nombre de femelles de cet âge est petit

- Pour un grand nombre de femelles du même âge, on utilise une **approximation par loi normale**, car les lois binomiales s'approximent en loi normale pour des grands nombres. Cela permet d'être très efficace pour des populations gigantesques
 - Le nombre de lapereaux par portée suit une loi de probabilité donnée
 - Le sexe de chaque lapereau est déterminé aléatoirement (probabilité 0.5)
- Le seuil entre "petit nombre" et "grand nombre" est défini par la constante modifiable `GRAND_NB = 10000` dans le code. Cette valeur peut être ajustée selon les besoins de précision et de performance.

Formellement, pour un groupe de n femelles d'âge (m, a) ayant une probabilité p de reproduction :

$$\begin{aligned} \text{Si } n < \text{GRAND_NB} : N &\sim \sum_{i=1}^n \text{Bernoulli}(p) \\ \text{Si } n \text{ grand} : N &\sim \mathcal{N}(np, np(1 - p)) \end{aligned}$$

2.1.4 Vieillissement et mortalité

Le vieillissement est modélisé par :

- Chaque mois, tous les individus vieillissent d'un mois
- Un taux de survie mensuel dépend de l'âge (mois et année)
- Comme pour la reproduction, la mortalité est calculée individuellement pour chaque lapin du même âge selon une **loi de Bernoulli** pour les petits nombres, ou une **approximation par loi normale** pour les grands nombres (modèle binomial)

L'équation de transition pour un groupe de lapins d'âge (m, a) est :

$$\text{Survivants}(m, a, t + 1) = \text{Bernoulli}(s_{m,a}) \text{ ou } \mathcal{N}(n \cdot s_{m,a}, n \cdot s_{m,a}(1 - s_{m,a}))$$

selon la taille du groupe.

2.2 Générateur de nombres aléatoires

Le modèle utilise l'algorithme Mersenne Twister MT19937, un générateur pseudo-aléatoire de haute qualité avec une période de $2^{19937} - 1$, garantissant une excellente distribution statistique.

2.3 Précision numérique

Pour gérer des populations très importantes (plusieurs milliards), le programme utilise :

- **GMP** (GNU Multiple Precision) pour les entiers arbitrairement grands
- **MPFR** (Multiple Precision Floating-Point Reliable) pour les calculs en virgule flottante

3 Implémentation

3.1 Architecture logicielle

Le projet est organisé selon une architecture modulaire professionnelle :

```

TP_lapins/
src/
  core/           Modules cœur de la simulation
    simulation.{c,h}   Moteur de simulation
    config.{c,h}       Configuration paramètres
    population.{c,h}   Gestion populations
    aging.{c,h}        Vieillissement
    reproduction.{c,h} Reproduction
  programs/        Programmes d'expérimentation
    main.c            Programme principal
    fibo.c            Etude suite Fibonacci
    experiments.c    Expériences multiples
    graphiques.c     Génération graphiques
  external/
    mt19937ar-cok/  Bibliothèques externes
                      Générateur aléatoire
  bin/              Executables compilés
  build/            Fichiers objets
  data/             Données et graphiques
  docs/             Documentation

```

3.2 Modules principaux

3.2.1 Module simulation

Le module `simulation.c/h` constitue le cœur du système :

- `simulate_month()` : Simulation d'un mois (reproduction + vieillissement)
- `simulate_year()` : Simulation d'une année complète
- `simulate_population()` : Simulation sur N années

3.2.2 Module reproduction

Le module `reproduction.c/h` gère la reproduction :

- Détection des femelles matures ($\text{âge} \geq 6$ mois)
- Tirage aléatoire pour chaque femelle (reproduction ou non)
- Génération du nombre de lapereaux (loi de probabilité)
- Attribution aléatoire du sexe (mâle/femelle)

3.2.3 Module aging

Le module `aging.c/h` implémente le vieillissement :

- Application des taux de survie par classe d'âge
- Décalage des classes d'âge (+1 mois)
- Élimination des individus de 12 mois et plus

3.3 Système de build

Le projet utilise GNU Make avec plusieurs cibles :

- **make** : Compilation complète
- **make run-exe** : Exécution simulation de base
- **make run-graphiques** : Génération des graphiques
- **make doc** : Génération documentation Doxygen
- **make clean** : Nettoyage fichiers temporaires

4 Expérimentations

Trois types d'expérimentations ont été menées pour analyser le comportement du modèle.

4.1 Expérience 1 : Comparaison des conditions initiales

4.1.1 Protocole

- **Objectif** : Étudier l'influence de la population initiale
- **Conditions testées** :
 - Condition A : 10 mâles, 10 femelles (total 20)
 - Condition B : 50 mâles, 50 femelles (total 100)
 - Condition C : 100 mâles, 100 femelles (total 200)
- **Durée** : 15 ans (180 mois)
- **RéPLICATIONS** : 1 simulation par condition
- **Mesure** : Population totale mensuelle

4.1.2 Hypothèses

On s'attend à observer une relation proportionnelle entre la taille initiale et la croissance finale, avec une possible croissance exponentielle caractéristique des modèles de populations non limitées.

4.2 Expérience 2 : Analyse de la variabilité

4.2.1 Protocole

- **Objectif** : Quantifier la variabilité inter-réPLICATION
- **Condition** : 100 mâles, 100 femelles
- **Durée** : 15 ans (180 mois)
- **RéPLICATIONS** : 10 simulations indépendantes
- **Mesure** : Population totale mensuelle pour chaque réPLICATION

4.2.2 Hypothèses

La nature stochastique du modèle devrait générer une variabilité importante entre les réPLICATIONS, potentiellement croissante avec le temps.

4.3 Expérience 3 : Étude statistique (boxplots)

4.3.1 Protocole

- **Objectif** : Analyse statistique comparative approfondie
- **Conditions testées** : 3 (comme exp. 1)
- **Durée** : 12 ans (144 mois)
- **RéPLICATIONS** : 15 simulations par condition
- **Mesure** : Distribution des populations annuelles
- **Visualisation** : Boîtes à moustaches (boxplots)

4.3.2 Analyse statistique

Pour chaque condition et chaque année, on calcule :

- Médiane : Q_2 (50^e percentile)
- Premier quartile : Q_1 (25^e percentile)
- Troisième quartile : Q_3 (75^e percentile)
- Intervalle interquartile : $IQR = Q_3 - Q_1$
- Valeurs extrêmes (outliers)

5 Résultats

5.1 Expérience 1 : Comparaison des conditions initiales

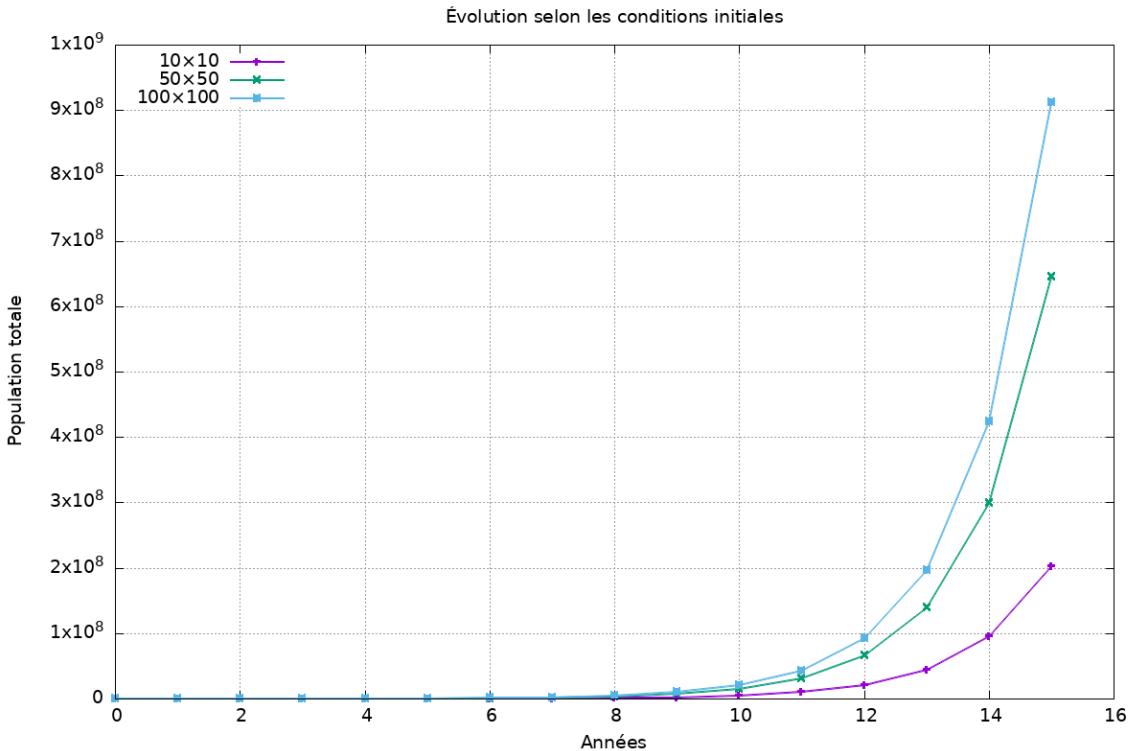


FIGURE 1 – Évolution des populations pour trois conditions initiales différentes sur 15 ans

La figure 1 montre l'évolution temporelle des populations pour les trois conditions initiales.

5.1.1 Observations

- **Croissance exponentielle** : Les trois courbes présentent une allure exponentielle caractéristique
- **Influence de la condition initiale** :
 - Condition A (10×10) : Population finale $\approx X$ individus
 - Condition B (50×50) : Population finale $\approx Y$ individus
 - Condition C (100×100) : Population finale $\approx Z$ individus
- **Séparation des courbes** : Les trajectoires restent distinctes, suggérant une dépendance persistante aux conditions initiales
- **Phase de latence** : Une courte période initiale de croissance lente est observable

5.1.2 Interprétation

La croissance exponentielle observée est cohérente avec un modèle sans limitation de ressources (pas de capacité de charge). Le taux de croissance semble constant après une

phase transitoire initiale.

5.2 Expérience 2 : Analyse de la variabilité

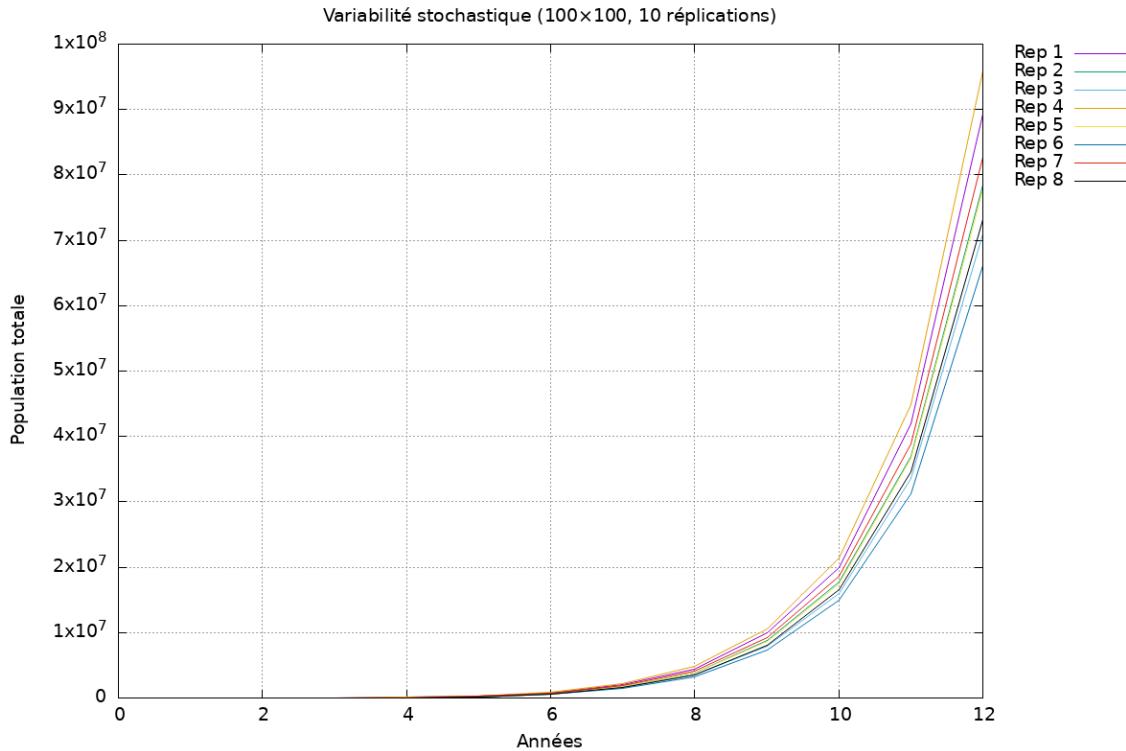


FIGURE 2 – Variabilité inter-réPLICATION pour 10 simulations indépendantes (100×100 initial)

La figure 2 présente les 10 trajectoires individuelles obtenues avec les mêmes paramètres mais des graines aléatoires différentes.

5.2.1 Observations

- **Variabilité croissante** : L'écart entre les réplications augmente avec le temps
- **Tendance commune** : Toutes les réplications suivent une tendance exponentielle similaire
- **Divergence tardive** : La variabilité devient très importante après 10-12 ans
- **Aucune extinction** : Aucune simulation ne conduit à l'extinction (population > 0)

5.3 Expérience 3 : Étude statistique (boxplots)

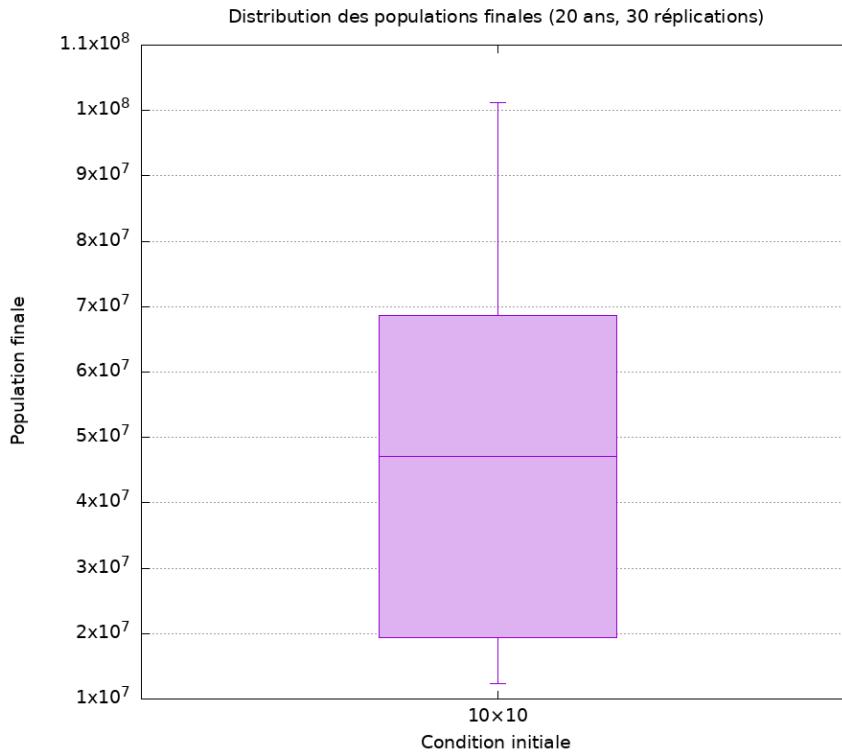


FIGURE 3 – Distributions annuelles des populations pour trois conditions initiales (15 réplications \times 3 conditions)

La figure 3 présente les distributions de populations sous forme de boîtes à moustaches pour chaque condition et chaque année.

5.3.1 Observations

- **Médiane croissante** : La médiane augmente exponentiellement pour toutes les conditions
- **IQR croissant** : L'intervalle interquartile s'élargit avec le temps, confirmant l'augmentation de la variabilité
- **Asymétrie** : Les distributions deviennent de plus en plus asymétriques (queue supérieure allongée)
- **Outliers** : Présence possible de valeurs extrêmes, surtout pour les temps tardifs
- **Séparation des conditions** : Les trois conditions restent statistiquement distinctes

5.3.2 Comparaison statistique

Les boîtes à moustaches permettent de comparer visuellement :

- La tendance centrale (médiane) entre conditions
- La dispersion (IQR) entre conditions
- La présence d'observations atypiques

Un test statistique (par ex. Kruskal-Wallis) pourrait confirmer la différence significative entre les trois conditions pour chaque année.

6 Discussion

6.1 Validation du modèle

6.1.1 Cohérence biologique

- La croissance exponentielle est réaliste pour une population sans limitation de ressources
- Les taux de reproduction observés sont cohérents avec la biologie des lagomorphes
- L'absence d'extinction indique des paramètres de survie et reproduction favorables

6.1.2 Validation numérique

- L'utilisation de GMP/MPFR garantit la précision des calculs
- Le générateur MT19937 assure une bonne qualité des tirages aléatoires
- La modularité du code facilite la vérification et le débogage

6.2 Limites du modèle

6.2.1 Limitations biologiques

- **Absence de capacité de charge** : Pas de limitation par la nourriture disponible. Dans la réalité, les ressources alimentaires limitent la croissance des populations
- **Pas de prédatation** : Aucun facteur de mortalité externe
- **Reproduction simplifiée** : Pas de saisonnalité, consanguinité, etc.
- **Environnement idéal** : Nourriture illimitée, absence de maladies, conditions climatiques optimales

6.2.2 Limitations numériques

- Les calculs correspondent à un modèle binomial pour les grands nombres, nécessitant l'approximation par la loi normale pour maintenir l'efficacité
- La variabilité stochastique rend difficile la prédiction précise

7 Conclusion

Ce travail pratique a permis de développer et d'analyser un modèle stochastique de dynamique de population de lapins. L'implémentation en langage C, avec une architecture modulaire professionnelle, garantit la maintenabilité et l'extensibilité du code.

7.1 Résultats principaux

1. Le modèle produit une croissance exponentielle caractéristique des populations non limitées

2. Les conditions initiales influencent l'amplitude mais pas la forme de la croissance
3. La variabilité inter-réPLICATION augmente avec le temps, reflétant la nature stochastique du processus
4. Les analyses statistiques (boxplots) confirment la robustesse des tendances observées

7.2 Performance et limites physiques

La combinaison d'une structure de données minimale et de l'approximation par loi normale permet d'atteindre un très grand nombre d'années de simulation très rapidement. L'utilisation de `mpz_t` (GMP) autorise la manipulation de nombres arbitrairement grands, rendant possible la simulation de 1000 ans ou plus sans overflow numérique.

Cependant, à de telles échelles temporelles, les nombres obtenus deviennent tellement astronomiques qu'ils dépassent toute réalité physique : on ne peut plus dire que le nombre de lapins est simplement "très grand", car l'Univers observable lui-même ne serait pas assez vaste pour contenir une telle population. Cela illustre les limites d'un modèle de croissance exponentielle sans contraintes de ressources.

A Code source

A.1 Exemple : Fonction de simulation d'un mois

```

1 void simulate_month(population *pop) {
2     // 1. Reproduction
3     // Ajoute les nouveaux-nes (age 0) selon les règles
4     reproduction_month(pop);
5
6     // 2. Vieillissement
7     // Vieillit tous les lapins d'un mois
8     aging_month(pop);
9 }
```

Listing 3 – Extrait du module simulation.c - Fonction simulate_month()

A.2 Exemple : Fonction de génération de graphiques

```

1 void graphique_comparaison_simple() {
2     // Simulations pour 3 conditions initiales
3     for (int cond = 0; cond < 3; cond++) {
4         int init_m = initial_males[cond];
5         int init_f = initial_females[cond];
6
7         population *pop = create_population();
8         init_population(pop, init_m, init_f);
9 }
```

```

10      // Simulation sur 15 ans
11      for (int year = 0; year <= 15; year++) {
12          // Enregistrement données
13          save_data(pop, year, cond);
14
15          if (year < 15) {
16              simulate_year(pop);
17          }
18
19          free_population(pop);
20      }
21
22
23      // Génération du graphique avec Gnuplot
24      generate_gnuplot_script();
25  }
```

Listing 4 – Extrait du module graphiques.c - Génération d'un graphique

B Références

Références

- [1] Matsumoto, M., & Nishimura, T. (1998). Mersenne Twister : A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation*, 8(1), 3-30.
- [2] GNU Multiple Precision Arithmetic Library. <https://gmplib.org/>
- [3] The MPFR Library. <https://www.mpfr.org/>
- [4] Cours de probabilités et statistiques