ATCHOUM ou BLABLA

Année 2014- 2015

CAMBUS Paul, DJEBAR Vincent, élèves de première scientifique

Lycée Jean Lurçat , Perpignan, jumelé avec le lycée Arago, Perpignan

Enseignant-e-s : Pascal Berthe, Eric Moccand, Martine Vergnac.

Chercheur : Robert Brouzet, Université de Perpignan (U.P.V.D)

Enoncé du sujet :
Une maladie se propage dans une population constante de la manière suivante : la population est partagée en deux groupes d'individus, ceux qui ne sont pas malades et ceux qui sont malades . On suppose qu'à chaque instant ces individus peuvent être amenés à se rencontrer avec, lors d'une rencontre, une certaine probabilité de transmission de la maladie.
Comment va diffuser la maladie au sein de la population ?

Ainsi, on cherche à étudier l'évolution de la population malade parmi la population totale ou autrement dit comment va évoluer la propagation de la maladie et donc des contaminés parmi la population.

Le sujet clairement analysé et identifié, nous avons pu commencé à réfléchir à quelques idées pour répondre à cette problématique. Après quelques semaines de réflexion et d'échanges, c'est finalement vers un modèle simplifié que nous nous orientons afin d'apporter une éventuelle solution.

Nous verrons dans un premier temps comment nous avons interprété cette situation par un modèle afin de conduire des expérimentations sur des populations variables, puis dans un second temps, nous présenterons le nouveau modèle mathématique théorique élaboré afin de généraliser l'évolution de la population malade.
De cette manière, nous serons en mesure de montrer comment nous avons mis en évidence les trois phases principales et universelles d'évolution de la population contaminée, indépendantes des paramètres initiaux.

I – Construction d'un modèle

a) Nous choisissons donc de modéliser la population par des boules de couleur disposées dans une urne. On considère par exemple que les personnes saines seront représentées par des boules vertes et les personnes malades représentées par des boules rouges.
Ainsi, au commencement de l'épidémie, on est en présence d'une urne contenant autant de boules que de personnes :
- des vertes pour les saines,
- et des rouges pour les malades.

b) On va dans ce modèle découper le temps en instants précis selon un modèle discret. Ainsi, *l'instant n* sera un nombre entier. Cette unité représentera une unité de temps, un instant, qui pourra éventuellement se traduire plus tard par n'importe quelle unité de temps appropriée (millisecondes, secondes, minutes,...). Chaque *instant n*, correspondra dans notre modèle à une rencontre aléatoire entre deux personnes, donc à **un tirage** dans l'urne selon les conditions ci dessous :

A chaque *instant n*, on va simuler une rencontre aléatoire entre deux personnes en effectuant un tirage aléatoire dans l'urne de deux boules. Par conséquent, à chaque *instant n*, on tire simultanément deux boules dans l'urne **au hasard** et on compare leur couleur :
- Si on obtient deux boules de couleur identique, alors il ne se passe rien, on remet les deux boules de couleur dans l'urne sans changement : cela signifie dans notre modèle qu'une personne saine a rencontré une personne saine OU qu'une personne malade a rencontré une personne malade ; il n'y a donc pas de nouvelle contamination possible.
- Si on obtient deux boules de couleur différente (c'est à dire une boule verte et une boule rouge), alors la boule rouge "contamine" la boule verte. On remplace la boule verte par une boule rouge et on remet les deux boules rouges dans l'urne pour le prochain tirage. Cela signifie dans notre modèle qu'une personne saine a rencontré une personne malade, et que la personne malade a contaminé la personne saine qui est donc à son tour devenue malade.

c) On va considérer dans notre modèle simplifié que **la probabilité de transmission de la maladie est de 1**, c'est à dire qu'elle est transmise à tous les coups lorsqu'un malade rencontre un non-malade.

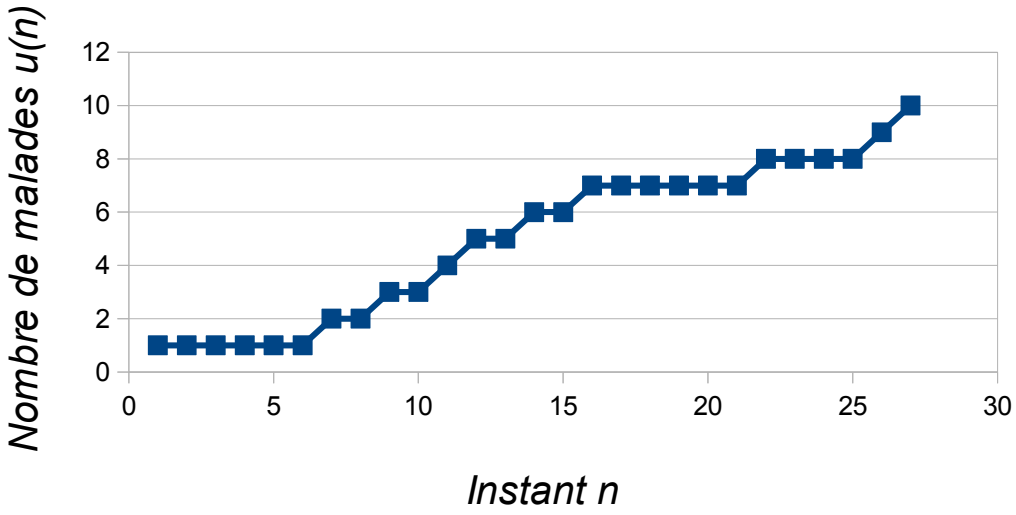
d) La seconde spécificité que l'on peut noter est qu'à chaque *instant n*, deux personnes **SEULEMENT** se rencontrent. C'est donc une limite importante mais si l'on considère l'instant n comme une unité de temps assez petite : des millisecondes par exemple, le modèle peut être vraisemblable car deux personnes se rencontrent systématiquement à chaque milliseconde. **(1)**

II – Premières constatations

Avec ce modèle simple élaboré, nous voulons obtenir de premiers résultats et décidons d'observer comment se comporte la population malade si l'on répète plusieurs tirages successifs.
Ainsi, munis d'un couvercle retourné pour faire office d'urne, d'une vingtaine de légos verts et rouges, nous commençons l'expérience et les tirages successifs en remplaçant à chaque fois manuellement les légos, et en notant à l'issue de chaque tirage la population de "légos rouges" ou de "malades".

Le test est lancé avec une population totale constante de 10 personnes dont 1 malade de départ, soit 9 légos verts et 1 légo rouge dans l'urne au commencement.

Fig 1 : Evolution de la population malade sur 10 personnes



Ce graphique traduit l'évolution de la population malade obtenue ; comme on peut le voir 27 tirages environ ont été nécessaires pour contaminer l'ensemble de la population.

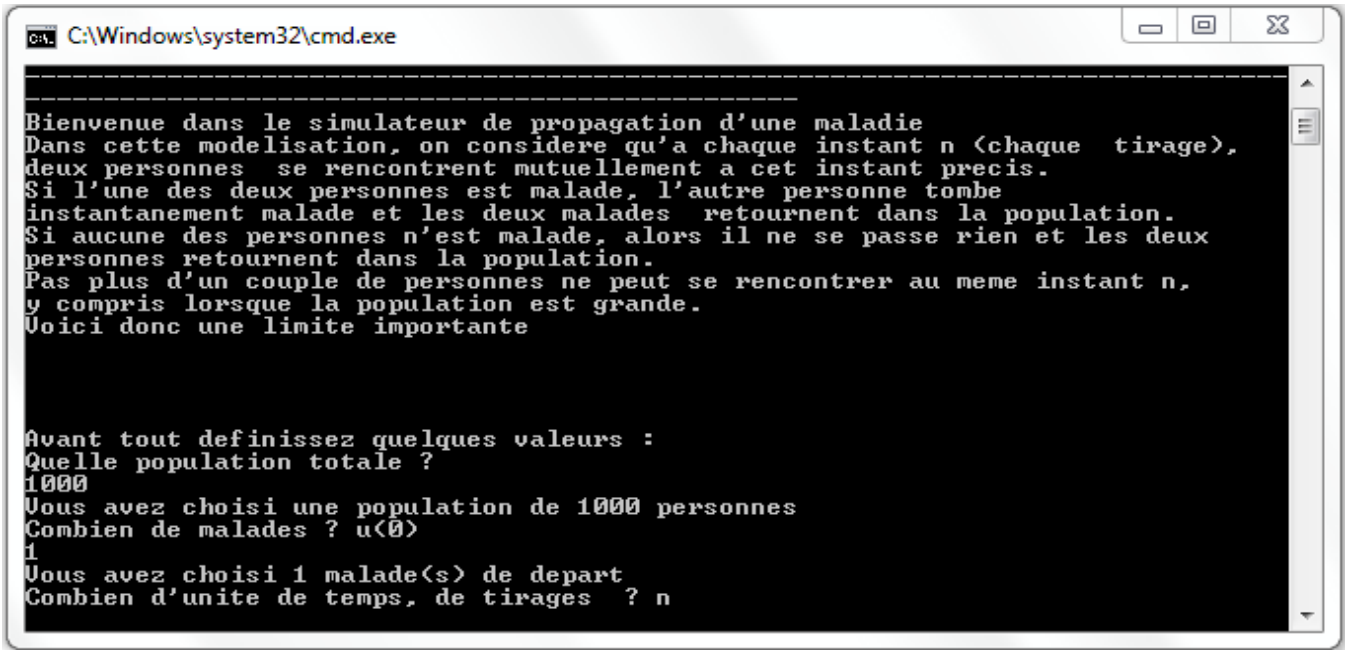
Ce premier résultat nous donne une première idée de la tendance générale mais ne nous permet pas de conclure quoi que ce soit. En effet, chaque expérience est différente car les tirages sont effectués au hasard. Par conséquent, il faudrait répéter cette même expérience plusieurs dizaines de fois et comparer les graphiques obtenus pour émettre nos premières conjectures.
Mais réaliser une expérience de ce type à la main prend du temps, et est assez fastidieux. De plus, on est limité par des contraintes matérielles et humaines : on ne peut en effet pas tester ce modèle sur une population importante comme 1000 personnes de départ, et encore moins la répéter X fois pour s'assurer du résultat.

Face à ce problème, une solution s'impose : réaliser un programme informatique qui effectue ces expériences virtuellement.
En effet, par la réalisation d'un programme informatique, nous pourrions en un instant générer les résultats de centaines de tirages, modifier en une seconde les modalités de l'expérience (nombre de malades de départ, population totale, nombre de tirages à effectuer), et surtout la répéter plusieurs fois pour s'assurer que les résultats ne sont pas erronés.

Nous décidons donc d'écrire un tel programme en JAVA, qui se trouve en annexe 1 et dont nous donnons ci dessous une explication succincte.

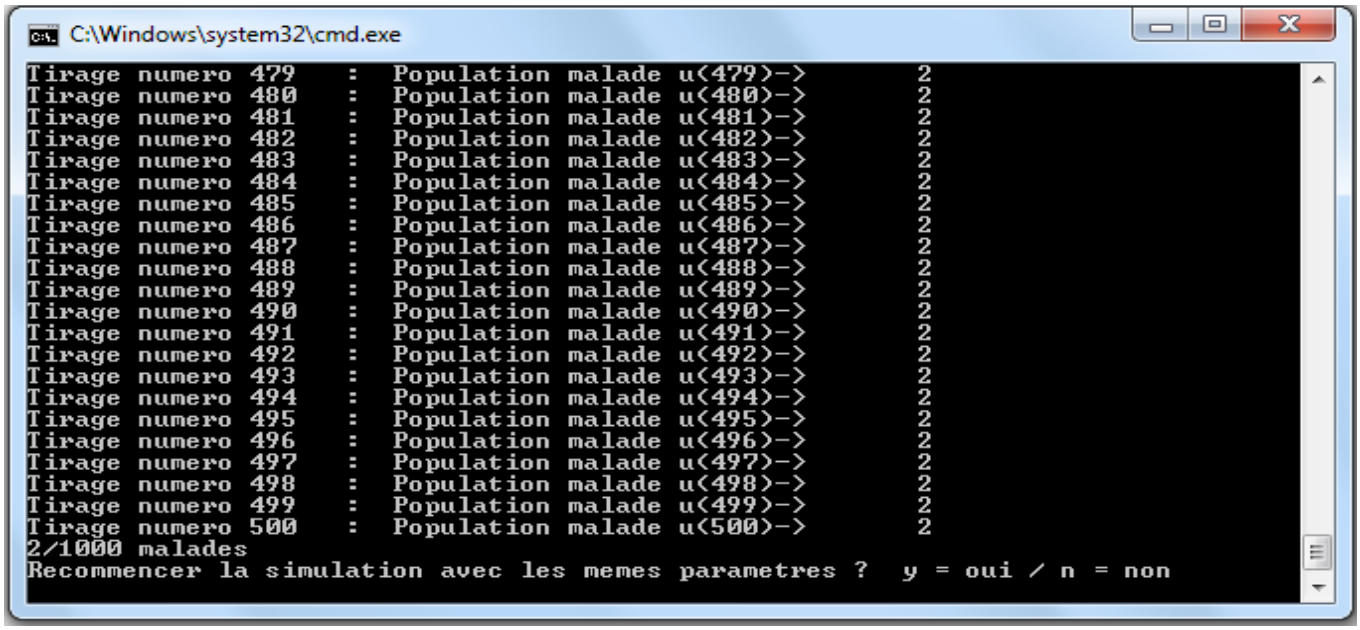
Après avoir entré les valeurs souhaitées pour la population totale, le nombre de malades initial et le nombre de tirages, le programme créera une urne virtuelle de la taille souhaitée en y disposant le nombre de malades de départ indiqué.

Fig 2 : Entrée des valeurs initiales dans le programme



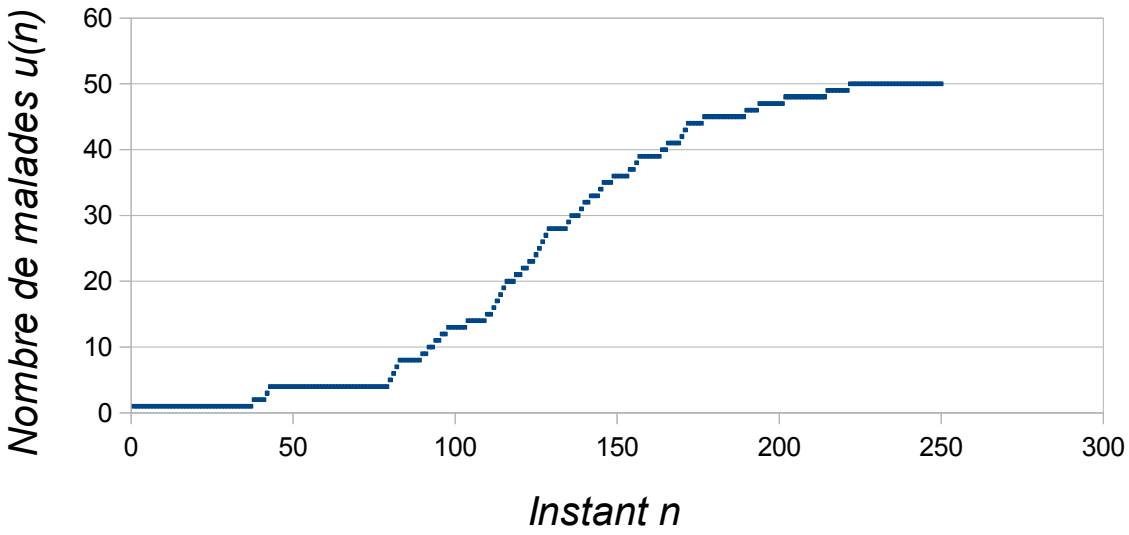
Par la suite, chaque tirage/rencontre sera simulé un à un, en appelant la fonction mathématique *Math.random()* pour imiter la "pioche" dans l'urne au hasard, et entrainera les modifications qui s'imposent, en fonction des "boules" piochées (voir l'annexe 1 pour plus de détails). A la fin de l'opération, le programme affiche le nombre de malades contaminés sur la population totale.

Fig 3 : Tirages simulant les rencontres aléatoires et résultat final



Grâce à ce code, on est en mesure d'obtenir un programme en console de commandes qui fonctionne et nous permet d'effectuer autant d'expériences que l'on veut avec les variables que l'on veut. Les différents tests effectués informatiquement nous ont permis de confirmer la tendance générale dégagée par le premier test manuel. Voici par exemple le graphique obtenu sur une population de 50 personnes avec 1 malade de départ :

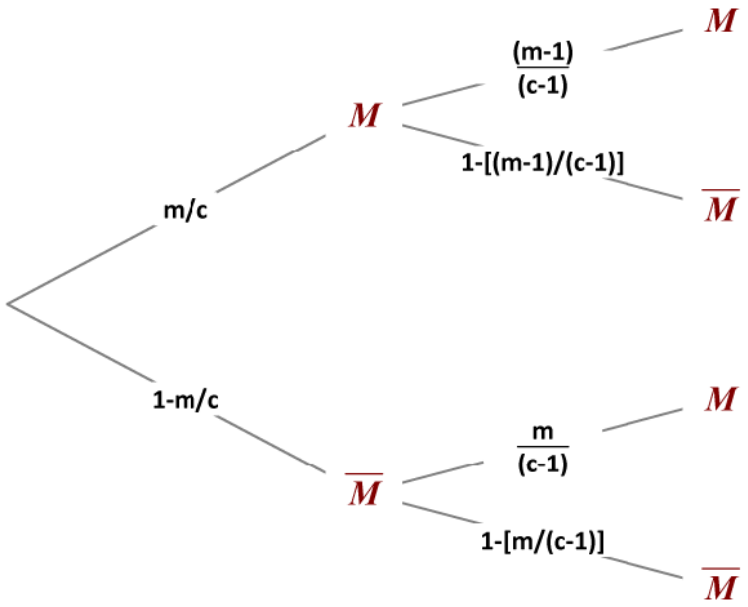
Fig 4 : Evolution de la population de malades sur 50 personnes



III – Généralisation des résultats

Maintenant que nous avons une idée de l'évolution qui se précise, nous cherchons à généraliser ce résultat suivant n'importe quelles valeurs. Nous décidons pour cela de modéliser l'expérience aléatoire du tirage sous la forme d'un arbre de probabilités :

Fig 5 : Arbre de probabilités modélisant la situation de pioche dans l'urne



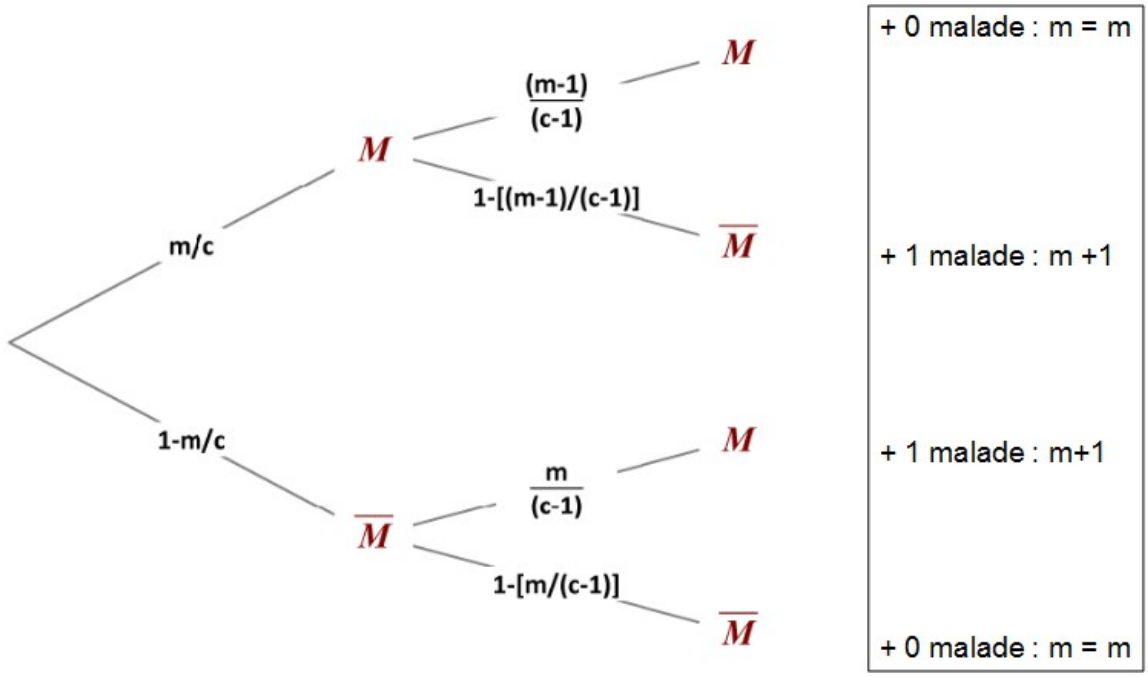
- L'événement M correspond à l'action "On tire une boule rouge (personne malade)"
- m = population de malades à cet instant
- c = population totale constante

Le tirage simultané des deux boules peut être représenté ainsi car il est équivalent à deux tirages successifs **sans remise** des boules.

A partir de cette représentation, il est aisé de déduire les chemins qui mènent à l'apparition d'un malade de plus, à une contamination. Ce sont les chemins qui font se succéder les événements M puis Mbarre OU Mbarre puis M.

Ces chemins correspondent en fait à la situation suivante :
- On tire un malade et un non-malade => Il y a donc contamination
- On tire un non-malade et un malade => Il y a donc contamination

Fig 6 : Arbre de probabilités modélisant les résultats possibles à l'issue du tirage



Connaissant les chemins menant à l'apparition d'un nouveau malade, il nous est possible de déduire la probabilité qu'un nouveau malade apparaisse à un instant précis où on a *m malades* :

$$P(m+1) = m/c * [1-(m-1)/(c-1)] + (1-m/c) * m/(c-1)$$

$$P(m+1) = 2m(c-m) / c(c-1)$$

Par la suite, si l'on considère la variable aléatoire X qui prend **la valeur du nombre de malades à l'instant suivant**, il devient possible d'établir la loi de probabilité de X suivante :

Fig 7 : Loi de probabilité de la variable aléatoire associée au nombre de malades à l'instant suivant

x_i	m + 1	m
$p(X=x_i)$	P = 2m(c-m) / c(c-1)	1 - P

Afin de connaitre une approximation du nombre de malades à l'instant suivant d'après ces probabilités, il ne reste plus qu'à calculer l'espérance de X :

Calcul de l'espérance

$$E(X) = [2m(c-m) / c(c-1)]*(m+1) + (1-P)*m$$

$$\mathbf{E(X) = m(c^2+c-2m) / c(c-1)}$$

Grâce à cette espérance, on connait **la moyenne du nombre de malades que l'on peut espérer avoir à l'instant n+1, lorsque à l'instant n on avait m malades**.

Or, on peut aussi s'apercevoir que cette formule de l'espérance repose sur un raisonnement par récurrence (elle dépend du nombre de malades de l'instant d'avant) ; ainsi, on peut facilement la traduire par une suite avec relation de récurrence.

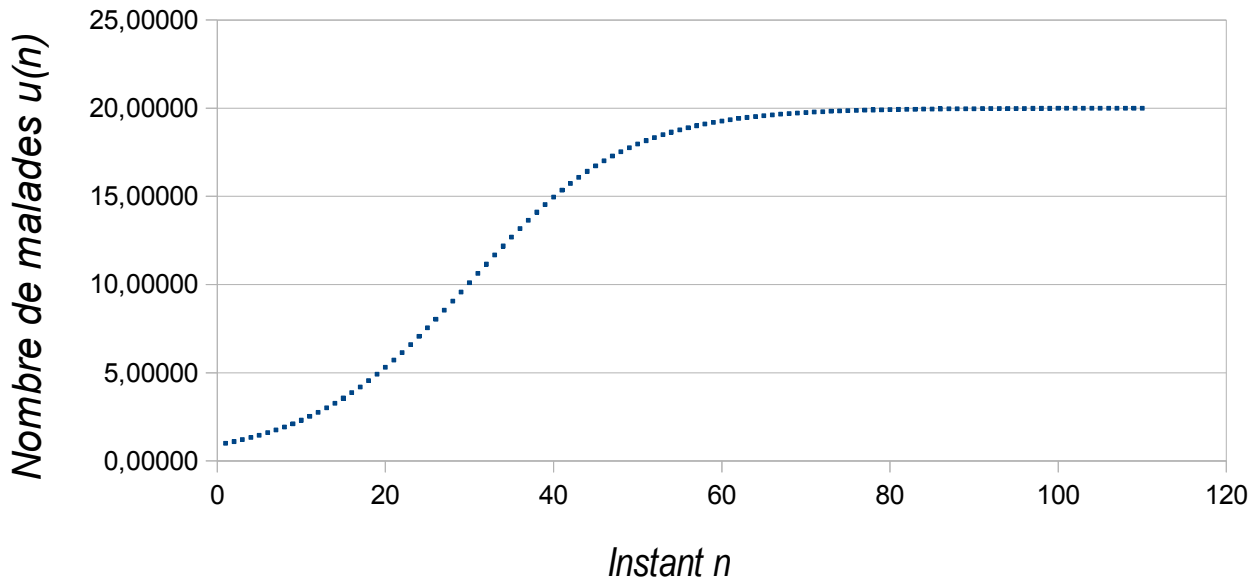
On va donc noter **u(n)** le nombre de malades **à l'instant n** afin que la suite u ainsi définie vérifie la relation de récurrence qui suit :

$$\mathbf{u(n+1) = u(n)(c^2+c-2u(n)) / c(c-1)}$$

Grâce à cette nouvelle suite, il est désormais plus facile d'observer l'évolution de la population de malades sur n'importe quelle population totale, en définissant simplement le nombre de malades de départ u(0).

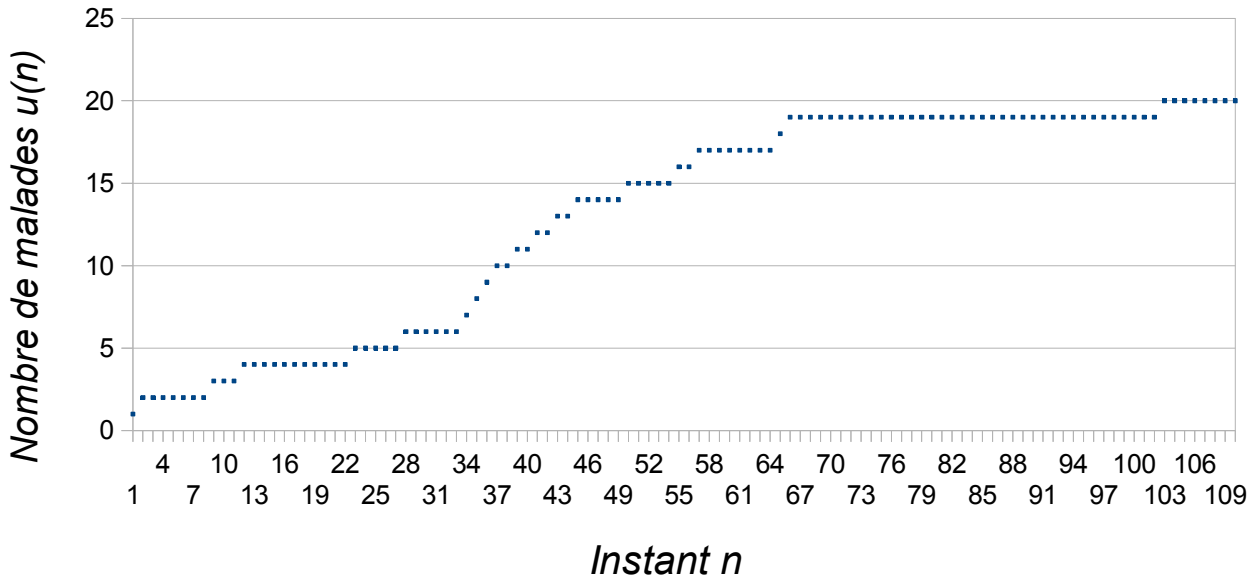
Voici donc l'évolution théorique d'une population de malades sur 20 personnes obtenue par ce raisonnement :

Fig 8 : Evolution théorique de la population de malades sur 20 personnes



On peut comparer ce résultat avec les résultats obtenus expérimentalement lors d'une simulation effectuée à l'aide du programme avec les mêmes paramètres initiaux :

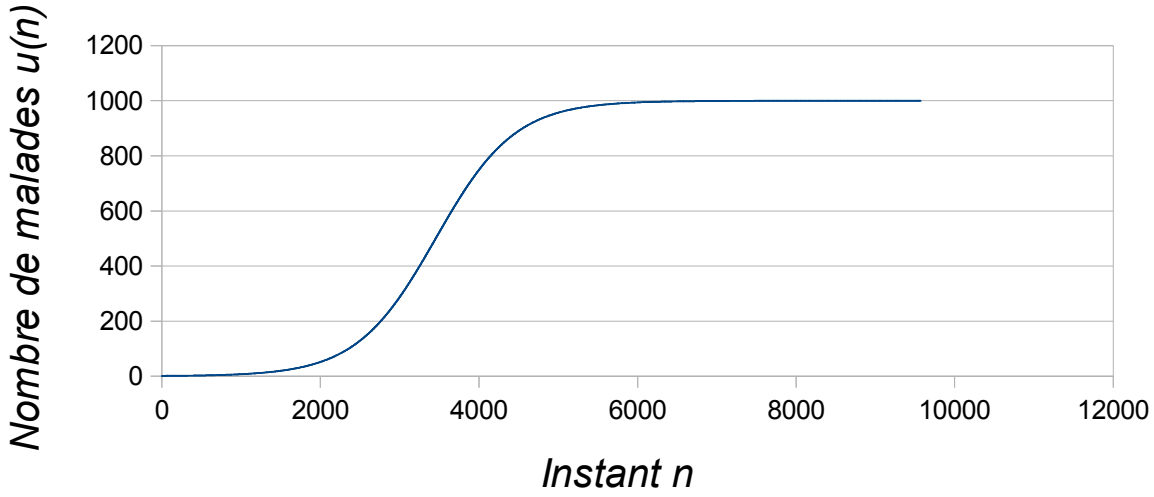
Fig 9 : Evolution expérimentale de la population de malades sur 20 personnes



On constate que la tendance qui se dégage est bien la même, nos résultats sont cohérents. Notre modèle semble donc se valider.

Voyons maintenant l'évolution théorique avec une population de 1000 personnes dont 1 malade de départ :

Fig 10 : Evolution théorique de la population de malades sur 1000 personnes



Il faut environ 9547 tirages pour contaminer la population totale !

IV – Conclusion

Ainsi, notre étude touche à sa fin et nous avons pu apporter une réponse à notre problématique en mettant en évidence l'évolution typique d'une population de malades grâce à un modèle simplifié.
On remarque donc que quelle que soit la population totale, la propagation de la maladie se divise en trois grands stades :

- **L'amorce** : Il n'y a au départ qu'un seul malade parmi toute la population ce qui le rend dans notre modèle très difficile à "piocher" dans l'urne. La contamination est donc lente au départ car ce ne sont principalement que des personnes saines qui se rencontrent.
- **La contamination s'accélère** : Au fur et à mesure qu'il y a de plus en plus de malades dans la population totale, de plus en plus de malades sont piochés et sont amenés à rencontrer des personnes saines, ce qui a pour conséquence d'accélérer grandement la propagation de la maladie.
- **Le ralentissement** : Finalement, lorsque presque toute la population est contaminée, c'est l'inverse de "l'amorce" qui se produit. Dans notre modèle, uniquement des personnes malades sont piochées dans l'urne et se rencontrent ce qui ne change rien, et les quelques personnes saines restantes sont très difficiles à piocher car très peu nombreuses. Cela explique donc pourquoi le dernier survivant est le plus long à contaminer.

Cependant, il est bien évidemment possible de compliquer ce modèle en ajoutant de nouvelles contraintes comme : une guérison au bout d'un certain temps ou encore une mort de certains malades,...
Tout cela représente autant de perspectives d'amélioration du modèle, aussi intéressantes les unes que les autres qui permettront de prolonger cette recherche en modifiant les conditions que nous avons posées initialement.

ANNEXE 1 : LE PROGRAMME JAVA

```
package programme;

import java.util.Scanner;

public class Frame {

 public static void main(String[] args){

   boolean restart =true;

   //VARIABLES

   int c = 50; //Population totale
   int m = 1; //Population de malades u(n)
   int  s = c - m; //Population de personnes saines.
   int n = 20; //Nombre d'unité de temps, de tirages

   //ENTREE DES VALEURS PAR L'UTILISATEUR

   Scanner sc = new Scanner(System.in);
   System.out.println("<Simulateur de tirages de couples  avec remise et contamination pour recherches mathematiques - Programme en langage JAVA  par Vincent Djebar le 18/11/2014 ->");
   System.out.println("----------------------------------------------------------------------------------------------------------------------------");
   System.out.println("Bienvenue dans le simulateur de propagation d'une maladie");
   System.out.println("Dans cette modelisation, on considere qu'a chaque instant n (chaque  tirage), \ndeux personnes  se rencontrent mutuellement a cet instant precis.");
   System.out.println("Si l'une des deux personnes est malade, l'autre personne tombe \ninstantanement malade et les deux malades  retournent dans la population."
       + "\nSi aucune des personnes n'est malade, alors il ne se passe rien et les deux \npersonnes retournent dans la population. ");
   System.out.println("Pas plus d'un couple de personnes ne peut se rencontrer au meme instant n, \ny compris lorsque la population est grande. \nVoici donc une limite importante ");
   System.out.println("\n \n \n");
   System.out.println("Avant tout definissez quelques valeurs :");
   System.out.println("Quelle population totale ?");
   int c1 = (int)sc.nextInt();
   sc.nextLine();

    System.out.println("Vous avez choisi une population de "+c1+" personnes");
    System.out.println("Combien de malades ? u(0) ");
    int m1 = (int)sc.nextInt();
    sc.nextLine();

    System.out.println("Vous avez choisi "+m1+" malade(s) de depart");
    System.out.println("Combien d'unite de temps, de tirages  ? n ");
    int n1 = (int)sc.nextInt();
    sc.nextLine();

    System.out.println("Vous avez choisi "+n1+" tirages");

    while(restart){

            c = c1;
            m = m1;
            n = n1;
            ///FORMATION DE LA POPULATION DANS UN TABLEAU
            int pop[] = new int[c] ;
            // 0 = personne saine
            // 1 = personne malade

            for(int i = 0; i < c;i++){
              pop[i] = 0; //Toutes les personnes sont saines
            }
            for(int i = 0; i < m;i++){
              double  ya = Math.random()*pop.length;
              int y = (int) ya;
              //System.out.println(pop.length);

              if(pop[y] == 0){ //MECANISME QUI ASSURE QUE LES MALADES SONT TOUJOURS DIFFERENTS POUR RESPECTER LE BON NOMBRE
                //System.out.println(y+" est la personne contaminee");
                pop[y] = 1;  //Ajout de la ou des personnes malades aléatoirement
```

```
        }
        else{
          //System.out.println(y+" est déjà contaminee");
          i--;
        }
      }

      System.out.println("Debut de la procedure");

      //TIRAGE DANS URNE DE DEUX PERSONNES ALEATOIRES AVEC REMISE
      for(int na =0; na <n; na++){

        /// On choisit deux personnes au hasard

        // PERSONNE 1
        double pers1a =Math.random()*pop.length;   // On génère un nombre aléatoire entre 0 et la taille de la population
        int pers1 =(int) pers1a; // On garde la partie entière de ce nombre généré

        // PERSONNE 2
        double pers2a = Math.random()*pop.length;
        int pers2 =(int) pers2a;


        /// On compare la nature des deux personnes. Si l'une est malade, l'autre le devient aussi
        // 0 =personne saine          1 =personne malade
        if(pop[pers1] ==0 &&pop[pers2]==1){
          pop[pers1] =1;
        }
        else if(pop[pers2] ==0 && pop[pers1] ==1)    {
          pop[pers2] =1;
        }
        m =0;
        for(int a=0;a<pop.length;a++){ //Décompte du nbr de malades
            if(pop[a] ==1) m++;
          }
          if(na+1 <10) System.out.println(            "Tirage numero "+(na+1)+"     : Population malade u("+(na+1)+")->        "+m);
          else if(na+1<100) System.out.println(       "Tirage numero "+(na+1 )+"    : Population malade u("+(na+1 )+")->       "+m);
          else if(na+1<1000) System.out.println(      "Tirage numero "+(na+1  )+"   : Population malade u("+(na+1  )+")->      "+m);
          else if(na+1<10000) System.out.println(     "Tirage numero "+(na+1   )+" : Population malade u("+(na+1   )+")->    "+m);
          else   System.out.println(                  "Tirage numero "+(na+1   )+": Population malade u("+(na+1   )+")->     "+m);

        }

        System.out.println(m+"/"+c+ " malades");

        //GESTION DE LA FIN DE PARTIE RECOMMENCER OU QUITTER
        System.out.println("Recommencer la simulation avec les memes parametres ?  y =oui / n =non");
        if(sc.nextLine().equals("y")){
          restart =true;
        }else{
          System.out.println("Parametrer une nouvelle simulation ?  y =oui / n =non");
          if(sc.nextLine().equals("y")){
            restart =true;
            System.out.println("Quelle population totale ?");
             c1 =(int)sc.nextInt();
            sc.nextLine();

            System.out.println("Vous avez choisi une population de "+c1+" personnes");
            System.out.println("Combien de malades ? u(0) ");
             m1 =(int)sc.nextInt();
            sc.nextLine();

            System.out.println("Vous avez choisi "+m1+" malade(s) de depart");
            System.out.println("Combien d'unite de temps, de tirages  ? n ");
             n1 =(int)sc.nextInt();
            sc.nextLine();
```

```
133
134                 System.out.println("Vous avez choisi "+n1+" tirages");
135               }
136               else{
137                 restart =false;
138               }
139
140             }
141     //return;
142     }
143   }
144 }
145
```

Voici un petit détail expliquant comme fonctionne ce programme :

Tout d'abord, voici comment fonctionne la génération de la population totale avec le bon nombre de malades, la génération de "l'urne" :
- Il génère en fait un tableau indicé de valeurs contenant autant de valeurs que le nombre de personnes dans la population totale. Chaque valeur est accessible via son indice, comme si chaque valeur se trouvait dans une "case numérotée".
- Dans chaque "case" de ce tableau on affecte la valeur 0, qui signifie que la personne à l'indice X est saine.
- Puis on choisit aléatoirement une case de ce tableau, (en générant un indice de façon aléatoire), et on change sa valeur de 0 à 1 ; qui signifie que la personne à cet indice devient malade. On répète cette opération autant de fois qu'il faut placer de malades dans la population totale saine.
Une fois la population prête, il ne reste plus qu'à effectuer *n tirages* selon le modèle que l'on a défini :

```
//TIRAGE DANS URNE DE DEUX PERSONNES ALEATOIRES AVEC REMISE
for(int na =0; na <n; na++){

  /// On choisit deux personnes au hasard

  // PERSONNE 1
  double pers1a =Math.random()*pop.length;   // On génère un nombre aléatoire entre 0 et la taille de la population
  int pers1 =(int) pers1a; // On garde la partie entière de ce nombre généré

  // PERSONNE 2
  double pers2a = Math.random()*pop.length;
  int pers2 =(int) pers2a;


  /// On compare la nature des deux personnes. Si l'une est malade, l'autre le devient aussi
  // 0 =personne saine          1 =personne malade
  if(pop[pers1] ==0 &&pop[pers2]==1){
    pop[pers1] =1;
  }
  else if(pop[pers2] ==0 && pop[pers1] ==1)    {
    pop[pers2] =1;
  }
```

Le tirage fonctionne de cette manière :

- On génère deux indices aléatoires, puis on vient comparer les valeurs du tableau (qui se trouvent à ces indices) entre elles. Si l'on a un 0 et un 1 dans les cases aux indices générés, qui signifie donc une personne saine et une personne malade, on transforme le 0 en 1 (la personne saine devient malade).

- Pour générer les indices aléatoires, voici comment on procède :
1 – On appelle la fonction *Math.random()* qui génère un nombre aléatoire entre 0 et 1.
2 – On multiplie le nombre généré entre 0 et 1 par la taille totale de la population afin que le nombre généré se trouve entre 0 et la taille de la population : *Math.random()*pop.length*
3 – Finalement, on garde la partie entière du nombre généré de manière à avoir un indice entier généré aléatoirement entre 0 et la taille du tableau de valeurs (...et donc de la population) :
int pers1 = (int) pers1a

- On effectue cela deux fois pour générer deux indices différents, puis on compare les valeurs qui se trouvent à ces indices simplement grâce à la structure conditionnelle *if {...} else if {...}*

Notes d'édition :
(1) « La seconde spécificité que l'on peut … chaque miliseconde ». Dans ce court paragraphe les auteurs tentent de justifier le fait que l'on peut supposer qu'il y a une rencontre par unité de temps. Ce n'est évidemment pas un modèle réaliste (ce que les auteurs semblent ignorer) mais il est évidemment valide pour étudier la question (ce que les auteurs semblent ignorer également). La dernière phrase est tout à fait fantaisiste. Voici une proposition de remplacement : « La seconde spécificité que l'on peut noter est qu'à chaque instant n, deux personnes SEULEMENT se rencontrent. Dans un modèle plus réaliste, nous devrions faire intervenir les rencontres à des temps aléatoires. »