Cet article est rédigé par des élèves. Il peut comporter des oublis et imperfections, autant que possible signalés par nos relecteurs dans les notes d'édition.

Le jeu du chat et de la souris

Année 2013 - 2014

Elèves de Seconde :
Valère El Alami, Gianna Mönig, Max Gnamonei Koumou-Okandze, Julian Kern, Anton Neike

Etablissement :
Lycée français de Berlin

Enseignants :
Julien Duthil, Jean-Luc Pérotin

Chercheuse :
Sylvie Roelly, Université de Potsdam, laboratoire de probabilité

Introduction à la problématique

Mme Roelly, chercheuse de l'Université de Potsdam, nous a proposé un thème pour lequel il faut bien apprendre à manipuler des probabilités. Il s'agit d'un échiquier n'ayant pas de bords. Sur cet échiquier, on a deux animaux fictifs : un chat et une souris.

La question qui se pose alors est celle de la survie de la souris : Le chat va-t-il arriver à manger la souris ?

Ensuite, on peut se demander s'il y a une certaine stratégie que pourrait suivre le chat pour manger la souris le plus tôt possible. Et pour la souris : Y a-t-il une possibilité de prolonger sa vie ?

Nous allons essayer de répondre le mieux possible à ces questions dans l'article. Nous nous contenterons d'étudier un échiquier de dimension 2x2, car Mme Roelly nous a déconseillé de nous lancer sur un échiquier de dimensions supérieures, puisque les calculs deviennent impossibles à réaliser pour des élèves.

Pour pouvoir répondre, nous avons étudié différents cas par des calculs, mais également par la simulation d'un grand nombre de jeux à l'aide de programmes qui sont présentés à la fin de l'article.

Introduction aux symboles utilisés :

$A \cap B$ représente l'intersection de A avec B

$A \cup B$ représente l'union de A avec B

$$\bigcap_{i=n}^{m} A_i = A_n \cap A_{n+1} \cap \cdots \cap A_{m-1} \cap A_m$$

$$\bigcup_{i=n}^{m} A_i = A_n \cup A_{n+1} \cup \cdots \cup A_{m-1} \cup A_m$$

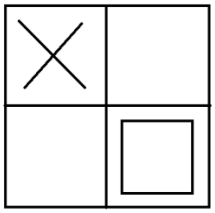
$$\sum_{i=n}^{m} a_i = a_n + a_{n+1} + \cdots + a_{m-1} + a_m$$

$$\prod_{i=n}^{m} a_i = a_n \times a_{n+1} \times \cdots \times a_{m-1} \times a_m$$

$a \to b$: a tend vers b

Règles du jeu

Soit le jeu suivant : un chat et une souris se trouvent chacun sur une case d'un échiquier fictif. Cet échiquier est de dimension 2×2 . Le temps sera mesuré en tours. Les animaux se déplaceront à chaque tour vers une case adjacente (dans une des quatre directions « haut », « bas », « droite », « gauche » [et non pas diagonalement !]). La direction des mouvements du chat et de la souris sera complètement aléatoire. Les animaux n'ont pas le droit de dépasser les bords l'échiquier.



La croix et le carré représentent chacun un animal

La souris et le chat se rencontrent lorsqu'ils se trouvent sur la même case lors du même tour.

On établit la fin du jeu à la mort de la souris. Cette mort se produit lors de la rencontre entre les deux animaux.

Définitions d'événements

On appelle T(k) l'événement « La souris meurt lors de la transition de $t = k - 1$ à $t = k$ » avec $k \in \mathbf{N}^*$.

On appelle $M(k)$ la fin du jeu qui se produit exactement au tour $t = k$ *avec* $k \in \mathbf{N}^*$, autrement dit l'événement : « la souris a survécu k tours », avec

$$M(k) := \left(\bigcap_{i=1}^{k-1} \overline{T(i)}\right) \cap T(k). \qquad (1)$$

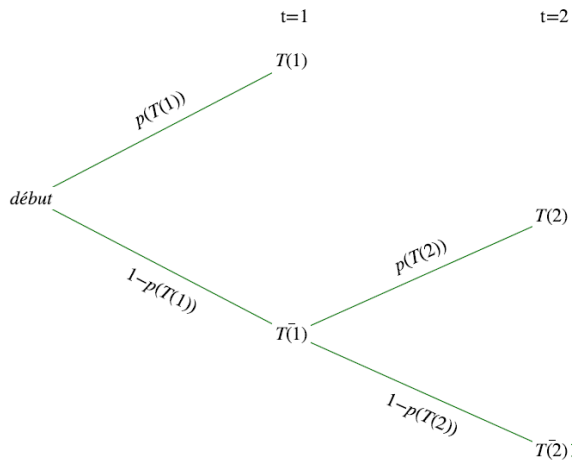
Comme chaque tour est indépendant des autres, on peut écrire

$$p(M(k)) = \left(\prod_{i=1}^{k-1} p(\overline{T(i)})\right) \times p(T(k))$$

$$= \left(p(\overline{T(i)})\right)^{k-1} \times p(T(k))$$

où $p(M(k))$ est la probabilité de l'événement $M(k)$.

Visualisation de l'arbre de probabilité pondéré représentant les deux premiers tours :



Le chemin « $\overline{T(1)}, T(2)$ » représente l'événement $M(2)$.

Or le jeu est fini après $T(1)$, l'arbre est tronqué.

On appelle m_k l'événement « la souris meurt au plus tard au tour $t = k$ *avec* $k \in \mathbf{N}^*$ », avec

$$m_k := \bigcup_{i=1}^{k} M(i)$$

Comme les événements sont incompatibles, on peut écrire :

$$p(m_k) = \sum_{i=1}^{k} p(M(i)),$$

où $p(m_k)$ est la probabilité de l'événement m_k.

Cas étudiés

Premier cas étudié :

La situation initiale est univoque : les animaux se trouvent sur deux cases diagonalement opposées (peu importe lesquelles, de façon à ce que, si on tourne l'échiquier, on arrive à la même position de départ), voir l'image de la page 1. Si les animaux ne commencent pas diagonalement opposés, mais l'un à côté de l'autre, ils ne peuvent pas se rencontrer, car ils se déplacent en même temps.

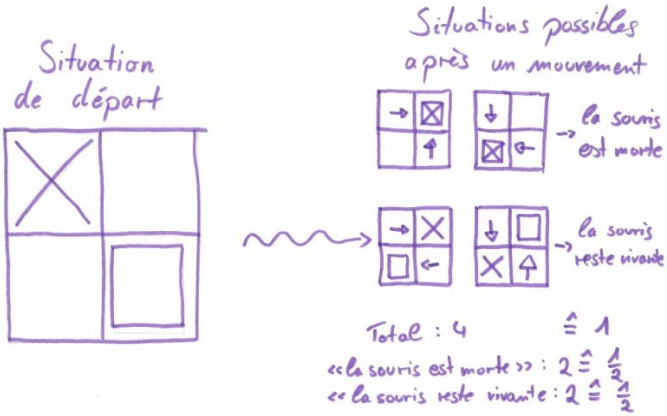


Schéma visualisant tous les mouvement possibles

Or la direction du mouvement de chaque animal, étant aléatoire, le chat, comme la souris,

a une probabilité d'un demi de tomber sur la case en haut à droite et une chance sur deux de se trouver sur la case en bas à gauche.

On peut donc conclure que la chance que les deux animaux se rencontrent sur la case en haut à gauche est d'un quart ; de même pour la case en bas à gauche.

On en déduit que

$$p(M(1)) = \frac{1}{2}.$$

On remarque que, si les animaux ne se sont pas rencontrés, ils sont de nouveau diagonalement opposés. On peut donc affirmer que la situation du tour $t = 1$ se reproduit pour chaque tour $t = k$, ce qui veut dire que :

$$\forall k \in \mathbf{N}^*, \qquad p(T(k)) = p(\overline{T(k)}) = \frac{1}{2}.$$

Alors à chaque nouveau tour, la souris a une probabilité d'un demi de mourir et une chance sur deux de vivre. Or, pour mourir exactement au 2nd tour ($t = 2$), la souris doit avoir survécu au premier tour.

Nous avons donc :

$$M(2) = \overline{T(1)} \cap T(2).$$

Cela veut dire que

$$p(M(2)) = \frac{1}{2} \times \frac{1}{2} = \frac{1}{4}.$$

Or

$$p(M(k)) = (p(\overline{T(i)}))^{k-1} \times p(T(k))$$

$$= \left(\frac{1}{2}\right)^{k-1} \times \frac{1}{2},$$

Alors :

$$p(M(k)) = \frac{1}{2^k}.$$

Nous nous intéressons maintenant à $p(m_k)$.

On a :

$$p(m_k) := \sum_{i=1}^{k} p(M(i))$$

$$= \sum_{i=1}^{k} \frac{1}{2^i}.$$

Affirmation :

$$\forall k \in \mathbf{N}^*, \qquad p(m_k) = \frac{2^k - 1}{2^k}.$$

Initialisation : L'affirmation $p(m_k) = \frac{2^k - 1}{2^k}$ est vraie pour $k = 1$, car

$$p(m_1) = \sum_{i=1}^{1} \frac{1}{2^i} = \frac{1}{2} = \frac{2^1 - 1}{2^1}.$$

Hérédité : Il suffit maintenant de démontrer que si l'affirmation est vraie pour $k = n$, alors elle est vraie pour $k = n + 1$.

Avec :

$$p(m_{n+1}) = \sum_{i=1}^{n+1} \frac{1}{2^i}$$

$$= \sum_{i=1}^{n} \frac{1}{2^i} + \frac{1}{2^{n+1}}$$

$$= \frac{2^n - 1}{2^n} + \frac{1}{2^{n+1}}$$

$$= \frac{2^{n+1} - 2 + 1}{2^{n+1}}$$

$$= \frac{2^{n+1} - 1}{2^{n+1}}.$$

Conclusion : Alors, par récurrence, l'affirmation $p(m_k) = \frac{2^k - 1}{2^k}$ est vraie pour tout $k \in \mathbf{N}^*$.

On se demande alors comment évolue $p(m_k)$ quand $t \to +\infty$, c'est à dire quelle est la probabilité que la souris meure si le chat a une infinité de tours pour la rencontrer. On trouve alors

$$\lim_{k \to +\infty} p(m_k) = \lim_{k \to +\infty} \left(\frac{2^k - 1}{2^k}\right)$$

$$= \lim_{k \to +\infty} \left(1 - \frac{1}{2^k}\right)$$

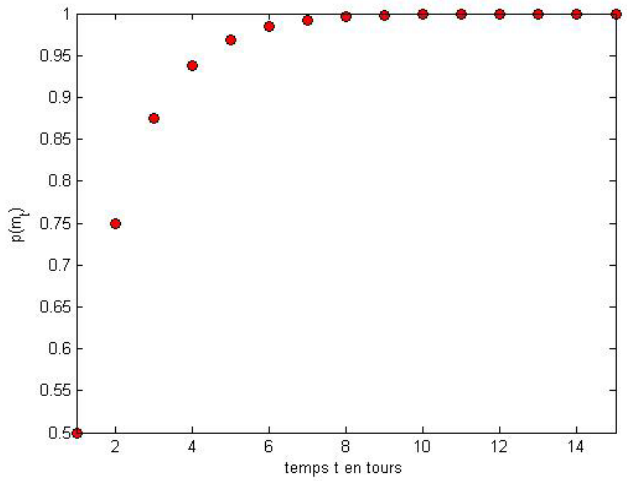
$$= 1 - \lim_{k \to +\infty} \left(\frac{1}{2^k}\right).$$

Comme le second terme converge vers 0, on peut conclure que

$$\lim_{k \to +\infty} p(m_k) = 1.$$

On vient ainsi de démontrer que la souris, lorsqu'elle se trouve avec un chat sur un échiquier 2×2, mourra avec probabilité de 1. Mais il faut rappeler qu'il existe des configurations dans lesquelles le chat ne mange pas la souris (p. ex. : les animaux bougent parallèlement), mais que ces cas, vu la multitude de cas où la souris meurt, sont négligeables.

Le graphique suivant nous montre une visualisation de cette suite (cf programme 1) :



L'évolution de la suite $p(m_t)$ en fonction du nombre de tours t

On voit bien avec quelle vitesse les points s'approchent de 1.

Se pose alors la question la plus intéressante :

QUAND le chat mangera-t-il la souris ?

Autrement dit : Quelle est l'espérance de vie d'une souris qui se trouve avec un chat sur un échiquier de dimension 2×2 ?

L'espérance de vie est la moyenne des probabilités de mourir au tour k, pondérée par k. On peut écrire l'espérance (nommée E) avec

$$E := \sum_{i=1}^{+\infty} i \times p(M(i)).$$

Pour pouvoir calculer l'espérance de vie, on réorganise la somme. Ce réarrangement permet de subdiviser la somme en d'autres sommes qui sont plus simples à calculer :

$$\sum_{i=1}^{+\infty} i \times \frac{1}{2^i} = \frac{1}{2} + \frac{2}{4} + \frac{3}{8} + \frac{4}{16} + \frac{5}{32} + \cdots$$

$$= \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \cdots + \frac{1}{8} + \cdots$$

$$= \sum_{i=1}^{+\infty} \frac{1}{2^i} + \sum_{i=2}^{+\infty} \frac{1}{2^i} + \sum_{i=3}^{+\infty} \frac{1}{2^i} + \sum_{i=4}^{+\infty} \frac{1}{2^i} + \cdots$$

$$= \sum_{i=1}^{+\infty} \frac{1}{2^{i+0}} + \sum_{i=1}^{+\infty} \frac{1}{2^{i+1}} + \sum_{i=1}^{+\infty} \frac{1}{2^{i+2}} + \cdots$$

$$= \sum_{i=1}^{+\infty} \frac{1}{2^i} + \frac{\sum_{i=1}^{+\infty} \frac{1}{2^i}}{2} + \frac{\sum_{i=1}^{+\infty} \frac{1}{2^i}}{4} + \frac{\sum_{i=1}^{+\infty} \frac{1}{2^i}}{8} + \cdots$$

$$= \sum_{j=0}^{+\infty} \frac{\sum_{i=1}^{+\infty} \frac{1}{2^i}}{2^j} = \sum_{j=0}^{+\infty} \frac{1}{2^j}$$

$$= 1 + \sum_{j=1}^{+\infty} \frac{1}{2^j} = 1 + 1$$

$$= 2.$$

Annotation : Le remplacement peut se faire car

$$\sum_{i=1}^{+\infty} \frac{1}{2^i} = 1,$$

comme on l'a vu auparavant.

On peut donc conclure que l'espérance de vie de la souris dans cette situation est de deux tours.

<u>Deuxième cas étudié :</u>

Les animaux se trouvent comme dans la situation précédente sur un échiquier de

dimension 2x2. Le chat reste immobile pendant toute la durée du jeu, alors que la souris bouge à chaque tour (de la même façon que dans la situation précédente). Les autres règles ne changent pas.

Les conséquences de ce changement rendent ce problème équivoque. On choisit alors le même état initial que lors du premier cas, c'est à dire un positionnement des animaux sur des cases diagonalement opposées.

On en déduit que $p(M(1)) = 0$, car la souris ne peut se déplacer que d'une case par tour, et n'a donc pas la possibilité de rencontrer le chat au premier tour.

Elle se trouve alors sur une case adjacente à celle du chat. On peut en déduire que la souris a une chance sur deux de se déplacer sur la case du chat et une chance d'un demi de revenir sur la case initiale. Comme la souris est sûre de survivre au premier tour, on peut écrire :

$$M(2) = \overline{T(1)} \cap T(2),$$

On a donc :

$$p(M(2)) = 1 \times \frac{1}{2} = \frac{1}{2}.$$

Mais si la souris ne meurt pas au deuxième tour, elle est sûre de ne pas pouvoir être mangée au troisième tour. On en déduit donc que cette situation se répète tous les deux tours. La souris ne peut donc être mangée que tous les tours où $t = 2k \text{ avec } k \in \mathbf{N}^*$. On en déduit que :

$$\forall k \in \mathbf{N}^*, \qquad p(M(2k-1)) = 0.$$

Cela signifie que les tours où $t \text{ est impair}$ n'affectent pas la somme qui représente l'espérance de vie

$$E = \sum_{i=1}^{+\infty} p(M(i)).$$

On en déduit que

$$\forall k \in \mathbf{N}^*, \qquad p(M(2k)) = \frac{1}{2^k},$$

semblable à ce que nous avons trouvé pour le premier cas.

Maintenant que l'on connaît $p(M(k))$, on s'intéresse de nouveau à $p(m_k)$. Or on sait que

$$p(m_{2k}) = \sum_{i=1}^{2k} p(M(i)),$$

on peut dire que

$$p(m_{2k}) = 0 + \frac{1}{2} + 0 + \cdots + \frac{1}{2^{k-1}} + 0 + \frac{1}{2^k}.$$

Or, en réorganisant cette somme, on obtient

$$p(m_{2k}) = \frac{1}{2} + \frac{1}{4} + \cdots + \frac{1}{2^{k-1}} + \frac{1}{2^k} + 0,$$

on peut en déduire que

$$p(m_{2k}) = \sum_{i=1}^{k} \frac{1}{2^i}.$$

Cela nous permet de dire que

$$\lim_{k \to +\infty} p(m_{2k}) = 1.$$

On démontre de même que :

$$\lim_{k \to +\infty} p(m_{2k+1}) = 1$$

et on obtient donc :

$$\lim_{k \to +\infty} p(m_k) = 1.$$

Donc, comme dans le premier cas, la souris est condamnée à mourir.

Une fois que nous sommes sûrs que la souris meurt, nous nous intéressons à la durée de la vie de la souris. On la notera E, avec

$$E := \sum_{i=1}^{+\infty} i \times p(M(i)).$$

Mais comme

$$\forall k \in \mathbf{N}^*, \qquad p(M(2k-1)) = 0,$$

on ne s'intéresse qu'aux termes où $i \text{ est pair}$. On a donc :

$$E = \frac{2}{2} + \frac{4}{4} + \frac{6}{8} + \frac{8}{16} + \frac{10}{32} + \frac{12}{64} + \frac{14}{128} + \cdots$$

$$= 2 \times \left(\sum_{i=1}^{+\infty} \frac{1}{2^i} \right) + 2 \times \left(\frac{\sum_{j=1}^{+\infty} \frac{1}{2^j}}{2} \right) + \cdots$$

$$= 2 \times \left(\sum_{j=0}^{+\infty} \frac{\sum_{i=1}^{+\infty} \frac{1}{2^i}}{2^j} \right).$$

Comme on sait que la somme à l'intérieur converge vers 2 (voir cas précédent), on peut en déduire que

$$E = 4.$$

On peut donc conclure que la souris vit en moyenne 4 tours. Il est possible d'utiliser le même raisonnement pour le cas où le chat bouge, mais la souris reste immobile.

Il est donc préférable pour le chat de bouger plutôt que de rester immobile.

Troisième cas étudié :

Le jeu prend lieu sur le même échiquier que dans les autres situations. Le chat reste immobile. La souris peut bouger diagonalement (mais peut également bouger comme avant). Aucune direction n'est préférée.

La position initiale, n'étant pas univoque, est choisie comme avant : Les animaux sont diagonalement opposés.

Soit V_1 l'état où les animaux se trouvent diagonalement opposés et V_2 l'état où les animaux se trouvent sur des cases adjacentes.

Après l'état V_1, on se retrouve dans l'état V_2 avec une chance de $\frac{2}{3}$ ou la souris meurt avec une chance d'un tiers.

Après l'état V_2 on retrouve soit V_1 soit V_2 soit le jeu est terminé (chaque événement ayant une chance sur trois).

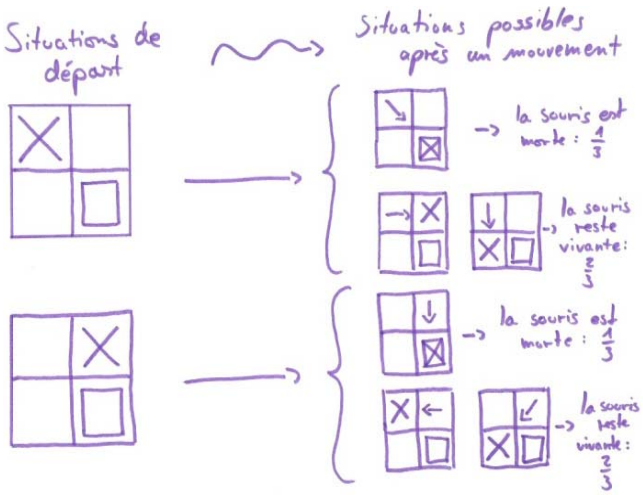


Schéma visualisant tous les mouvement possibles

On en déduit que pour les deux états où la souris est vivante, celle-ci meurt avec une probabilité d'un tiers dans le prochain tour, soit

$$\forall k \in \mathbf{N}^*, \qquad p(T(k)) = \frac{1}{3},$$

ce qui est équivalent à

$$\forall k \in \mathbf{N}^*, \qquad p(\overline{T(k)}) = \frac{2}{3}.$$

On en déduit que

$$p(M(k)) = \left(\frac{2}{3}\right)^{k-1} \times \frac{1}{3},$$

et donc que

$$p(m_k) = \sum_{i=1}^{k} \left(\frac{2}{3}\right)^{i-1} \times \frac{1}{3}$$

$$= \frac{1}{3} \times \sum_{i=1}^{k} \left(\frac{2}{3}\right)^{i-1}.$$

Affirmation :

$$\forall k \in \mathbf{N}^*, \qquad p(m_k) = 1 - \frac{2^k}{3^k}.$$

Initialisation : L'affirmation $p(m_k) = 1 - \frac{2^k}{3^k}$ est vrai pour $k = 1$, car

$$p(m_1) = \frac{1}{3} \times 1 = \frac{1}{3} = 1 - \frac{2}{3}.$$

Hérédité : Il suffit maintenant de démontrer que si l'affirmation est vraie pour $k = n$, alors elle est vraie pour $k = n + 1$.

Avec :

$$p(m_{n+1}) = \frac{1}{3} \times \sum_{i=1}^{n+1} \left(\frac{2}{3}\right)^{i-1}$$

$$= \frac{1}{3} \times \left(\sum_{i=1}^{n} \left(\frac{2}{3}\right)^{i-1}\right) + \frac{1}{3} \times \frac{2^n}{3^n}$$

$$= \left(1 - \frac{2^n}{3^n}\right) + \frac{1}{3} \times \frac{2^n}{3^n}$$

$$= \frac{3^n - 2^n}{3^n} + \frac{2^n}{3^{n+1}}$$

$$= \frac{3^{n+1} - 3 \times 2^n + 2^n}{3^{n+1}}$$

$$= \frac{3^{n+1}}{3^{n+1}} - \frac{2^n(3-1)}{3^{n+1}}$$

$$= 1 - \frac{2^{n+1}}{3^{n+1}}.$$

Conclusion : Alors, par récurrence, l'affirmation

$p(m_k) = 1 - \frac{2^{n+1}}{3^{n+1}}$ est vraie pour tout $k \in \mathbf{N}^*$.

On a donc

$$\lim_{k \to +\infty} p(m_k) = 1 - \lim_{k \to +\infty} \frac{2^k}{3^k}.$$

On peut donc conclure que

$$\lim_{k \to +\infty} p(m_k) = 1.$$

On peut en déduire que la souris a une vie finie.

Pour l'espérance de vie E, il ne nous reste plus qu'à calculer

$$E = \frac{1}{3} \times \sum_{i=1}^{+\infty} i \times \left(\frac{2}{3}\right)^{i-1}.$$

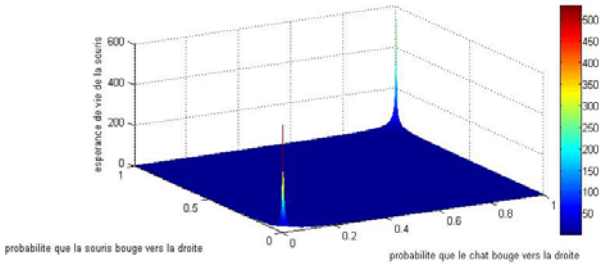
Expression dont nous déterminerons la valeur plus tard.

Quatrième cas étudié (généralisation partielle) :

On s'est ensuite intéressé aux changements de l'espérance de vie de la souris lorsque l'on introduit une préférence dans la direction de mouvement choisie par l'animal.

La position initiale redevient univoque, les animaux ne peuvent se trouver que sur les cases diagonalement opposés. Les deux bougent à chaque tour, mais il existe une préférence pour une direction (droite/gauche). Les animaux n'ont pas le droit de bouger diagonalement. (2)

Pour visualiser cette situation, on l'a simulée (cf programme n°2) :



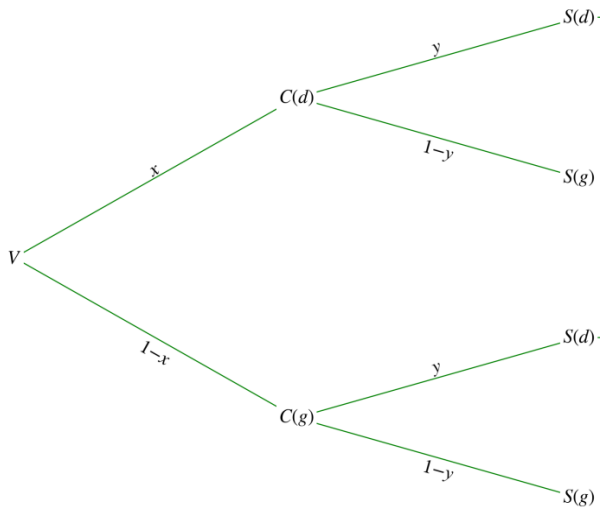
Espérance de vie de la souris en fonction des probabilités de chaque animal de se déplacer vers la droite

On peut d'abord faire deux constats qui semblent naturels : Si les deux animaux bougent à chaque tour vers la droite ou toujours vers la gauche, ils ne se rencontrent jamais. Et si l'un bouge vers la droite et l'autre vers la gauche, ils se rencontrent toujours au premier tour. Ce qui est intéressant est plutôt qu'on a une certaine uniformité pour la plus grande partie du graphique, soit une espérance de vie à peu près constante sauf pour les cas extrêmes.

On s'est alors demandé, s'il est possible, de déterminer l'espérance de vie par des calculs. Pour cela, nous avons d'abord défini deux nouvelles variables x et y qui représentent respectivement la probabilité du chat et de la souris de bouger vers la droite. Nous avons ensuite remarqué que la souris doit aller vers la droite et le chat vers la gauche ou inversement pour que les animaux se rencontrent.

Soient :
C(d) : « chat bouge vers la droite »,
C(g) : « chat bouge vers la gauche »
et de même pour la souris, S(d) et S(g).

On peut donc construire l'arbre de probabilité suivant :



On remarque que soit la souris meurt, soit les animaux reviennent à la position de départ.

Or il n'y a que deux cas possibles (V : « souris vivante » et M : « souris morte ») qui pourrait impliquer d'autres probabilités, on peut dire que $p(T(k))$ reste constant pour tout $k \in \mathbf{N}^*$, avec

$$p(T(k)) = x(1-y) + y(1-x)$$

$$= x + y - 2xy.$$

On peut donc continuer :

$$p(M(k)) = \left(p(\overline{T(k)})\right)^{k-1} \times p(T(k)),$$

soit

$$E = p(T(k)) \times \sum_{i=1}^{+\infty} i \times \left(p(\overline{T(k)})\right)^{i-1}$$

$$= p(T(k)) \times \sum_{n=0}^{+\infty} (n+1) \times \left(p(\overline{T(k)})\right)^{n}$$

en remplaçant i par $n+1$ (changement d'indice).

Sur internet (wolframalpha.com), vérifié par nos professeurs, nous avons trouvé que

$$\sum_{n=0}^{+\infty} (n+1) \times x^n = \frac{1}{(x-1)^2}$$

$$pour\ tout\ |x| < 1.$$

On peut donc conclure que

$$E = \frac{p(T(k))}{(p(\overline{T(k)}) - 1)^2} \qquad (3)$$

$$= \frac{p(T(k))}{(1 - p(T(k)) - 1)^2}$$

$$= \frac{p(T(k))}{\left(-p(T(k))\right)^2}$$

$$= \frac{p(T(k))}{\left(p(T(k))\right)^2}$$

$$= \frac{1}{p(T(k))}$$

$$= \frac{1}{x + y - 2xy}.$$

On peut tout de suite affirmer qu'il existe une symétrie : x et y ont exactement la même influence et on peut les échanger sans changer le résultat. On remarque ensuite qu'en remplaçant une des deux variables par $\frac{1}{2}$ on obtient :

$$E = \frac{1}{\frac{1}{2} + y - 2 \times \frac{1}{2} y}$$

$$= \frac{1}{\frac{1}{2} + y - y}$$

$$= \frac{1}{\frac{1}{2}} = 2.$$

Ceci dit que **l'autre variable peut prendre toute valeur** et **le résultat ne change pas.**

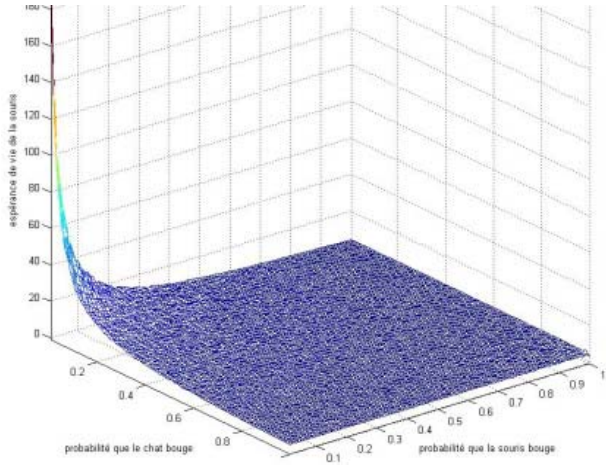
Quel rapport avec notre situation ? Ce calcul nous dit que si la souris n'utilise pas de stratégie (elle n'a pas de préférence pour une direction), il n'y a aucune différence si le chat bouge plutôt vers la gauche ou vers la droite.

Réussite de programmation :

Une simulation de la première situation. Cette simulation (cf. programme n°3) donne des résultats très proches à la valeur théorique pour un nombre de jeux simulés supérieur à 10 000 000 : $E_{simulation} = 2{,}0080683$ (une faute d'environ 0,4 %).

Le programme est écrit en python 3.3 et est joint en document.

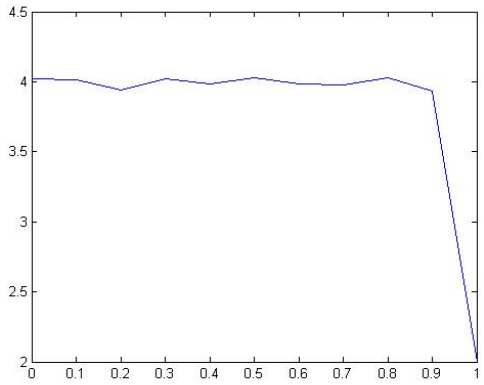
Des simulations à l'aide de matlab (cf programme n°4) nous permettent enfin de faire quelques affirmations, qui sont très intéressantes :



Espérance de vie de la souris en fonction de la probabilité de chaque animal de bouger

Ce graphe nous permet d'affirmer ce qui semble normal : si les deux animaux bougent très peu, il dure plus longtemps jusqu'à ce que le chat rencontre la souris. Il nous dit aussi que la croissance de l'espérance de vie est exponentielle. Mais il révèle aussi quelque chose d'incroyable : Si la souris bouge tout le temps, l'espérance de vie de la souris reste constante à 4 pour toute probabilité du chat de bouger. Seulement pour les probabilités proches de 1, elle baisse jusqu'atteindre 2.

Si on ne prend en compte que la probabilité du chat et admet que la souris bouge tous les tours (cf. programme n°5), on obtient



Espérance de vie de la souris en fonction de la probabilité du chat de bouger (la souris bouge à chaque tour)

où les abscisses représentent la probabilité que le chat bouge et les ordonnées l'espérance de vie de la souris en tours.

Conclusion

Nous pouvons donner en conclusion quelques conseils aux chats et aux souris qui veulent s'essayer à ce jeu :

- Aux souris : Ne le faites pas! Vous mourrez certainement. Si vous le voulez néanmoins, ne bougez surtout pas! Cela peut rallonger votre vie de quelques tours.

- Aux chats : Faites-le ! Vous ne pouvez que gagner. Mais bougez à chaque tour. Cela peut vous remporter votre repas plus rapidement.

- Aux deux : Peu importe la direction que vous prenez ; si votre adversaire n'a pas de préférence, la direction que vous prenez ne fait pas de différence.

ANNEXE : Programmes

Programme n°1 :

```
clc;clf;
%Visualisation de la probabilité de la mort de la souris après un certain
%temps.

i = 0; % Le nombre de tour
a = zeros(1,20);
x = []; % Le temps (abscisses)
y = []; % La probabilité de la mort de la souris (ordonnées)

for i=1:1:15

   a(i) = 0.5.^i;            % l'élément actuel de la suite
   x(i) = i;                 % construction des abscisses
   y(i) = sum(a);            % construction des ordonnées

   figure(1);   % construction de l'image
   plot(x,y,'ok','MarkerSize', 7, 'MarkerFaceColor', 'r');
   xlabel('temps t en tours');
   ylabel('p(m_t)');

   axis([1 15 0.5 1.0]);
end;
```

Programme n°2:

```
%{
Dieses Programm simuliert das Spiel in einem 2x2 Feld während wir die
Richtungswahrscheinlichkeit ändern. Es ist möglich, dass die beiden Tiere
verschiedene Richtungswahrscheinlichkeiten haben. Es wird sich also um
einen 3D-plot handeln.

x-Achse: Richtungswahrscheinlichkeit der Katze
y-Achse: Richtungswahrscheinlichkeit der Maus
z-Achse: Lebenserwartung der Maus
//
Ce programme simule le jeu sur un échiquier 2x2 alors que la probabilité que l'animal aille
dans une certaine direction varie. Il est possible que la probabilité diffère entre les deux
animaux. On aura alors un graphique
en 3D.

x: probabilité que le chat se dirige vers la droite
y: probabilité que la souris se dirige vers la droite
z: espérance de vie de la souris (en tours)
%}

clc;
clf;
clear;

p = 1000;                 % proba change de 1/p pour chaque simulation
jeux = 1000;              % précision des résultats

x = 1/p:1/p:1-1/p;
[X,Y] = meshgrid(x,x);      % abscisses et ordonnées
Z = zeros(p-1, p-1);        % z

r1 = randi(p, 100000, 1);   % 100000 nombres aléatoires
r2 = randi(p, 100000, 1);   % ""
l = 0;                      % compteur

for i = 1:1:p-1             % boucle pour proba de la souris
   i
   for j = 1:1:i            % boucle pour proba du chat
      m = zeros(1,jeux+1); % vecteur dans lequel on écrit la "date de mort"
      for k = 1:1:jeux      % boucle des jeux
         s = [0,0];
         c = [1,1];
         n = 0;             % compteur
         while 1 == 1       % boucle seulement interrompue par un "break"
            v1 = 0;
            v2 = 0;
            if l == 100000 % si chaque nombre aléa. fut utilisé une fois
               l = 0;       % on crée de nouveaux nombres
               r1 = randi(p, 100000, 1);
```

```
            r2 = randi(p, 100000, 1);
          end;
          m(k+1) = m(k+1)+1;
          l = l + 1;
          if r1(l) <= i          % avec proba de la
souris pour mvt
             v1 = 1;
          end;
          if r2(l) <= j          % avec proba du chat
pour mvt
            v2 = 1;
          end;
          s = mouvement_special(s, v1); % si
v1=1: vers la droite
          c = mouvement_special(c, v2); % si
v2=1: vers la droite
          if n > 200                  % s'il y a trop de
tours
            break;
          end;
          if c == s                 % fini dès leur
rencontre
            break;
          end;
        end;
      end;
      Z(i,j) = sum(m)/jeux;        % inscription de
l'esp de vie dans z
      Z(j,i) = Z(i,j);             % comme il y a
symétrie
    end;
end;

surf(X,Y,Z);              % création du graphe
```

Avec la fonction « mouvement_special.m » :

```
%{
fonction spécial qui prend en compte la
préférence des animaux d'aller vers
la droite.
La direction fur déjà affectée dans le
programme principale :

v = 0 : droite;
v = 1 : gauche;


%}

function a = mouvement_special(a, v)
if v == 0                          % si vers la gauche:
   if a == [0,0]
      a = [1,0];
   elseif a == [0,1]
      a = [0,0];                   % dépendant de la
position
   elseif a == [1,0]
      a = [1,1];
   else
      a = [0,1];
   end;
else                             % si vers la droite
   if a == [0,0]
      a = [0,1];
   elseif a == [0,1]
      a = [1,1];                   % dépendant de la
position
   elseif a == [1,0]
      a = [0,0];
   else
      a = [1,0];
   end;
end;
```

Programme n°3:

(4)

Programme n°4:

```
%{
Sowohl die Katze, als auch die Maus, haben
jetzt eine
Bewegungswahrscheinlichkeit, die jeweils
unterschiedlich sein kann.
Es wird sich also um einen 3D-plot handeln.

x-Achse: Bewegungswahrscheinlichkeit der
Katze
y-Achse: Bewegungswahrscheinlichkeit der
Maus
z-Achse: Lebenderwartung der Maus

//
Ce programme simule l'espérance de vie de la
souris en fonction de la
paresse des deux animaux. Or chaque individu
aura une probabilité propre,
```

```
il s'agira d'un graphique à 3 dimensions.

x: probabilité que le chat se déplace
y: probabilité que la souris se déplace
z: espérance de vie de la soursi en tours
%}

clf;
clc;
clear;

p = 100;                % proba change de 1/p
pour chaque simulation
jeux = 1000;            % précision des résultats


x = 1/p:1/p:1;
[X,Y] = meshgrid(x,x);      % abcisses et
ordonnées
Z = zeros(p, p);            % z

r1 = randi(p, 100000, 1);   % 100000 nombres
aléatoires
r2 = randi(p, 100000, 1);   % """"
l = 0;                  % compteur

for i = 1:1:p               % boucle pour proba de la
souris
   i
   for j = 1:1:i            % boucle pour proba du
chat
      m = zeros(1,jeux+1); % vecteur dans lequel
on érit la "date de mort"
      for k = 1:1:jeux      % boucle des jeux
         s = [0,0];
         c = [1,1];
         n = 0;            % compteur
         while 1 == 1       % boucle seulement
interrompue par un "break"

            if l == 100000 % si chaque nombre
aléa. fut utilisé une fois
               l = 0;       % on crée de nouveaux
nombres
               r1 = randi(p, 100000, 1);
               r2 = randi(p, 100000, 1);
            end;
            m(k+1) = m(k+1)+1;
            l = l + 1;
            if r1(l) <= i  % avec proba de la souris
pour mvt
               s = mouvement(s);
            end;
            if r2(l) <= j  % avec proba du chat pour
mvt
               c = mouvement(c);
            end;
            if n > 200     % s'il y a trop de tours
               break;
            end;
            if c == s      % fini dès leur rencontre
               break;
            end;
         end;
      end;
      Z(i,j) = sum(m)/jeux;    % inscription de
l'esp de vie dans z
      Z(j,i) = Z(i,j);         % comme il y a
symmétrie
   end;
end;

mesh(X,Y,Z);                % création du graphe
```

Avec la fonction « mouvement.m » :

```
%{
Diese Funktion dient der Bewegung der Tiere.
Da die Spiele in einem 2x2-Feld stattfinden,
kann man die Funktion stark
vereinfachen.
Das Tier a besitzt eine x-Koordinate (a(1)) und
eine y-Koordinate (a(2)).
Das Feld besitzt 4 Felder: (0,0), (0,1), (1,0) und
(1,1). Auch wird die
Funktion dadurch vereinfacht, dass keine
Richtung von den Tieren bevorzugt
wird (immer halb, halb).

//

Cette fonction sert à bouger les animaux.
Or le jeu prend place dans un échiquier 2x2, on
peut simpligier la fonction
énormément.
L'animal possède une abscisse (a(1)) et une
ordonnée (a(2)).
L'échiquier est constitué de 4 cases: (0,0), (0,1),
(1,0), (1,1). Aussi,
comme les animaux n'ont pas de préférence
pour une certaine dirrection, la
fonction reste assez primitive.
```

```
%}

function a = mouvement(a)
r = randi(2);            % probabilité 1/r d'aller
vers la droite
if (r == 1)
   if (a(1) == 0)
      a(1) = 1;         % dependant de la position
   else
      a(1) = 0;
   end
else
   if (a(2) == 0)
      a(2) = 1;         % dépendant de la position
   else
      a(2) = 0;
   end
end
end
```

Programme n°5 :

```
clc;
clear;
clf;

%{
Le programme simule des jeux avec un
changement de la probabilité que le
chat bouge (de 0 à 1 dans des pas de 1/p).
%}

p = 10;                  % nombre de
simmulation
x = 0:1/p:1;             % construction des
abcisses
y = zeros(1,p+1);        % préparation des
ordonnées
jeux = 10000;            % nombre de jeux par
simulation
r = randi(p, 10000, 1);
l = 0;                   % compteur

for i = 0:1:p            % boucle extérieure des
simulations
   m = zeros(1,jeux+1);  % préparation pour
inscrire le moment de la mort de la souris

   for j = 0:1:jeux      % boucle intérieure des
jeux
      s = [0 0];         % initialisation souris
      c = [1 1];         % initialisation chat
      n = 0;             % compteur

      while 3 > 2        % pendant un jeu
         if (l > 9999)   % si tous les
nombres aléatoires
            l = 0;       % furent utilisés, on
crée de
            r = randi(p, 10000, 1); % nouveaux.
         end;
         m(j+1) = m(j+1)+1;   % jusqu'à la
mort de la souris,
         l = l + 1;      % il augmente le
numéro de tours
         s = mouvement(s);    % de 1.
         if (r(l) <= i)  % simule la
probabilité
            c = mouvement(c);   % le chat
bouge
         end;
         if (n > 100)    % si trop long, la
boucle s'arrête (Ce qui n'est jamais arrivé)
            break
         end;
         if (c == s)     % si la souris est
morte
            break        % m(j+1) ne change
plus
         end;

      end;
   end;
   y(i+1) = sum(m)/jeux;     % la moyenne
des espérances de vie
end;

plot(x,y);               % construction du
graphique
```

Notes d'édition

(1) La barre au dessus d'un événement signifie que l'on considère l'événement « complémentaire ». Dans le cas de cette formule, T(i) avec la barre correspond donc à l'événement « La souris ne meurt pas de la transition de t = i-1 à t= i ».

(2) La notion de direction droite/gauche n'est pas très claire. Si la souris (ou le chat) allait toujours vers la droite (ou la gauche), elle serait bloquée par le bord de l'échiquier. Les auteurs ont probablement voulu dire que la souris et le chat ne peuvent bouger sur l'échiquier qu'en tournant autour de son centre dans le sens horlogique ou anti-horlogique, avec éventuellement une préférence pour l'un des sens.

(3) Dans cette formule, 'k' apparait comme un argument dans le membre de droite, et pas dans le membre de gauche ('E' n'est pas une fonction de 'k'). En réalité, 'k' n'est plus un argument du membre de droite puisque, comme mentionné juste avant, la probabilité p(T(k)) est indépendante de 'k'. C'est également le cas pour la probabilité de l'événement complémentaire. 'k' devrait être retiré du membre de droite de cette formule.

(4) Le programme n°3 n'a pas été transmis.