Le lièvre, la tortue et l'ordinateur

Illustration de quelques notions du programme de probabilités du lycée à l'aide d'Algobox

L. Goudenège et P.-A. Zitt

16 janvier 2014

1 Objectifs

Dans les mots de son auteur Pascal Brachet, « AlgoBox est un logiciel libre, multi-plateforme et gratuit d'aide à l'élaboration et à l'exécution d'algorithmes dans l'esprit des nouveaux programmes de mathématiques du lycée. » Il est disponible gratuitement pour Linux, MacOS et Windows sur `http://www.xm1math.net/algobox/`.

Ce document vise à développer un exemple disponible sur le site d'Algobox sur la simulation d'un jeu aléatoire simple. On peut ainsi illustrer informatiquement des notions clés du programme : les intervalles de fluctation, quand le phénomène aléatoire est parfaitement connu, et les intervalles de confiance, quand on cherche à estimer des paramètres du phénomène.

Tous les fichiers source (Algobox et R) sont disponibles sur le site :
`http://zitt.perso.math.cnrs.fr/journee_formation.html`

2 Prise en main d'Algobox sur un exemple simple

La prise en main d'Algobox est aisée. Une fois le programme lancé, on peut ouvrir un fichier préexistant ou éditer un nouveau programme. Le code source du programme s'affiche dans la grande zone blanche. Tout le programme s'édite en cliquant sur les boutons : à droite du cadre principal on trouve de quoi rajouter, supprimer ou éditer les lignes de programme ; en bas de l'écran (dans l'onglet « Opérations standards ») on peut remplir les lignes avec des instructions (affectations, boucles, ...).

On jette un dé 50 fois. On veut illustrer cette expérience. Pour faire cela avec Algobox on doit savoir :

1. déclarer des variables : `n` (le nombre d'essais), `resultatCourant`, `i` (un indice pour la boucle) ;
2. faire une boucle de longueur fixée à l'avance (boucle "pour/for") ;
3. faire un tirage aléatoire ;
4. afficher à l'écran du texte et une variable.

En pratique

Écrire un tel programme avec Algobox. Les fonctions de calcul disponibles dans Algobox s'affichent dans un menu lorsqu'on clique sur `AFFECTER valeur a Variable` ou `AFFICHER Calcul`.

On veut ensuite calculer la moyenne empirique. On peut garder en mémoire la liste complète des résultats et utiliser les fonctions mises à disposition par Algobox :

1. on déclare une variable `resultats` de type liste ;
2. on la remplit par l'instruction `resultats[i] prend_la_valeur` (obenue en cliquant sur le bouton `AFFECTER valeur a Variable`) ;
3. dans l'onglet `Dessiner dans un repère` du cadre inférieur, on coche la case « utiliser le repère » et on règle l'échelle ;
4. on trace le graphique dans le programme avec `TRACER_POINT`.

Le code complet, donné dans le fichier `lancer_de_LGN.alg`, est le suivant.

En pratique

Ouvrir et tester le programme « `lancer_de_LGN.alg` ».

```
VARIABLES
n EST_DU_TYPE NOMBRE
i EST_DU_TYPE NOMBRE
de EST_DU_TYPE NOMBRE
moyenne_courante EST_DU_TYPE NOMBRE
resultats EST_DU_TYPE LISTE
DEBUT_ALGORITHME
    n PREND_LA_VALEUR 100
    // Lancers des dés
    POUR i ALLANT_DE 1 A n
        DEBUT_POUR
        de PREND_LA_VALEUR ALGOBOX_ALEA_ENT(1,6)
        resultats[i] PREND_LA_VALEUR de
        FIN_POUR
    // Évolution de la moyenne empirique
    POUR i ALLANT_DE 1 A n
        DEBUT_POUR
        moyenne_courante PREND_LA_VALEUR ALGOBOX_MOYENNE(resultats,1,i)
        TRACER_POINT (i,moyenne_courante)
        FIN_POUR
FIN_ALGORITHME
```

3 Le lièvre et la tortue : problème et solution mathématique

Le lièvre et la tortue décident de faire la course sur une piste à 5 cases. Ils commencent tous les deux sur la case 1. Ils décident de jouer la course aux dés, en jetant à plusieurs reprises un dé à 6 faces.

— si le dé donne un chiffre entre 1 et 5, la tortue avance d'une case — si elle atteint ainsi la case 5 elle gagne ;
— si le dé donne 6 le lièvre va immédiatement à la case 5 et gagne.

On cherchera ici simplement à calculer la probabilité de gain du lièvre. Notons T le numéro de la case où se trouve la tortue à la fin de la partie. La variable T prend ses valeurs entre 1 et 5, et on a :

$$\{T = 5\} = \text{« la tortue gagne »}, \qquad \{T \leq 4\} = \text{« le lièvre gagne »}.$$

Comme le jeu consiste à répéter de façon indépendante une expérience identique et à attendre le premier « succès » (ici le premier 6), en s'arrêtant quoiqu'il arrive au 5e essai, la variable T suit une loi géométrique tronquée en 5. Autrement dit, en notant $p = 1/6$ la probabilité de « succès » et $q = 1 - p = 5/6$ la probabilité d'« échec »,

$$\mathbb{P}[T = k] = \begin{cases} q^{k-1}p & \text{si } 1 \leq k \leq 4, \\ q^4 & \text{si } k = 5. \end{cases}$$

Les valeurs numériques sont, à 10^{-2} près,

k	1	2	3	4	5
$\mathbb{P}[X = k]$	0.17	0.14	0.12	0.10	0.48

La probabilité de gain de la tortue est donc $\mathbb{P}[T = 5] = (5/6)^4 \approx 0.48$; le lièvre est légèrement avantagé.

4 Le lièvre et la tortue : illustration avec algobox

4.1 Simulation d'une partie

Le programme suivant (`lievre_tortue.alg`, écrit [1] par le créateur d'Algobox Pascal Brachet et disponible sur le site d'Algobox) permet de simuler une partie. Après la déclaration des variables (position du lièvre, position de la tortue, résultat courant du dé), on répète un lancer de dé jusqu'à ce que la partie soit finie.

```
VARIABLES
face_du_de EST_DU_TYPE NOMBRE
case_tortue EST_DU_TYPE NOMBRE
DEBUT_ALGORITHME
    case_tortue PREND_LA_VALEUR 1
    face_du_de PREND_LA_VALEUR 1
    TANT_QUE (face_du_de<6 ET case_tortue<5) FAIRE
        DEBUT_TANT_QUE
        //Le jeu continue : on lance le dé
        face_du_de PREND_LA_VALEUR floor(6*random()+1)
        AFFICHER "Le dé donne un "
        AFFICHER face_du_de
        SI (face_du_de<6) ALORS
            DEBUT_SI
            //La tortue avance d'une case
            case_tortue PREND_LA_VALEUR case_tortue+1
            AFFICHER " -> la tortue passe à la case "
            AFFICHER case_tortue
            FIN_SI
```

1. La seule modification apportée ici est de jouer sur 5 cases et pas 6, ceci permet d'obtenir un jeu « presque équilibré », pour lequel il est difficile de savoir sur peu de parties si la tortue est avantagée ou non.

```
20:     FIN_TANT_QUE
21:   SI (case_tortue==5) ALORS
22:     DEBUT_SI
23:     AFFICHER "La tortue gagne"
24:     FIN_SI
25:   SINON
26:     DEBUT_SINON
27:     AFFICHER " -> le lièvre gagne"
28:     FIN_SINON
29: FIN_ALGORITHME
```

4.2 Parties répétées : loi des grands nombres

Si l'on répète un grand nombre de fois l'expérience en enregistrant le nombre de succès, on peut obtenir la fréquence empirique des succès et illustrer la loi des grands nombres comme dans la première partie.

En pratique

Modifier le programme précédent en rajoutant une boucle extérieure pour répéter le jeu un grand nombre de fois ; afficher la fréquence empirique de succès de la tortue.

4.3 Parties répétées : intervalle de fluctuation

Une extension possible est de calculer et représenter un intervalle de fluctuation.

Il faut pour cela calculer à part l'intervalle, puis intégrer le cœur du programme existant (celui de la loi des grands nombres) dans une boucle extérieure supplémentaire (il faut répéter un grand nombre de séries de parties pour voir si les réalisations tombent bien avec forte probabilité dans l'intervalle).

On touche ici une des limitations d'algobox : on ne peut pas découper l'algorithme en fonctions ou procédures réutilisables. Si l'on veut faire un programme plus long et complexe, il est plus simple d'utiliser un langage plus adapté comme R ou Scilab.

Des programmes R équivalents à ceux développés ici en Algobox sont disponibles sur le site `http://zitt.perso.math.cnrs.fr/journee_formation.html`.

5 Un jeu plus complexe

La piste est allongée et fait dix cases. Sur un résultat de 6, le lièvre avance de 5 cases, sur un autre résultat, la tortue avance d'une case. Le lièvre doit donc faire deux sauts avant que la tortue n'ait fait 9 pas.

L'analyse probabiliste complète est toujours possible mais sort du cadre du programme de lycée (elle est présentée plus loin). La probabilité de gain de la tortue est donc un réel r inconnu, que l'on va chercher à approcher en simulant l'expérience sur Algobox. On cherche en particulier à répondre à la question : « Qui du lièvre ou de la tortue est le plus avantagé dans cette nouvelle course ? »

En pratique

Programmer une réalisation du jeu sous Algobox en affichant la progression du lièvre et de la tortue.

5.1 Estimation ponctuelle et intervalle de confiance

Pour estimer r il faut répéter le jeu de nombreuses fois. On peut le faire avec le code suivant.

```
VARIABLES
position_lievre EST_DU_TYPE NOMBRE
position_tortue EST_DU_TYPE NOMBRE
i EST_DU_TYPE NOMBRE
n EST_DU_TYPE NOMBRE
resultats EST_DU_TYPE LISTE
moyenne_empirique EST_DU_TYPE NOMBRE
marge EST_DU_TYPE NOMBRE
DEBUT_ALGORITHME
    LIRE n
    POUR i ALLANT_DE 1 A n
        DEBUT_POUR
        position_lievre PREND_LA_VALEUR 1
        position_tortue PREND_LA_VALEUR 1
        TANT_QUE ((position_lievre < 10) ET (position_tortue < 10)) FAIRE
            DEBUT_TANT_QUE
            SI ( ALGOBOX_ALEA_ENT(1,6) < 6 ) ALORS
                DEBUT_SI
                position_tortue PREND_LA_VALEUR position_tortue + 1
                FIN_SI
            SINON
                DEBUT_SINON
                position_lievre PREND_LA_VALEUR position_lievre + 5
                FIN_SINON
            FIN_TANT_QUE
        SI (position_tortue==10) ALORS
            DEBUT_SI
            resultats[i] PREND_LA_VALEUR 1
            FIN_SI
        SINON
            DEBUT_SINON
            resultats[i] PREND_LA_VALEUR 0
            FIN_SINON
        FIN_POUR
    moyenne_empirique PREND_LA_VALEUR ALGOBOX_MOYENNE(resultats,1,n)
    marge PREND_LA_VALEUR 1/sqrt(n)
    AFFICHER "La proportion de courses gagnées par la tortue est "
    AFFICHER moyenne_empirique
    AFFICHER "Intervalle de confiance à 95% :"
    AFFICHER "[f - 1/sqrt(n), f + 1/sqrt(n)] = ["
    AFFICHERCALCUL moyenne_empirique - marge
    AFFICHER ", "
    AFFICHERCALCUL moyenne_empirique + marge
    AFFICHER "]. "
FIN_ALGORITHME
```

Remarque 1. L'affichage mêlant chaînes fixes et résultats calculés est assez long à écrire.

Dans les langages plus complexes on utilise plus volontiers une chaîne “gabarit” (c'est la fonction `printf` en C).

En pratique

Tester le code en ouvrant le fichier `lievre_tortue_long_estimation.alg`. On peut faire varier n pour voir (empiriquement) à partir de combien de répétitions la borne supérieure de l'intervalle de confiance est (souvent) en dessous de 0.5.

5.2 Solution mathématique

Changeons légèrement l'expérience : la tortue ne s'arrête plus à la case 10, mais continue indéfiniment, jusqu'à ce que le lièvre ait fait deux sauts. La tortue gagne le jeu d'origine si et seulement si, dans le jeu modifié, elle a atteint la case 10 avant l'arrêt du jeu. Si l'on appelle « succès » les 6 et « échecs » les autres résultats, il s'agit de compter le nombre d'échecs observés avant le deuxième succès dans une répétition d'expériences de Bernoulli. Ce nombre d'échecs X suit une loi classique appelée « loi binomiale négative », de paramètres $n = 2$ (le nombre de succès à atteindre) et $p = 1/6$ (la probabilité de succès). Pour tout entier k, l'événement $X = k$ correspond à obtenir $n-1$ succès et k échecs sur les $n+k-1$ premiers essais, et à réussir le $(n+k)^{\text{e}}$ essai. Il y a $\binom{n+k-1}{k}$ façons de répartir les k échecs, et une fois la répartition fixée la probabilité d'obtenir cette répartition est $p^n q^k$. On en déduit la loi de X :

$$\mathbb{P}[X = k] = \binom{n+k-1}{k} p^n q^k.$$

La tortue gagne si $X \geq 9$:

$$\mathbb{P}\left[\text{« la tortue gagne »}\right] = \mathbb{P}[X \geq 9] = 1 - F_X(8).$$

La fonction de répartition n'a pas d'expression agréable, mais les logiciels usuels permettent d'en donner une valeur approchée. En Scilab la fonction correspondante est `cdfnbn` (le nom de la fonction vient de « Cumulative Distribution Function for the Negative BiNomial ») [2]. Dans notre cas il faut taper `1 - cdfnbn("PQ",8,2,1/6,5/6)` , ce qui renvoie la valeur approchée $r \approx 0.4845167 < 1/2$.

6 Références techniques sur les logiciels utilisés

Ces trois logiciels sont libres et disponibles gratuitement sur les trois plateformes logicielles usuelles (Linux, MacOS et Windows).

6.1 Algobox

Le logiciel et de nombreux exemples, écrits pour illustrer les programmes de lycée, sont disponibles sur le site `http://www.xm1math.net/algobox/`.

2. En R c'est la fonction `pnbinom`. Il faut taper 1-pnbinom(8,2,1/6).

6.2 Scilab

Disponible sur `http://www.scilab.org/`, Scilab est un logiciel libre de calcul numérique, à peu près équivalent en termes de fonctionnalités à Matlab. Il dispose d'un « module lycée » développé pour faciliter son utilisation en classe. Un livret intitulé *Scilab pour l'enseignement des mathématiques* est téléchargeable gratuitement sur `http://www.scilab.org/fr/resources/documentation/tutorials`, page sur laquelle sont rassemblés de nombreux manuels en anglais et/ou en français.

6.3 R

Le logiciel libre R est spécialisé dans les statistiques. Il est disponible en téléchargement à l'adresse `http://www.r-project.org/`. Plusieurs tutoriels sont disponibles en anglais et/ou en français sur internet. L'installation de R contient plusieurs manuels en anglais dans le répertoire /doc/manual/. Une FAQ (Frequently Asked Questions) (Foire Aux Questions) est également incluse dans le répertoire /doc/html. Une version html est régulièrement mise à jour à l'adresse `http://cran.r-project.org/doc/FAQ/R-FAQ.html`. Enfin citons le tutoriel en français disponible à l'adresse `http://cran.r-project.org/doc/contrib/Paradis-rdebuts_fr.pdf`.