



Journées Nationales de l'APMEP

Partageons les mathématiques

METZ - 27 au 30 octobre 2012



Atelier P3-36

Marche aléatoire vers un paradoxe du calcul des probabilités

1 Notations et formules

(nombre de tous les chemins (s_0, s_1, \dots, s_n) avec $s_0 = 0$ et $s_n = x$) = $N_{n,x} = C_n^{\frac{n+x}{2}}$

$$P(S_n = x) = p_{n,x}$$

$$P(S_{2,\nu} = 0) = u_{2,\nu}$$

$$u_{2,\nu} = \frac{C_{2,\nu}^\nu}{2^{2,\nu}} \quad (1)$$

(probabilité que le premier passage à l'origine ait lieu à l'instant $2.n$) = $f_{2,n} = u_{2,n-2} - u_{2,n} = \frac{u_{2,n}}{2.n-1}$

$$P(S_1 \neq 0 \text{ et } \dots \text{ et } S_{2,n} \neq 0) = u_{2,n} \quad (2)$$

$$P(S_1 \geq 0 \text{ et } \dots \text{ et } S_{2,n} \geq 0) = u_{2,n} \quad (3)$$

(probabilité que la dernière visite à l'origine avant l'instant $2.n$ ait lieu à l'instant $2.k$ avec $k \leq n$) = $\alpha_{2,k,2,n} = u_{2,k} \cdot u_{2,n-2,k}$

$$u_{2,n} = f_2 \cdot u_{2,n-2} + f_4 \cdot u_{2,n-4} + \dots + f_{2,n} \cdot u_0 \quad (4)$$

(probabilité que, dans l'intervalle de temps de 0 jusqu'à $2,\nu$, la différence des scores soit positive pendant exactement $2.k$ unités de temps) = $b_{2,k,2,\nu} = \alpha_{2,k,2,\nu}$

2 Référence : JavaScript

GeoGebra 4, October 2011

Markus Hohenwarter, Michael Borchers

Méthode	Description
boolean evalCommand(String cmdString)	Évalue la chaîne comme si vous l'aviez entrée dans le champ de saisie de GeoGebra. ex : evalCommand('s = Droite[A, B]'). Vous pouvez utiliser plusieurs commandes en les séparant par "\n". Note: Utilisez le paramètre "language" pour vous assurer que la commande dans votre langue est reconnue.
void deleteObject(String objName)	Efface l'objet cité. Exemple : deleteObject('A').
void setValue(String objName, double value)	Affecte la valeur à l'objet cité. Exemple: setValue('n',36). Note: Si cet objet n'est pas un nombre, rien ne se passe. Si l'objet est un booléen, utiliser la valeur 1 pour le mettre à "Vrai", n'importe quelle valeur pour le mettre à "Faux".
void setColor(String objName, int red, int green, int blue)	Affecte la couleur RVB à l'objet. Exemple: setColor('A',200,100,100).
void setVisible(String objName, boolean visible)	Affiche (true) ou Cache (false) l'objet cité dans la vue Graphique. Exemple: setVisible('A',true).
void setLabelVisible(String objName, boolean visible)	Affiche (true) ou Cache (false) l'étiquette de l'objet cité dans la vue Graphique. Exemple: setLabelVisible('A',false).
void setLineThickness(String objName, int thickness)	Définit l'épaisseur de l'objet (de 1 à 13) (-1 pour "default")
void setPointSize(String objName, int size)	Définit la taille du point (de 1 à 9).
double getXcoord(String objName)	Retourne l'abscisse de l'objet cité. Exemple: getXcoord('A'). Note : Retourne 0 si l'objet n'est ni un point ni un vecteur.
double getYcoord(String objName)	Retourne l'ordonnée de l'objet cité. Exemple: getYcoord('u'). Note: Retourne 0 si l'objet n'est ni un point ni un vecteur.
double getValue(String objName)	Retourne la valeur de l'objet cité (longueur pour un segment, aire pour un polygone, etc). Exemple: getValue('poly1'). Si l'objet est un booléen, retourne 1 pour "Vrai", 0 pour "Faux". Note: Retourne 0 si l'objet n'a pas de valeur associée.
String getObject Type(String objName)	Retourne le type de l'objet cité sous forme de chaîne (comme "point", "line", "circle", etc. en Anglais !).
boolean exists(String objName)	Retourne un booléen attestant ou non de l'existence de l'objet cité dans la construction.
int getObjectNumber()	Retourne le nombre d'objets dans la construction.
String getObject Name(int n)	Retourne le nom du n-ième objet de la construction (attention, le compteur commence à 0).
void setCoordSystem(double xmin, double xmax, double ymin, double ymax)	Définit la fenêtre graphique. Exemple: setCoordSystem(-5,10,-1,20).

3 Référence : GGBScript

Point[Liste]	Convertit une liste contenant 2 nombres en un Point, par exemple : Point[{1, 2}]
Segment[Point A, Point B]	Segment [AB].
Histogramme[Liste Bornes Classes, Liste Hauteurs]	Histogramme avec les barres de hauteurs données. Les limites de classes déterminent la largeur et la position de chacune des barres de l'histogramme. i Exemple: Histogramme[{0,1,2,3,4,5},{2,6,8,3,1}]
Histogramme[Liste Bornes Classes, Liste Données]	Les bornes des classes déterminent la largeur et la position de chacune des barres de l'histogramme et sont utilisées pour déterminer combien des données appartiennent à chaque classe ; hauteur = (effectif de la classe) / (largeur de la classe).

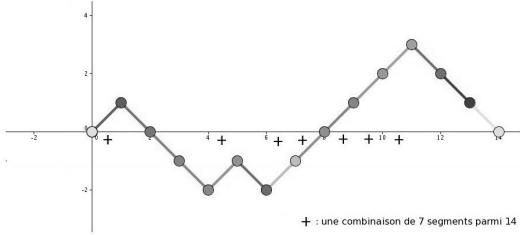


Figure pour les diapos n° 4 et 5

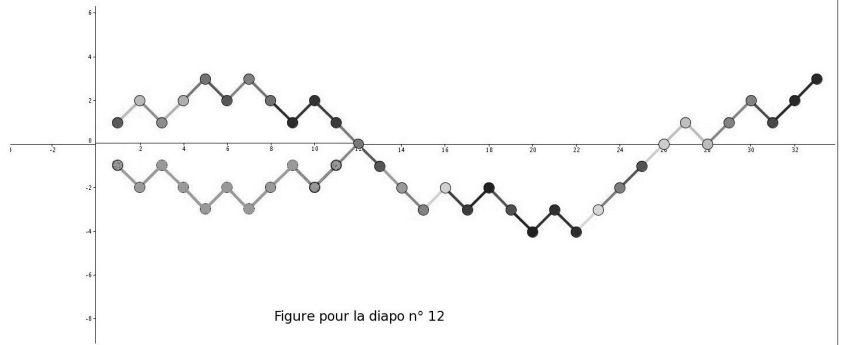


Figure pour la diapo n° 12

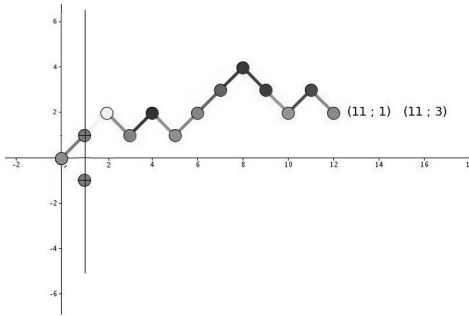
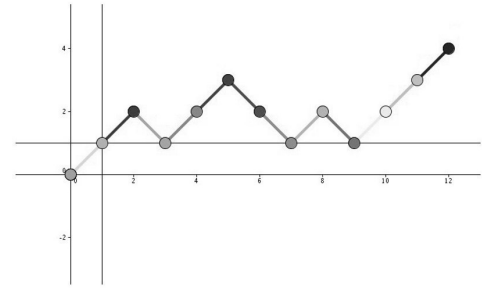
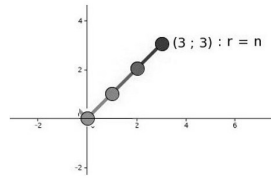


Figure pour la diapo n° 13



Figures pour la diapo n° 14

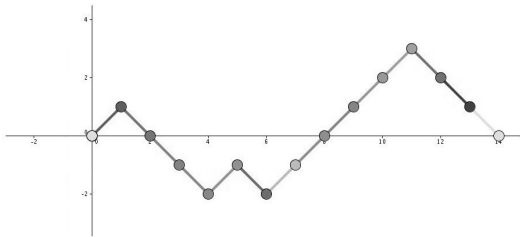


Figure pour la diapo n° 17

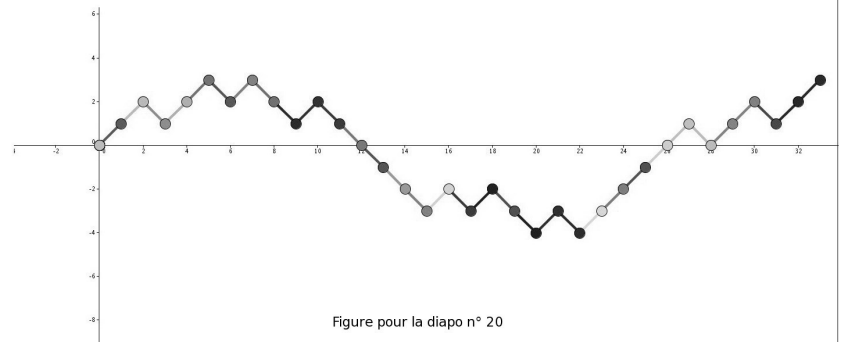


Figure pour la diapo n° 20