

Pour aller plus loin avec XCas : version 0.8.6 et 0.9

I. Remarque

Un programme XCas est une fonction ; il est donc possible de placer des variables dans son nom dont les valeurs seront passées au programme lors de son exécution :

Par exemple :

- on crée un programme qui doit simuler N lancers d'un dé et on le définit par : lancer_des(N) ; dans le code du programme N représente le nombre de lancers ;
- pour simuler 500 lancers du dé on tapera ensuite en ligne de commande : lancer_des(500).

Cette méthode évite d'avoir recours à l'instruction **saisir** qui ouvre ensuite une boîte de dialogue demandant la saisie de la valeur de la variable.

II. Utiliser des listes

On va illustrer ici la création d'un programme permettant de répondre au problème suivant :

On lance simultanément deux dés bien équilibrés.
 Quelles sont les fréquences obtenues pour la somme des nombres apparaissant sur la face supérieure des deux dés ?

La somme prend les valeurs de 2 à 12 ; nous allons utiliser une liste R pour stocker le nombre de fois où chacune des sommes apparaît lors de la simulation (R[2] contiendra le nombre d'occurrences de la somme 2).

Méthode de travail :

On déclare la variable R	local R;
On initialise la liste vide R (ceci a pour effet de donner à R le statut de liste)	R:=[];
Afin de pouvoir accéder aux éléments de la liste R, on les initialise à 0	pour I de 2 jusque 12 faire R[I] := 0; fpour

Exemple :

Le programme

Déclaration des variables listes →

Initialisation des listes vides →

Création des éléments de chaque liste →

```

Xcas Nouvelle Interface
Fich Edit Cfg Aide CAS Expression Cmds Prg Graphic Geo Tableur
/cygdrive/d/Jean-Louis/Jean-Louis/Maths-Info 2009-2010/Stages/Site/xcas/essai6.xws
? Sauver Config essai6.xws : exact real RAD 12 xcas 14.375M
1 Prog Edit Ajouter |11|      |nxt|  |OK (F9)|      |Save|
Somme_deux_des(N) := {
local S, I, J, R, F;
//N : nombre d'expériences
//S : somme des deux dés
//I, J, K : variables compteur
//R[i] : nombre de sommes ayant la valeur i
//F[i] : fréquence des sommes ayant la valeur i
R:=[];
F:=[];
pour I de 2 jusque 12 faire
  R[I] := 0;
  F[I] := 0;
fpour
  pour J de 1 jusque N faire
    S := floor(alea(0,6)+1)+floor(alea(0,6)+1);
    R[S] := R[S]+1;
  fpour
  pour K de 2 jusque 12 faire
    F[K] := R[K]/N;
    afficher("F("+K+") = "+round(F[K],2));
  fpour
}
;
    
```

Exécution du programme

```

2 Somme_deux_des(500)
F(2) = 0.03
F(3) = 0.05
F(4) = 0.07
F(5) = 0.12
F(6) = 0.13
F(7) = 0.18
F(8) = 0.15
F(9) = 0.1
F(10) = 0.08
F(11) = 0.06
F(12) = 0.03
    
```

III. Utiliser des outils graphiques dans un programme

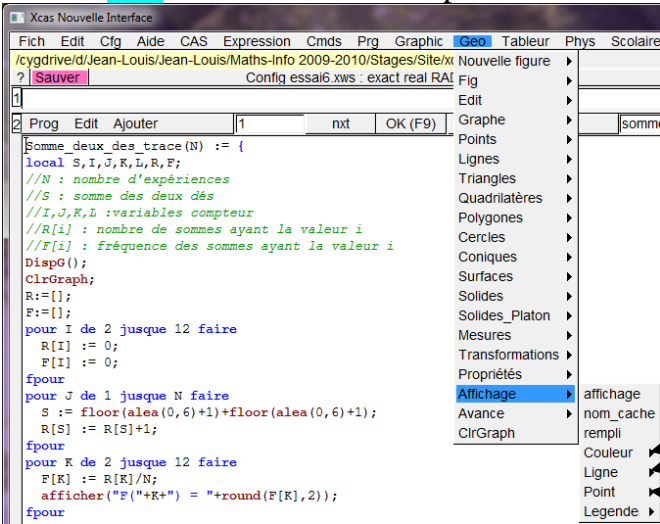
Les commandes de base :

- **DispG()** : ouvre la fenêtre graphique
- **ClrGraph** : efface la fenêtre graphique
- **affichage** : trace un objet géométrique avec des attributs de forme, couleur, ...

Les objets :

- point : **point**(abscisse,ordonnée)
- segment : **segment**(point(x_A, y_A),point(x_B, y_B))

Le bouton **Geo** de la barre de menus permet d'accéder à une aide concernant les objets dessinables :



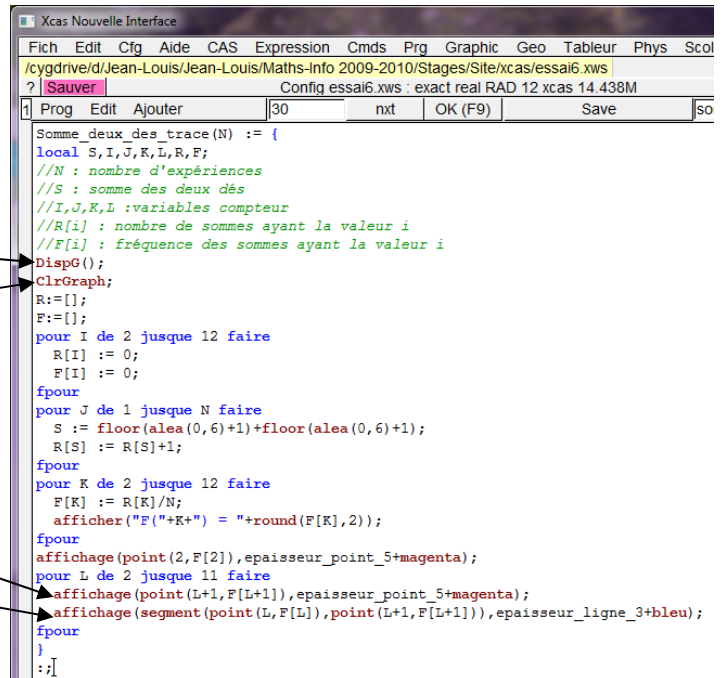
On va compléter le programme précédent afin d'obtenir une représentation graphique des fréquences

Ouverture de la fenêtre graphique

Effacement de la fenêtre graphique

Dessin d'un point de taille 5 et de couleur magenta

Dessin d'un segment d'épaisseur 3 et de couleur bleu



Lors de l'exécution du programme, la fenêtre graphique s'ouvre sur le bureau (elle peut éventuellement être cachée derrière la fenêtre principale de XCas).

Remarque : cette fenêtre ne se met pas automatiquement à l'échelle ; il est en général nécessaire d'utiliser le bouton **auto** pour y parvenir.



IV. Utiliser des sous programmes

On va illustrer ici la création d'un programme faisant appel à un autre programme pour répondre au problème suivant :

A l'aide d'une simulation, visualiser l'intervalle de fluctuation de la fréquence d'apparition de Pile dans E échantillons de N lancers d'une pièce bien équilibrée.

Étape 1 :

On crée un premier programme frequ(N) :
il calcule la fréquence de Pile lors de N lancers.

retourne : instruction qui fait sortir d'une fonction en renvoyant sa valeur.

```

1 Prog Edit Ajouter |12|      |nxt|  |OK (F9)|
frequ(N):={
//cette fonction calcule la frequence de Pile lors de N lancers
local I,X;
//I : numéro du lancer
//X : nombre de Pile
X := 0;
pour I de 1 jusque N faire
  si alea(0,1) <0.5 alors
    X := X+1;
  fsi
fpour
retourne (X/N);
};
    
```

On compile ce programme en cliquant sur OK.

Étape 2 :

On crée un second programme fluctuation(E,N) :
il simule l'exécution de E expériences de N lancers en utilisant le programme précédent.

Ouverture et nettoyage de la fenêtre graphique.

Appel du programme frequ(N).

Affichage de la fréquence et des bornes de l'intervalle de fluctuation.

Calcul du nombre de points dans l'intervalle de fluctuation.

```

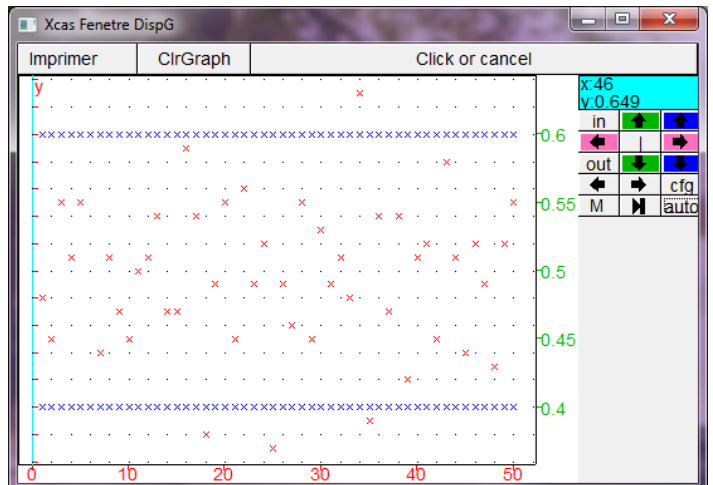
3 Prog Edit Ajouter |24|      |nxt|  |OK (F9)|
fluctuation(E,N):={
//ce programme simule l'execution de E experiences
//de N lancers d'une pièce
//E : nombre d'echantillons;
//N : nombre d'experiences dans chacun des echantillons
local V,S,J;
//J : numéro de l'échantillon examiné
//V : valeur de la fréquence dans l'échantillon examiné
//S : nombre de points dans l'intervalle de fluctuation
DispG();
ClrGraph;
S := 0;
pour J de 1 jusque E faire
  V := frequ(N);
  affichage(point(J,V),rouge);
  affichage(point(J,0.5-1/(sqrt(N))),bleu);
  affichage(point(J,0.5+1/(sqrt(N))),bleu);
  si (V >= 0.5-1/(sqrt(N)) et (V <= 0.5+1/(sqrt(N)))) alors
    S := S+1;
  fsi
fpour
afficher("Sur "+E+" echantillons, il y en a "+S+" pour
lesquels la frequence est dans l'intervalle de fluctuation,
soit : "+round(S/E,2)*100+" %");
};
    
```

Étape 3 :

Exécution du programme fluctuation(E,N)
(après l'avoir rendu exécutable).

```

A fluctuation(50,100)
Sur 50 echantillons, il y en a 46 pour lesquels
la frequence est dans l'intervalle de fluctuation, soit : 92.0 %
Evaluation time: 1.029
    
```



V. Programmer en ligne de commande

On désire comparer 10 séries de 50 échantillons ; cela peut se faire en tapant une instruction **pour** en ligne de commande.

```

5 pour K de 1 jusque 10 faire fluctuation(50,100);fpour;
Sur 50 echantillons, il y en a 48 pour lesquels la frequence est dans l'intervalle de fluctuation, soit : 96.0 %
Sur 50 echantillons, il y en a 50 pour lesquels la frequence est dans l'intervalle de fluctuation, soit : 100.0 %
Sur 50 echantillons, il y en a 49 pour lesquels la frequence est dans l'intervalle de fluctuation, soit : 98.0 %
Sur 50 echantillons, il y en a 50 pour lesquels la frequence est dans l'intervalle de fluctuation, soit : 100.0 %
Sur 50 echantillons, il y en a 48 pour lesquels la frequence est dans l'intervalle de fluctuation, soit : 96.0 %
Sur 50 echantillons, il y en a 50 pour lesquels la frequence est dans l'intervalle de fluctuation, soit : 100.0 %
Sur 50 echantillons, il y en a 49 pour lesquels la frequence est dans l'intervalle de fluctuation, soit : 98.0 %
Sur 50 echantillons, il y en a 47 pour lesquels la frequence est dans l'intervalle de fluctuation, soit : 94.0 %
Sur 50 echantillons, il y en a 48 pour lesquels la frequence est dans l'intervalle de fluctuation, soit : 96.0 %
Sur 50 echantillons, il y en a 47 pour lesquels la frequence est dans l'intervalle de fluctuation, soit : 94.0 %
Evaluation time: 12.324
    
```