Université des Sciences et Technologies de Lille
U.F.R. de Mathématiques Pures et Appliquées

Agrégation externe **Année 2003-2004**

Initiation à Scilab et LFGN

Après avoir lancé Scilab, visiter dans le menu `Demos` :

- `Introduction to Scilab` (sans s'attarder)
- `Graphics : Introduction`
- `Random`

Regarder ensuite dans le menu `Help` :

- `Graphic Library`
- `Utilities and Elementary Functions`

Tester la fenêtre `Apropos` du Help panel, d'abord en tapant « `cosinus` », puis « `cos` », puis « `cosi` ».

Dans `Help Scilab Programming`, regarder `dot(.)`. Tester les opérations ponctuelles sur les vecteurs et matrices. Dans `Utilities and Elementary Functions`, regarder et tester en recopiant les exemples : `input`, `zeros`, `sum`, `cumsum`, `prod`, `cumprod`. Dans `Help Scilab Programming`, regarder `function`. Ecrire une fonction simple, par exemple factorielle.

Ex 1. *Coefficients du binôme*
On se propose de calculer la liste des C_n^k, $0 \leq k \leq n$ par différentes méthodes.

1) Méthode bête avec les factorielles (utiliser la fonction définie ci-dessus).

2) Améliorer la méthode précédente en utilisant les simplifications de l'écriture avec factorielle : on utilisera directement `cumprod`, y compris avec un pas de -1. Tester avec $n = 168$ puis $n = 171$ en extrayant les 4 dernières valeurs de la liste [1]. Qu'en pensez vous ?

3) Méthode de fainéant rusé : utiliser la fonction de répartition de la loi binomiale de paramètres n et $1/2$. A cause des puissances de 2, il semble que l'on puisse aller jusqu'à $n = 1023$. Cependant, la méthode manque de fiabilité pour les grandes valeurs de k : comparer les valeurs obtenues pour C_{1000}^{350} et C_{1000}^{650}. On peut améliorer en utilisant la symétrie des coefficients du binôme et en faisant les calculs pour $k \leq n/2$.

4) Méthode la plus économique pour la machine et pour le programmeur : utilisez le triangle de Pascal ! Ainsi aucun calcul ne fera intervenir de nombre supérieur au plus grand des C_n^k à calculer. C'est donc beaucoup plus fiable que les méthodes précédentes. En plus le code est très court : il tient essentiellement en une ligne (non compris déclaration de fonction, initialisation et affectation de sortie). On peut aller jusqu'à $n = 1029$. Pour le voir, en appelant `coefbinA` ('A' pour additive) la fonction ainsi obtenue, taper

1. Bien sûr, vous aviez déjà pensé à tester avec des petites valeurs $n = 3, 4, 5, \ldots$

```
y=coefbinA(1029) ;max(y)
```
puis
```
y=coefbinA(1030) ;max(y)
```

Ex 2. *Entonnoirs déterministes pour la LFGN*
Selon l'inégalité de Hoeffding, si les X_k sont indépendantes, identiquement distribuées et s'il existe des constantes a et b telles que $P(a \leq X_1 \leq b) = 1$, alors

$$\forall \varepsilon > 0, \quad \mathbf{P}\Big(\Big| \frac{S_n - \mathbb{E}\, S_n}{n} \Big| \geq \varepsilon \Big) \leq 2 \exp\Big(-n \frac{2\varepsilon^2}{(b-a)^2} \Big). \tag{1}$$

On peut utiliser ce type d'inégalité pour étudier quantitativement les fluctuations asymptotiques de S_n/n autour de $\mathbb{E}\, X_1$. Pour simplifier, on suppose désormais $a = 0$ et $b = 1$.

1) Montrer que pout tout entier $N \geq 2$ et tout $\alpha > 1/2$,

$$\mathbf{P}\Big\{ \forall k > N, \ \Big| \frac{S_k}{k} - \mathbb{E}\, X_1 \Big| \leq \sqrt{\frac{\alpha \ln k}{k}} \Big\} \geq 1 - \frac{2}{2\alpha - 1} N^{1-2\alpha}. \tag{2}$$

2) Illustrer graphiquement cette inégalité en faisant tracer la ligne polygonale d'interpolation des S_k/k et les deux courbes encadrantes (l'entonnoir) correspondantes. On prendra d'abord pour X_i des variables uniformes sur $[0,1]$ puis des v.a. de Bernoulli de paramètres $p = 0,5$ puis $p = 0,1$.

Ex 3. *Jeu de pile ou face et lois singulières*
Les X_k sont des variables aléatoires de Bernoulli indépendantes de même paramètre p. La série $\sum_{k \geq 1} 2^{-k} X_k$ convergeant p.s., on note U sa somme. La loi de cette variable aléatoire U qui est donc une mesure de probabilité sur $[0,1]$ sera notée μ_p. On a vu en cours que $\mu_{1/2}$ est la loi uniforme sur $[0,1]$, que pour $p \neq 1/2$ les mesures μ_p sont singulières (et étrangères 2 à 2) et que pour $0 < p < 1$, la fonction de répartition F_p de μ_p est continue. Le but de cet exercice est de tracer une représentation graphique de F_p. On note $F_{p,n}$ la fonction de répartition de

$$U_n = \sum_{k=1}^{n} 2^{-k} X_k.$$

1) Expliciter et représenter graphiquement $F_{p,1}$ et $F_{p,2}$. On utilisera `plot2d2`.

2) Trouver un moyen simple pour obtenir de manière récursive la liste des masses ponctuelles de la loi de U_n, autrement dit les hauteurs de sauts de $F_{p,n}$.

3) Utiliser ceci pour tracer la représentation graphique de $F_{p,n}$. On écrira la suite des instructions dans un fichier `.sce`. Quelle valeur de n vous paraît raisonnable pour représenter F_p par $F_{p,n}$ à l'écran ?

4) Suggestion de prolongement : simuler un échantillon de la loi de U, tracer la f.d.r. empirique correspondante sur le même graphique que F_p et illustrer ainsi le théorème de Glivenko-Cantelli.