

Génération aléatoire par la GSL (GNU Scientific Library)

Compilation : le compilateur `gcc` doit être invoqué avec les options `-lgsl` et `-lgslcblas`. Selon le système, il est possible qu'il faille ajouter la localisation des fichiers d'en-tête par `-I /usr/lib/include` et/ou `-lm`. Quelques exemples :

```

2 //pour produire un fichier objet prog.o avec optimisation O2
gcc proc.c -O2 -lgsl -lgslcblas -I /usr/lib/include -lm
//pour produire un exécutable prog.exe
4 gcc -o prog.exe prog.c -lgsl -lgslcblas -I /usr/lib/include -lm

```

Syntaxe élémentaire :

```

#include <gsl/gsl_rng.h> // pour les générateurs
2 #include <gsl/gsl_randist.h> // pour les lois usuelles
#include <stdio.h> //pour l'affichage
4 #include <time.h> //pour lire l'heure
int main(void) {
6     // CREATION DU GENERATEUR:
    gsl_rng * Gege= gsl_rng_alloc(gsl_rng_mt19937);
8     // CHOIX DE LA GRAINE
    gsl_rng_set(Gege,2016);
10    //ou
    gsl_rng_set(Gege,time(NULL));
12
    // GENERATION D'UNE UNIFORME SUR [0,1]
14    double x=gsl_rng_uniform(Gege);

    // GENERATION D'UNE UNIFORME SUR 0,1,...,N-1:
16    int N=6;
18    int k=gsl_rng_uniform_int(Gege,N); //uniforme sur 0,1,...,5

    // GENERATION D'UNE V.A. DE LOI BINOMIALE(N,p)
    N=8; double p=0.25;
22    int u=gsl_ran_binomial(Gege,p,N);
    // AFFICHAGE DE QUELQUES RESULTATS:
24    printf("x=%f\n y=%f\n n=%d\n u=%d",x,y,n,u);

    // EFFACEMENT DU GENERATEUR (NE PAS OUBLIER !!!)
26    gsl_rng_free(Gege);
28    return 0;
}

```

Écrire et lire dans un fichier en langage C

Pour écrire ou lire dans un fichier, on adaptera l'exemple suivant :

```

#include <stdio.h>
2 int main(void) {
    double x=3.14, y=2.17; int n=2.16;
4 // Ouverture du fichier (w pour write)
    FILE * fichier_pour_ecrire = fopen("écriture.txt","w");
6 // Ecriture dans le fichier: \n pour aller à la ligne
    // %f pour les float, %lf pour les doubles, %d pour les entiers
8 fprintf(fichier_pour_ecrire,"%f %d toto \n", x,n);
    fprintf(fichier_pour_ecrire,"%f %d \n", y,n+1);
10 // Fermeture du fichier
    fclose(fichier_pour_ecrire);
12
    //Ouverture du fichier (r pour read)
14 FILE * fichier_pour_lire = fopen("lire.txt","r");
    // Stockage d'un double puis d'un entier dans x et n
16 fscanf(fichier_pour_lire,"%lf %d", &x,&n);
    //Fermeture du fichier
18 fclose(fichier_pour_lire);
    return 0;}

```

Utilisation de Gnuplot

- Dans un terminal, il faut aller dans le répertoire où sont les fichiers à visualiser puis entrer la commande `gnuplot`.
- Une fois dans `gnuplot`, représenter les données d'un fichier `exemple.txt` avec un nombre par ligne se fait par :

```
> plot "exemple.txt"
```

Les données seront représentées par des points. Pour avoir une ligne brisée, il faut taper :

```
> plot "exemple.txt" with lines
```

- Supposons qu'un fichier ait, sur toutes ses lignes, une suite de 4 valeurs séparées par des espaces. Pour représenter la quatrième valeur en fonction de la deuxième, on utilise :

```
> plot "exemple.txt" using 2:4 with lines
```

- On sort de `gnuplot` avec l'instruction `quit`.
- Une fois un graphique représenté, on peut zoomer en dessinant un rectangle avec le bouton *droit* de la souris enfoncé. Une icône de la barre de menu du graphique permet de revenir à la taille initiale.

Génération aléatoire par la GSL (GNU Scientific Library)

Compilation : le compilateur `gcc` doit être invoqué avec les options `-lgsl` et `-lgslcblas`. Selon le système, il est possible qu'il faille ajouter la localisation des fichiers d'en-tête par `-I /usr/lib/include` et/ou `-lm`. Quelques exemples :

```

//pour produire un fichier objet prog.o avec optimisation O2
2 gcc proc.c -O2 -lgsl -lgslcblas -I /usr/lib/include -lm
//pour produire un exécutable prog.exe
4 gcc -o prog.exe prog.c -lgsl -lgslcblas -I /usr/lib/include -lm

```

Syntaxe élémentaire :

```

#include <gsl/gsl_rng.h> // pour les générateurs
2 #include <gsl/gsl_randist.h> // pour les lois usuelles
#include <stdio.h> //pour l'affichage
4 #include <time.h> //pour lire l'heure
int main(void) {
6     // CREATION DU GENERATEUR:
    gsl_rng * Gege= gsl_rng_alloc(gsl_rng_mt19937);
8     // CHOIX DE LA GRAINE
    gsl_rng_set(Gege,2016);
10    //ou
    gsl_rng_set(Gege,time(NULL));
12
    // GENERATION D'UNE UNIFORME SUR [0,1]
14    double x=gsl_rng_uniform(Gege);

    // GENERATION D'UNE UNIFORME SUR 0,1,...,N-1:
16    int N=6;
18    int k=gsl_rng_uniform_int(Gege,N); //uniforme sur 0,1,...,5

    // GENERATION D'UNE V.A. DE LOI BINOMIALE(N,p)
    N=8; double p=0.25;
22    int u=gsl_ran_binomial(Gege,p,N);
    // AFFICHAGE DE QUELQUES RESULTATS:
24    printf("x=%f\n y=%f\n n=%d\n u=%d",x,y,n,u);

    // EFFACEMENT DU GENERATEUR (NE PAS OUBLIER !!!)
26    gsl_rng_free(Gege);
28    return 0;
}

```

Écrire et lire dans un fichier en langage C

Pour écrire ou lire dans un fichier, on adaptera l'exemple suivant :

```

#include <stdio.h>
2 int main(void) {
    double x=3.14, y=2.17; int n=2.16;
4 // Ouverture du fichier (w pour write)
    FILE * fichier_pour_ecrire = fopen("écriture.txt","w");
6 // Ecriture dans le fichier: \n pour aller à la ligne
    // %f pour les float, %lf pour les doubles, %d pour les entiers
8 fprintf(fichier_pour_ecrire,"%f %d toto \n", x,n);
    fprintf(fichier_pour_ecrire,"%f %d \n", y,n+1);
10 // Fermeture du fichier
    fclose(fichier_pour_ecrire);
12
    //Ouverture du fichier (r pour read)
14 FILE * fichier_pour_lire = fopen("lire.txt","r");
    // Stockage d'un double puis d'un entier dans x et n
16 fscanf(fichier_pour_lire,"%lf %d", &x,&n);
    //Fermeture du fichier
18 fclose(fichier_pour_lire);
    return 0;}

```

Utilisation de Gnuplot

- Dans un terminal, il faut aller dans le répertoire où sont les fichiers à visualiser puis entrer la commande `gnuplot`.
- Une fois dans `gnuplot`, représenter les données d'un fichier `exemple.txt` avec *un nombre par ligne* se fait par :

```
> plot "exemple.txt"
```

Les données seront représentées par des points. Pour avoir une ligne brisée, il faut taper :

```
> plot "exemple.txt" with lines
```

- Supposons qu'un fichier ait, sur toutes ses lignes, une suite de 4 valeurs séparées par des espaces. Pour représenter la quatrième valeur en fonction de la deuxième, on utilise :

```
> plot "exemple.txt" using 2:4 with lines
```

- On sort de `gnuplot` avec l'instruction `quit`.
- Une fois un graphique représenté, on peut zoomer en dessinant un rectangle avec le bouton *droit* de la souris enfoncé. Une icône de la barre de menu du graphique permet de revenir à la taille initiale.