

gsl: GNU scientific library : librairie mathématique

- But : générer des nombres pseudo-aléatoires dans $[0, 1[$

```
cd $HOME/projets1A_2021/histo
make clean; make all
```

```
# Makefile
CC  = gcc -g
OBJ  = main.o

LIBS = -lgsl -lgslcblas -lm
INCS =
BIN  = histo

all: $(BIN)

clean:
    rm -f $(OBJ) $(BIN)

$(BIN): $(OBJ)
    $(CC) $(INCS) $(OBJ) -o $(BIN) $(LIBS)

.c.o:
    $(CC) $(INCS) -c $<
```

Librairies de GSL

make all \Rightarrow lance la création du fichier exécutable aleat3

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <gsl/gsl_rng.h>
#include <gsl/gsl_histogram.h>
```

Fichiers header (include) contenant les prototypes des fonctions de GSL et des constantes prédéfinies

```
unsigned long int seed = 1564614431;
gsl_rng * r=NULL; /* generateur */
```

```
int main() {
/* initialisation */
const gsl_rng_type *T;
gsl_rng_env_setup();

T = gsl_rng_default;
r = gsl_rng_alloc (T);

cout << "Nom du generateur : "
      << gsl_rng_name (r) << endl;
gsl_rng_set(r, seed);
```

Un générateur de nombres pseudo-Aléatoires repose sur une suite Arithmétique complexe de nombres entiers. Cette suite est déterministe et ne dépend que de la valeur de la graine (seed). Autrement dit, on obtient toujours le même tirage avec la même graine.

Déclaration et initialisation du générateur

```
int N=10;
for (int n=0; n<N; n++) {
double aleat=gsl_rng_uniform(r);
printf("n : %d \t %lf\n", n, aleat);
}
```

Tirage de N nombres pseudo-aléatoires

```
}
```

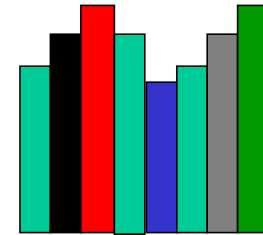
- But : générer des nombres pseudo-aléatoires dans $[0, 1[$ et les stocker dans un histogramme pour vérifier l'uniformité de la distribution

make clean; make all

```
int main()
{
...
// declaration et definition d'un histogramme gsl
int NBINS=20;
gsl_histogram * h = gsl_histogram_alloc(NBINS);
double borneinf=0.;
double bornesup=1.;
gsl_histogram_set_ranges_uniform(h, borneinf, bornesup);

int N=100000;
for (int n=0; n<N; n++) {
    double aleat=gsl_rng_uniform(r);
    gsl_histogram_increment(h, aleat);
}
FILE *fout=fopen("histo.out", "w");

for (int n=0; n<NBINS; n++){
    fprintf(fout, "%lf \t %lf\n" ,
        0.5*(h->range[n]+h->range[n+1]), h->bin[n] );
}
fclose(fout);
gsl_histogram_free (h);
}
```



NBINS : nb de barres

