Name: PORTILLO GALVAN Valentin

Algorithms

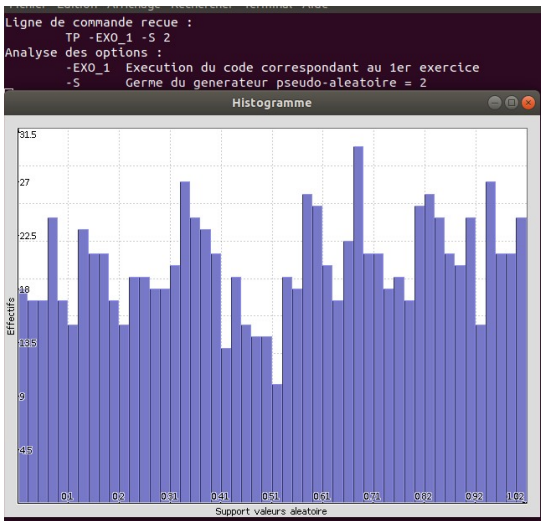
CS Fundamentals UVMSC-INF101A

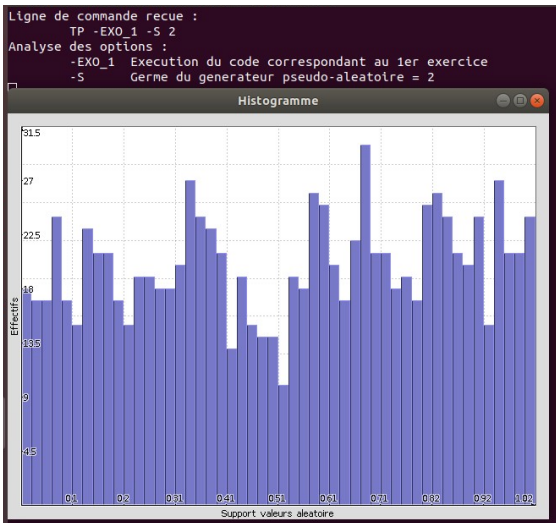
Laboratory 1: Randomness in algorithms

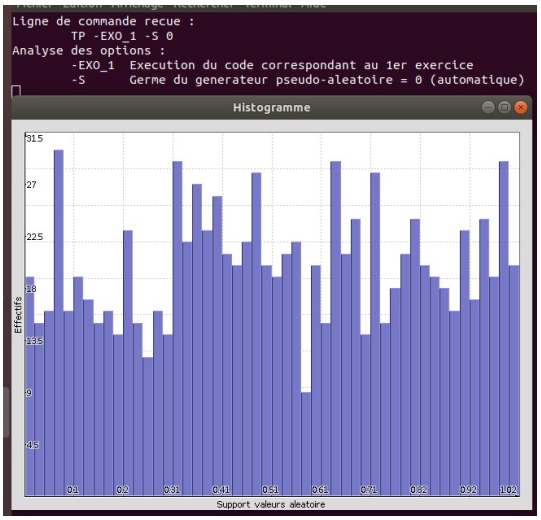
Télécom Bretagne – Département Image et Traitement de l'Information
October 2018

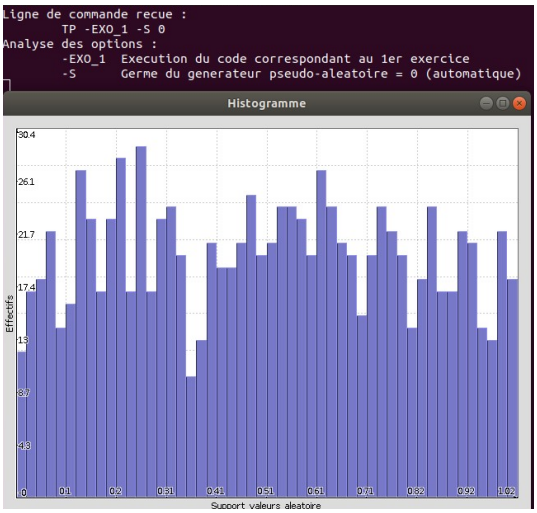
Q1. For which value of variable 'seed' an appropriate <u>random</u> distribution is obtained? Illustrate your analysis with maximum 4 images.

For the seed = 0, because it will dynamically assign itself a value, thus giving us a different distribution each time. If we use a number superior to 0, it will be fixed, and we're going to have same distribution each time.



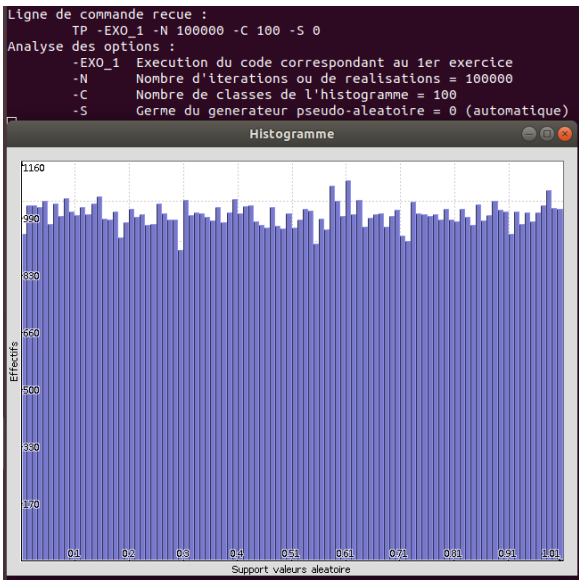


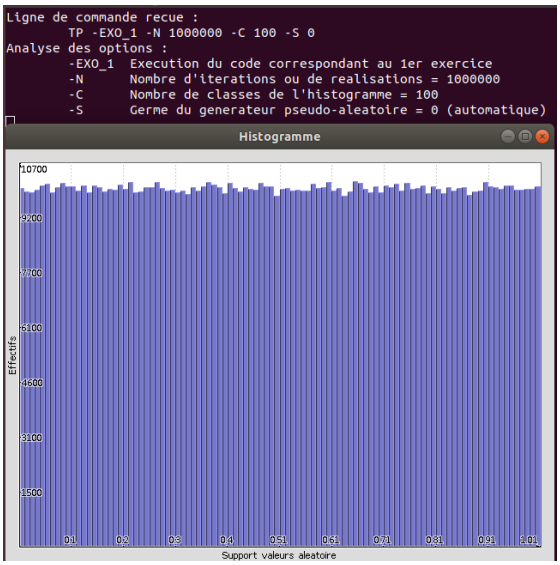


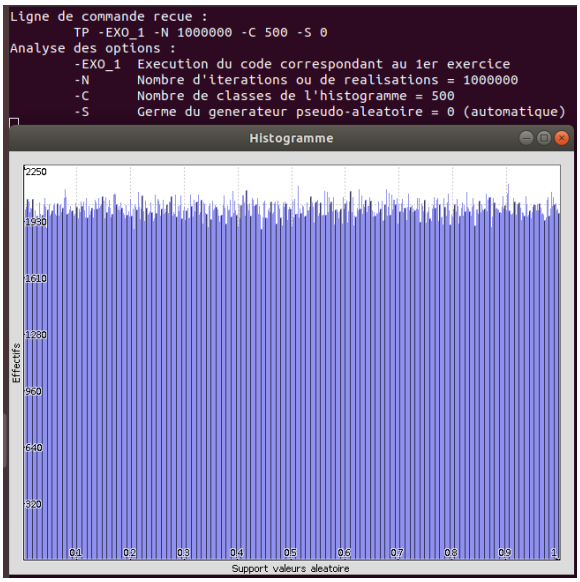


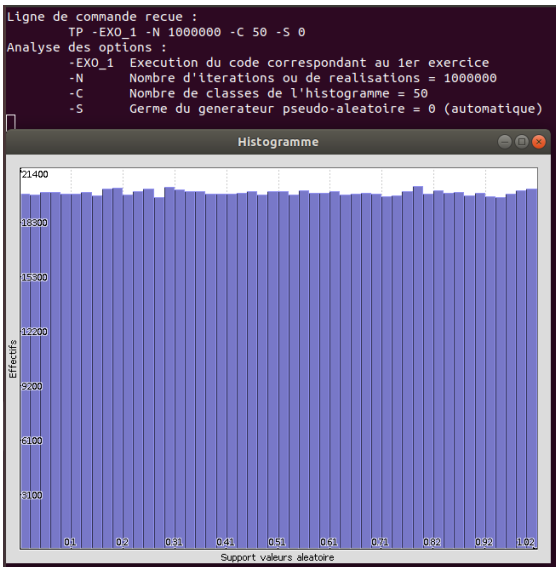
Q2. How does the uniform distribution shape change? For which values of ‘nbTir’ and ‘C’ an appropriate random uniform distribution is obtained? Illustrate your analysis with maximum 4 images.

The uniform distribution shape change to be more an horizontal line with a high nbTir and a C betzeen 50 and 100.





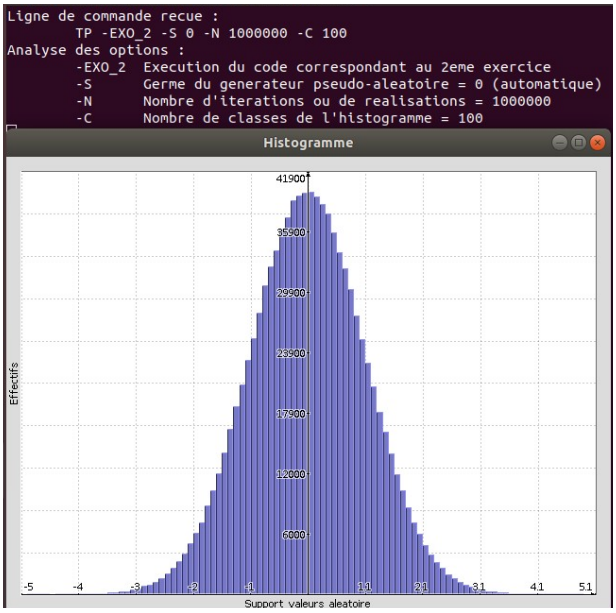


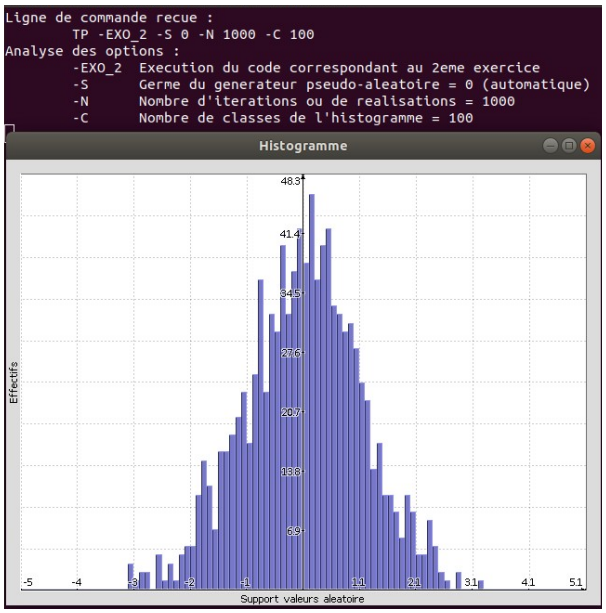


Q3. Under which conditions the generated distribution is likely to be different than the expected result? Under which conditions the

generated distribution is likely to be close to the expected result? Illustrate your answer using 2 images.

When we don't do a large quantity of experiments (-N) or a small or very large classes for the calculus (-C).





Q4. Which is the algorithm of the described Monte-Carlo method to estimate the value of π (i.e. the pseudocode to make function '`piMC`' work as expected)?

```
/* Input */

unsigned int seed;

int nbTir;

double valeurs[];

/* Output */

double pi;

/* Local Variables */

double x, y;

int in, all, I;

/* Initialize variables */

srand(seed);

in = all = 0;
```

```
/* begin */
for (i = 0; I < nbTir; i++) {
    x = assign random value between [0, 1];
    y = assign random value between [0, 1];
    if ((x² + y²) <= 1) {
        draw_the_point_with_inside_characteristics;
        i ++;
    } else {
        draw_the_point_with_outside_characteristics;
    }
    all ++;

    pi = 4 * (double) in / (double) all;
    valeurs[i] = pi;
}
return pi;
/* End */
```

Q5. Describe the influence of parameter ‘`nbTir`’ in the convergence of the algorithm and the obtained precision (be aware that the M_PI

constant of the standard library `math.h` assigns 3.141593 to π). Illustrate your answer using 2 images of the estimated value evolution.

As the nbTir increases the precision increases and there's a point where there is not a lot of difference. (nbTir = 5000)