CLERMONT AUVERGNE INP Isima

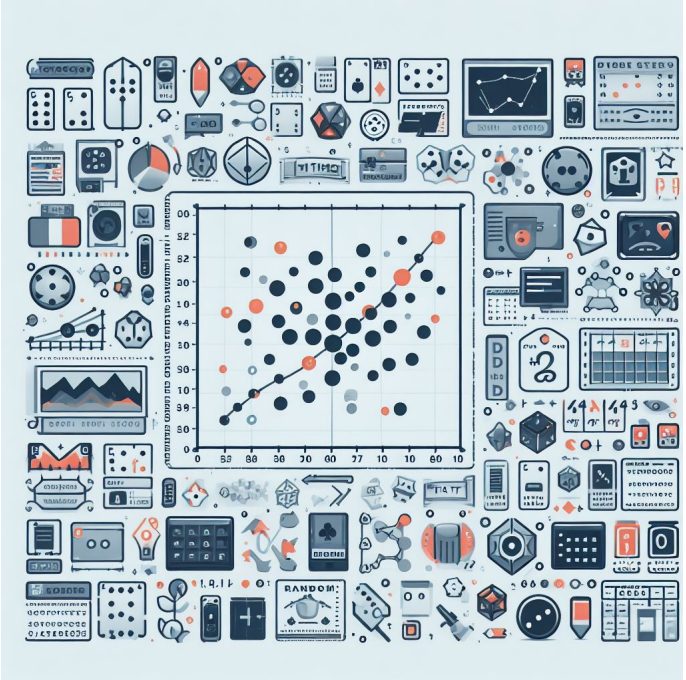
Cours de *Simulation*

Année Universitaire 2023-2024

ZZ2 promo 25

Chargée de TP : Hill David

Simulation de Monte Carlo & Intervalle de confiance



Dupois Thomas

Table des matières

Table des figures

1 Introduction

Le travaux pratique suivant vise à explorer les principes fondamentaux de la simulation stochastique et à mettre en œuvre la méthode de Monte-Carlo pour l'estimation de π.

L'objectif de cette étude est d'acquérir une compréhension approfondie des concepts de génération de nombres aléatoires, de l'estimation de la précision, de l'analyse statistique des résultats de simulation et de la création d'intervalles de confiance.

De plus, nous observerons l'utilisation de la méthode de Monte-Carlo et nous estimerons la valeur de π en utilisant des simulations basées sur des tirages aléatoires, et nous discuterons de la lenteur de la convergence de cette méthode.

Ensemble des questions

1.1 Approximer π avec la méthode de Monte Carlo

Afin d'approximer π avec la méthode de **Monte Carlo** on utilise la fonction **simPi()** suivante. Cette dernière va prendre en paramètre un nombre n de points.
Puis va générer n fois de manière aléatoire 2 nombres réels entre 0 et 1 qui représenteront les coordonnées x et y pour chacun des points. Si le point généré se trouve dans le cercle de rayon 1, alors on incrémente la variable **point_cercle** On effectue ensuite le rapport entre le nombre de points au sein du cercle (**point_cercle**) et le nombre de points total, on multipli ce rapport par 4 (car un seul quart de cercle ne suffit pas pour approximer π) et on retourne ce dernier devant être, si le nombre de points est suffisament grand, une approximation de π.

1.1.1 Visualisation graphique :

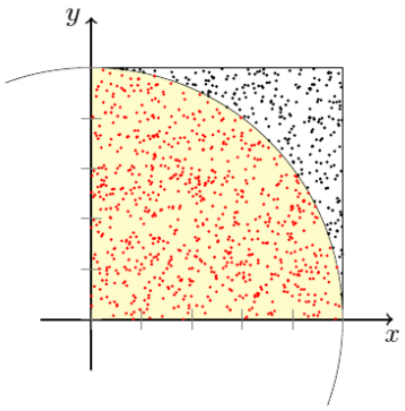
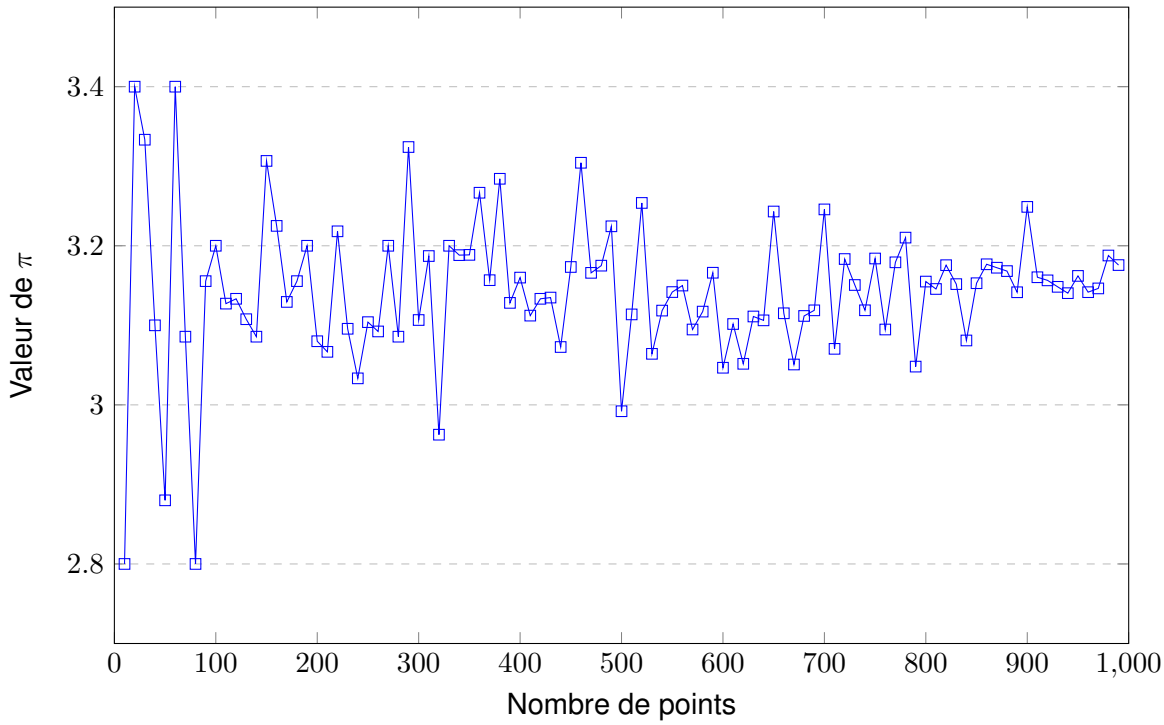


FIGURE 1 – Approximation de π par la méthode de Monte Carlo

1.1.2 code

```c
double simPi(int n)
{
    double x, y;
    int point_cercle = 0;
    for (int i = 0; i < n; i++)
    {
        x = genrand_real1();
        y = genrand_real1();
        if (x * x + y * y <= 1.0)
            point_cercle++;
    }
    return 4 * ((double) point_cercle / (double) n);
}
```

1.1.3 Convergence de π



Nous avons utilisé nos résultats afin de pouvoir observer la convergence de π grâce à la fonction **simPi()** sur les 1000 premiers points. On peut remarquer que malgré la fluctuation initiale des premiers points, on converge progressivement vers une valeur approximative de π.

1.1.4 output

```
----------------QUESTION 1---------------

Approximation de PI pour 1000 points:
3.124000
Approximation de PI pour 1_000_000 points:
3.144720
Approximation de PI pour 1_000_000_000 points:
3.141541
```

On rappel que $\pi \approx 3,14159265358979323846$,
Ainsi on constate bien que plus le nombre de points générés est grand, plus on s'approche de la valeur de π.
De plus, la méthode de Monte-Carlo pour estimer π a une convergence relativement lente, car elle dépend de la racine carrée du nombre total de points générés.
Cela signifie que pour augmenter la précision de l'estimation, nous sommes contraint de générer un nombre de points de plus en plus grand.

Selon les résultats, pour un certain nombre de points, on obtient la précision de π suivante :

Nombre de points	Précision
1000	0.1 [dixième]
1_000_000	0.01[centième]
1_000_000_000	0.0001 [cent-millième]

1.2 Expériences indépendantes et moyenne de l'approximation de π

Nous allons à présent aborder une approche légèrement différente pour approximer π. En effet, nous allons désormais utiliser le biais d'expériences indépendantes, c'est à dire que nous allons reproduire l'expérience précédente avec un nombre suffisant de points, à la différence que nous ne feront plus une, mais au moins 10 expériences dont chaque approximation de π sera stocker dans un tableau (l'approximation de la i^{eme} expérience sera sauvegardé dans la i^{eme} case du tableau).
Ensuite, après avoir stocker l'ensemble des valeurs des approximations de π on calcule la moyenne que l'on retourne.

La fonction suivante exerce **n** expériences indépendantes et stocke leur résultat dans un tableau de **n** cases.

```c
double * estimated_PIs(int n, int n2)
{
    double * tab = (double*) malloc(n * sizeof(double));
    for (int i = 0; i < n; i++)
        tab[i] = simPi(n2);
    return tab;
}
```

Tandis que cette fonction calcule la moyenne du tableau qu'on lui donne en paramètre.

```c
double moyenne_PI(double * tab, int size)
{
    double total = 0.0;
    for (int i = 0; i < size; i++)
        total = total + tab[i];
    return total / size;
}
```

1.2.1 Output

```
----------------QUESTION 2----------------

Estimation de PI pour 10 expériences indépendantes et 1000 points à chaque
    ↪ expérience:
3.124400
Erreur absolu:
0.017193
Erreur relative:
0.994527
Estimation de PI pour 10 expériences indépendantes et 1000000 points à
    ↪ chaque expérience:
3.142208
Erreur absolu:
0.000616
Erreur relative:
1.000196
Estimation de PI pour 10 expériences indépendantes et 1000000000 points à
    ↪ chaque expérience:
3.141569
Erreur absolu:
0.000024
Erreur relative:
0.999992
```

On peut constater que plus le nombre de points augmente, plus l'erreur absolue diminue, plus l'erreur relative se rapproche de 1, et bien évidement, plus la moyenne des approximations de π contenus dans le tableau tends vers la valeur réelle de π, ce qui signifie que les estimations deviennent plus précises avec un échantillon plus grand.
L'erreur relative tends vers 1 car elle est déterminée par la proportion de l'erreur absolue par rapport à la valeur réelle de la grandeur mesurée. Ainsi, à mesure que l'erreur absolue diminue, en se rapprochant de zéro, parallèlement à une estimation plus précise de la valeur réelle, l'erreur relative se rapprochera de 1. Cela indique que la mesure s'approche de la valeur réelle de la grandeur étudiée, ce qui est essentiel pour évaluer la précision de l'estimation.
On pourra noter que l'erreur absolue est plus importante avec un nombre de point plus faible, cette observation permet de prendre conscience des erreurs de simulation si notre nombre d'expériences est trop faible, et que malgré l'utilisation de multiple expérience aléatoire, il reste tout de même une certaine marge d'erreur.

1.3 Calcul des intervalles de confiance autour de la moyenne simulée

Rappelons qu'un intervalle de confiance permet l'encadrement une valeur réelle que l'on cherche à estimer à l'aide de mesures prises par un procédé aléatoire. Ainsi, à partir d'un relevé exhaustif de valeur, on peut définir une marge d'erreur entre les résultats.

On cherche à définir un intervalle de confiance après l'obtention de différente approximation de la valeur de π. Le calcul de se dernier est assez trivial. En effet, après avoir réalisé un certain nombre d'expériences indépendantes en ayant stocké leur résultat dans un tableau puis en ayant déduis la moyenne de ce dernier, on calcule une approximation de la variance :

$$S^2(n) = \frac{\sum_{i=1}^{n}[X_i - \overline{X}(n)]^2}{n-1} \tag{1}$$

1.3.1 code

```c
double variance_calc(double * tab, double moyenne, int size)
{
    double somme = 0.0;
    for (int i = 0; i < size; i++)
        somme = somme + ((tab[i] - moyenne) * (tab[i] - moyenne));
    return somme / (size - 1);
}
```

Puis, une fois la variance calculée, on peut utiliser la formule pour obtenir le rayon de confiance :

$$R = t_{n-1,1-\frac{\alpha}{2}} \times \sqrt{\frac{S^2(n)}{n}} \tag{2}$$

```c
double r_calc(double t, double variance, double n)
{
    return t * (sqrt(variance / n));
}
```

La variable inconnue "t" permet en réalité de corriger en partie l'erreur de calcul dû au fait que nous n'utilisons pas directement la variance mais son approximation.
Comme les simulations pour estimer π impliquent souvent un nombre fini d'expériences, la distribution de Student peut aider à évaluer la fiabilité de ces estimations en prenant en compte la variabilité des données et la taille limitée de l'échantillon.
Nous disposons du tableau suivant, avec un niveau de confiance α de 0.05 :

Table 1: Values of $t_{n-1,1-\alpha/2}$ of a Student law starting with $\alpha = 0.05$ depending on *n* experiments.

1≤n≤10	$t_{n-1,1-\alpha/2}$	11≤n≤20	$t_{n-1,1-\alpha/2}$	21≤n≤30	$t_{n-1,1-\alpha/2}$	n>30	$t_{n-1,1-\alpha/2}$
1	12.706	11	2.201	21	2.080	40	2.021
2	4.303	12	2.179	22	2.074	80	2.000
3	3.182	13	2.160	23	2.069	120	1.980
4	2.776	14	2.145	24	2.064	+∞	1.960
5	2.571	15	2.131	25	2.060		
6	2.447	16	2.120	26	2.056		
7	2.365	17	2.110	27	2.052		
8	2.308	18	2.101	28	2.048		
9	2.262	19	2.093	29	2.045		
10	2.228	20	2.086	30	2.042		

FIGURE 2 – Tableau des valeurs de t de la loi de Student avec α = 0.5

Sachant que l'intervalle de confiance est définit comme ceci : $[\overline{X} - R, \overline{X} + R]$ On peut utiliser la fonction suivante qui ne fait qu'appliquer les formules étudiées précédement :

```
1  void intervalle_confiance(int experiment, double t, int point)
2  {
3      double * tab = estimated_PIs(experiment, point);
4      double moyenne = moyenne_PI(tab, experiment);
5      double var = variance_calc(tab, moyenne, experiment);
6      double r = r_calc( t, var, experiment);
7      printf("L'intervalle de confiance obtenu avec %d experiences
       ↪  independantes et %d points pour chaque tirage est:\n\n", experiment,
       ↪  point);
8      printf("Moyenne:%lf\n", moyenne);
9      printf("Variance:%lf\n", var);
10     printf("R = %lf\n", r);
11     printf("Intervalle de confiance: [%lf;%lf]\n", moyenne-r, moyenne+r);
12     free(tab);
13 }
```

Nous avons utilisé la fonction précédente pour un nombre de points variable entre 1000 et 1_000_000 et un nombre d'expériences indépendantes allant jusqu'à 40 :

1.3.2 Output

```
----------------QUESTION 3----------------

Intervalle de confiance avec niveau de confiance alpha=0.05
L'intervalle de confiance obtenu avec 10 experiences independantes et
    ↪ 1000000 points pour chaque tirage est:
Moyenne:3.141684
Variance:0.000001
R = 0.000575
Intervalle de confiance: [3.141109;3.142259]

L'intervalle de confiance obtenu avec 20 experiences independantes et
    ↪ 1000000 points pour chaque tirage est:
Moyenne:3.141555
Variance:0.000003
R = 0.000797
Intervalle de confiance: [3.140758;3.142352]

L'intervalle de confiance obtenu avec 30 experiences independantes et
    ↪ 1000000 points pour chaque tirage est:
Moyenne:3.141674
Variance:0.000003
R = 0.000660
Intervalle de confiance: [3.141014;3.142334]

L'intervalle de confiance obtenu avec 40 experiences independantes et
    ↪ 1000000 points pour chaque tirage est:
Moyenne:3.141713
Variance:0.000003
R = 0.000555
Intervalle de confiance: [3.141157;3.142268]
```

On peut constater que les résultats de nos expériences sont très interessants, en effet, plus le nombre de points et d'expériences est élevé, plus l'intervalle de confiance se réduit tout en encadrant la valeur de π. De plus, malgré la variation des moyennes entre les expériences, l'intervalle de confiance reste relativement stable à mesure que la taille de l'échantillon augmente, ce qui indique une certaine fiabilité dans les résultats ci-dessus.

Après quelques recherches, on peut trouver le tableau suivant :

LOI DE STUDENT AVEC k DEGRÉS DE LIBERTÉ
QUANTILES D'ORDRE $1 - \gamma$

	γ										
k	0.25	0.20	0.15	0.10	0.05	0.025	0.010	0.005	0.0025	0.0010	0.0005
1	1.000	1.376	1.963	3.078	6.314	12.71	31.82	63.66	127.3	318.3	636.6
2	0.816	1.061	1.386	1.886	2.920	4.303	6.965	9.925	14.09	22.33	31.60
3	0.765	0.978	1.250	1.638	2.353	3.182	4.541	5.841	7.453	10.21	12.92
4	0.741	0.941	1.190	1.533	2.132	2.776	3.747	4.604	5.598	7.173	8.610
5	0.727	0.920	1.156	1.476	2.015	2.571	3.365	4.032	4.773	5.893	6.869
6	0.718	0.906	1.134	1.440	1.943	2.447	3.143	3.707	4.317	5.208	5.959
7	0.711	0.896	1.119	1.415	1.895	2.365	2.998	3.499	4.029	4.785	5.408
8	0.706	0.889	1.108	1.397	1.860	2.306	2.896	3.355	3.833	4.501	5.041
9	0.703	0.883	1.100	1.383	1.833	2.262	2.821	3.250	3.690	4.297	4.781
10	0.700	0.879	1.093	1.372	1.812	2.228	2.764	3.169	3.581	4.144	4.587
11	0.697	0.876	1.088	1.363	1.796	2.201	2.718	3.106	3.497	4.025	4.437
12	0.695	0.873	1.083	1.356	1.782	2.179	2.681	3.055	3.428	3.930	4.318
13	0.694	0.870	1.079	1.350	1.771	2.160	2.650	3.012	3.372	3.852	4.221
14	0.692	0.868	1.076	1.345	1.761	2.145	2.624	2.977	3.326	3.787	4.140
15	0.691	0.866	1.074	1.341	1.753	2.131	2.602	2.947	3.286	3.733	4.073
16	0.690	0.865	1.071	1.337	1.746	2.120	2.583	2.921	3.252	3.686	4.015
17	0.689	0.863	1.069	1.333	1.740	2.110	2.567	2.898	3.222	3.646	3.965
18	0.688	0.862	1.067	1.330	1.734	2.101	2.552	2.878	3.197	3.610	3.922
19	0.688	0.861	1.066	1.328	1.729	2.093	2.539	2.861	3.174	3.579	3.883
20	0.687	0.860	1.064	1.325	1.725	2.086	2.528	2.845	3.153	3.552	3.850
21	0.686	0.859	1.063	1.323	1.721	2.080	2.518	2.831	3.135	3.527	3.819
22	0.686	0.858	1.061	1.321	1.717	2.074	2.508	2.819	3.119	3.505	3.792
23	0.685	0.858	1.060	1.319	1.714	2.069	2.500	2.807	3.104	3.485	3.767
24	0.685	0.857	1.059	1.318	1.711	2.064	2.492	2.797	3.091	3.467	3.745
25	0.684	0.856	1.058	1.316	1.708	2.060	2.485	2.787	3.078	3.450	3.725
26	0.684	0.856	1.058	1.315	1.706	2.056	2.479	2.779	3.067	3.435	3.707
27	0.684	0.855	1.057	1.314	1.703	2.052	2.473	2.771	3.057	3.421	3.690
28	0.683	0.855	1.056	1.313	1.701	2.048	2.467	2.763	3.047	3.408	3.674
29	0.683	0.854	1.055	1.311	1.699	2.045	2.462	2.756	3.038	3.396	3.659
30	0.683	0.854	1.055	1.310	1.697	2.042	2.457	2.750	3.030	3.385	3.646
40	0.681	0.851	1.050	1.303	1.684	2.021	2.423	2.704	2.971	3.307	3.551
50	0.679	0.849	1.047	1.299	1.676	2.009	2.403	2.678	2.937	3.261	3.496
60	0.679	0.848	1.045	1.296	1.671	2.000	2.390	2.660	2.915	3.232	3.460
80	0.678	0.846	1.043	1.292	1.664	1.990	2.374	2.639	2.887	3.195	3.416
100	0.677	0.845	1.042	1.290	1.660	1.984	2.364	2.626	2.871	3.174	3.390
120	0.677	0.845	1.041	1.289	1.658	1.980	2.358	2.617	2.860	3.160	3.373
∞	0.674	0.842	1.036	1.282	1.645	1.960	2.326	2.576	2.807	3.090	3.291

FIGURE 3 – Loi de Student

Nous pouvons donc effectuer les mêmes tests mais avec un niveau de confiance plus important.

```
Intervalle de confiance avec niveau de confiance alpha=0.01
L'intervalle de confiance obtenu avec 10 experiences independantes et
    ↪ 1000000 points pour chaque tirage est:
Moyenne:3.141508
Variance:0.000002
R = 0.001207
Intervalle de confiance: [3.140301;3.142715]

L'intervalle de confiance obtenu avec 20 experiences independantes et
    ↪ 1000000 points pour chaque tirage est:
Moyenne:3.141853
Variance:0.000003
R = 0.000960
Intervalle de confiance: [3.140893;3.142813]

L'intervalle de confiance obtenu avec 30 experiences independantes et
    ↪ 1000000 points pour chaque tirage est:
Moyenne:3.141446
Variance:0.000003
R = 0.000720
Intervalle de confiance: [3.140726;3.142167]

L'intervalle de confiance obtenu avec 40 experiences independantes et
    ↪ 1000000 points pour chaque tirage est:
Moyenne:3.141328
Variance:0.000002
R = 0.000551
Intervalle de confiance: [3.140777;3.141879]
```

```
Intervalle de confiance avec niveau de confiance alpha=0.0005
L'intervalle de confiance obtenu avec 10 experiences independantes et
    1000000 points pour chaque tirage est:
Moyenne:3.142785
Variance:0.000003
R = 0.002469
Intervalle de confiance: [3.140316;3.145254]

L'intervalle de confiance obtenu avec 20 experiences independantes et
    1000000 points pour chaque tirage est:
Moyenne:3.141176
Variance:0.000005
R = 0.001964
Intervalle de confiance: [3.139212;3.143140]

L'intervalle de confiance obtenu avec 30 experiences independantes et
    1000000000 points pour chaque tirage est:
Moyenne:3.141603
Variance:0.000000
R = 0.000035
Intervalle de confiance: [3.141568;3.141639]

L'intervalle de confiance obtenu avec 40 experiences independantes et
    1000000000 points pour chaque tirage est:
Moyenne:3.141586
Variance:0.000000
R = 0.000023
Intervalle de confiance: [3.141563;3.141609]
```

Observons ces résultats au sein du tableau suivant :

Nombre d'expériences	$\alpha = 0.05$	$\alpha = 0.01$	$\alpha = 0.0005$
10	[3.141109 ;3.142259]	[3.140301 ;3.142715]	[3.140316 ;3.145254]
20	[3.140758 ;3.142352]	[3.140893 ;3.142813]	[3.139212 ;3.143140]
30	[3.141014 ;3.142334]	[3.140726 ;3.142167]	[3.141568 ;3.141639]
40	[3.141157 ;3.142268]	[3.140777 ;3.141879]	[3.141563 ;3.141609]

On constate que pour un même nombre d'expériences, l'intervalle de confiance varie selon le niveau de confiance utilisé, plus ce dernière est élevé, plus l'intervalle est réduit. Ce qui sous-entend que le rayon de confiance diminue lorsque niveau de confiance augmente.

2 Conclusion

En conclusion il est crucial de reconnaître l'importance d'utiliser des générateurs de nombres aléatoires fiables et efficaces tels que le Mersenne Twister, qui garantissent la robustesse et la qualité des simulations scientifiques. De plus, l'utilisation de la méthode Monte Carlo pour estimer des valeurs telles que π offre une précision accrue avec un nombre croissant de points de simulation, bien que la convergence de la méthode puisse être lente.

En ce qui concerne le calcul d'expériences indépendantes et l'obtention de la moyenne, l'initialisation appropriée du générateur de nombres aléatoires est essentielle pour garantir l'indépendance des expériences. L'évaluation des erreurs absolues et relatives par rapport à la constante π confirme la fiabilité et la précision de la méthode.

L'évaluation des intervalles de confiance autour de la moyenne simulée révèle une réduction significative du rayon de confiance à mesure que le nombre de répliques augmente, illustrant l'amélioration des résultats et la diminution de la marge d'erreur..

Enfin, le théorème central limite souligne l'importance de la taille de l'échantillon dans l'obtention de distributions de moyennes d'échantillons normaux, indépendamment de la distribution des données d'origine, garantissant ainsi des résultats fiables pour diverses applications scientifiques. Il est crucial de prendre en compte ces considérations lors de la mise en œuvre de simulations Monte Carlo pour garantir des résultats précis et fiables dans divers contextes scientifiques et techniques.

3 Référence

Tableau de la loi Student pour alpha = 0.0005 : https ://archimede.mat.ulaval.ca/stt1920/STT-1920-Loi-de-Student.pdf