

TP Probabilités

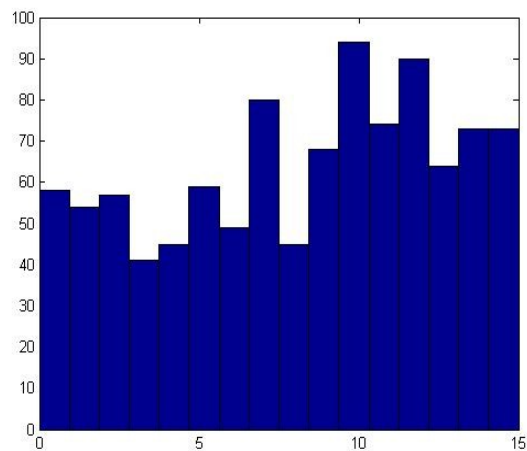
Génération de nombres aléatoires

B3334 : VICENTE Olivier, SORIN Nicolas

2.1) Test visuel

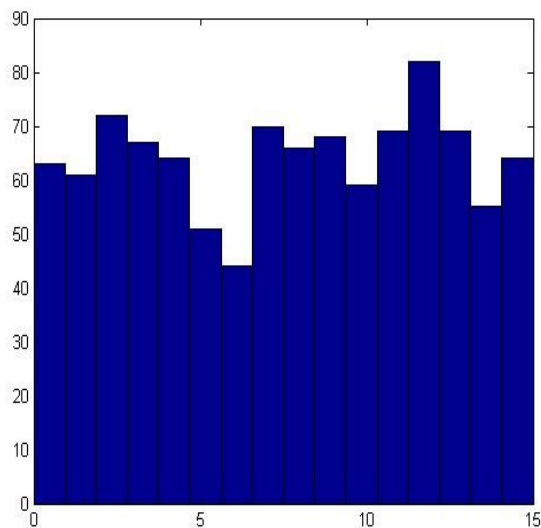
A partir de séquences de valeurs générées en utilisant notre programme, nous avons obtenu un histogramme avec Matlab pour chaque fonction :

Rand du C (poids faible) :



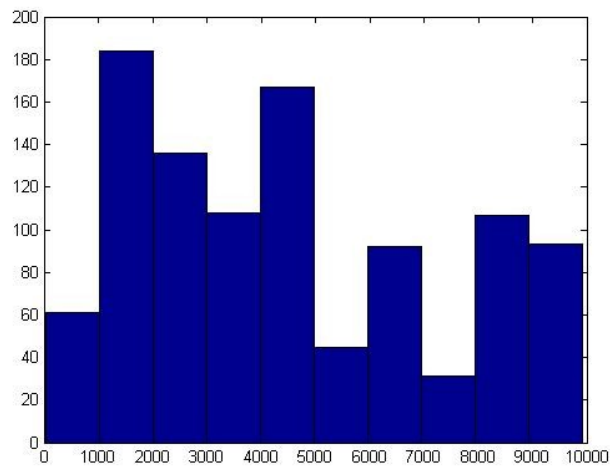
La répartition semble globalement homogène. Il faudra des tests plus précis.

Rand du C (poids fort) :



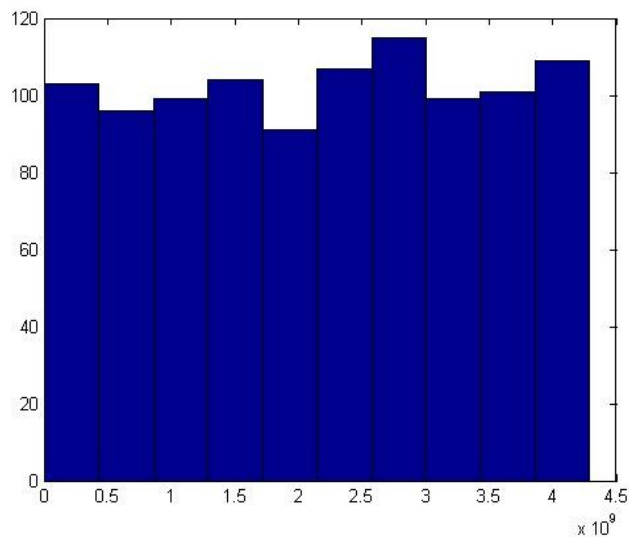
La répartition semble globalement homogène. Il faudra des tests plus précis.

Von Neumann :



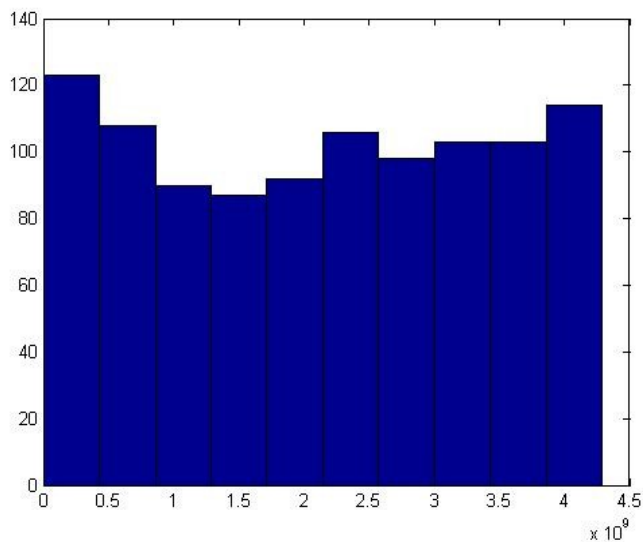
La répartition est loin d'être homogène. Ce test n'est donc pas fiable.

Mersenne-Twister :



La répartition semble bien homogène.

AES :



La répartition semble bien homogène.

2.2) Test de fréquence monobit

Nous avons ensuite effectué un test de fréquence monobit, qui permet de tester la proportion de 0 et de 1 dans une séquence. Cette proportion doit donc logiquement s'approcher de 50% pour chaque. Nous avons fait une moyenne des pvaleurs obtenues pour chaque fonction (en utilisant 20 résultats différents à chaque fois).

Rand du C faible : 0.458105

Rand du C fort : 0.422644

Von Neumann : 0.000000 → Cet algorithme est mauvais

Mersenne-Twister : 0.573077

AES : 0.433748

À part pour Von Neumann, les moyennes sont proches de 0,5 et on obtient des résultats qui fluctuent ce qui indique des distributions suivant une loi de probabilités uniforme sur $[0;1]$. Ainsi, on peut dire que les algorithmes sont efficaces.

2.3) Test des runs

Nous avons ensuite effectué un test des runs, qui permet de tester la proportion de cas où deux bits consécutifs ont la même valeur. Nous avons fait une moyenne des pvaleurs obtenues pour chaque fonction (en utilisant 20 résultats différents à chaque fois).

Rand du C faible : 0.502564

Rand du C fort : 0.466564

Von Neumann : 0.000000 → Cet algorithme est mauvais

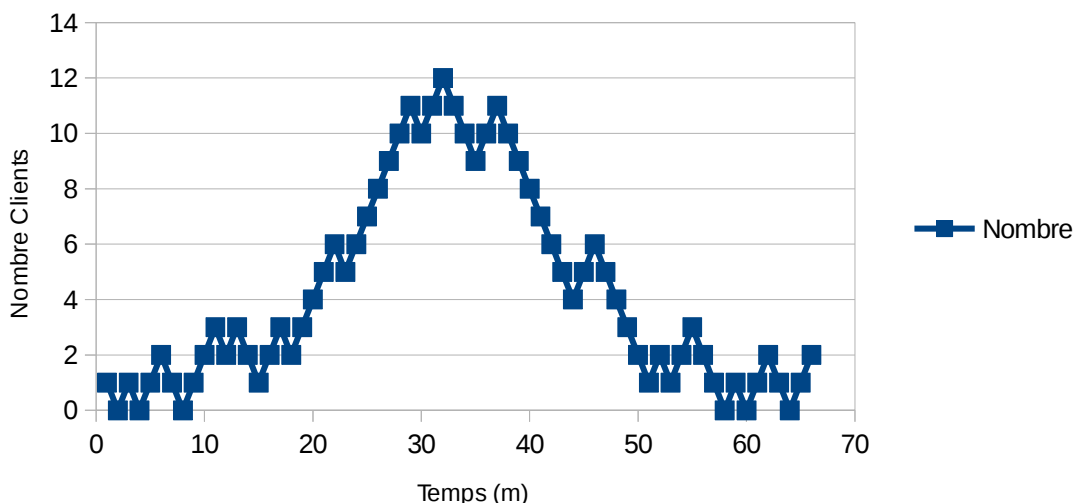
Mersenne-Twister : 0.415940

AES : 0.460536

À part pour Von Neumann, les moyennes sont proches de 0,5 et on obtient des résultats qui fluctuent ce qui indique des distributions suivant une loi de probabilités uniforme sur $[0;1]$. Ainsi, on peut dire que les algorithmes sont efficaces.

4.1) File M/M/1

Evolution du nombre de clients dans la file



Temps (m)	Nombre
4.193136	1
4.373429	0
9.265481	1
10.426352	0
13.155673	1
14.289285	2
15.834405	1
18.514717	0
22.215168	1
23.953111	2
25.158707	3
25.798488	2
28.247398	3
33.443576	2
33.920890	1
34.306592	2
35.548692	3
39.715469	2
41.062808	3
41.340692	4
41.382559	5
42.846974	6
43.040034	5
46.157789	6
47.739999	7
49.993621	8
57.760842	9
58.941086	10
59.093598	11
60.176609	10
63.030048	11
68.003888	12
69.464528	11
69.679418	10
70.763273	9
71.245351	10
71.279733	11
71.905668	10
72.390485	9
80.400396	8
81.087775	7
84.012441	6
86.569281	5
87.118020	4
87.591527	5
92.646137	6
92.976505	5
92.987858	4
94.136203	3
95.031919	2
97.832565	1
98.300688	2
99.412804	1
101.757640	2
103.461179	3
103.653739	2
106.056122	1
106.194699	0
128.062600	1
129.046315	0
133.939552	1
137.162850	2
138.845037	1
142.223544	0
159.413318	1
171.188523	2