擬似乱数の多次元一様性の検定

— C 言語で使用される擬似乱数の検証 —

2000MM026 伊藤　彰浩
指導教員　伏見 正則

1　はじめに

現在乱数は、不確定性を含むシミュレーションや暗号乱数を使うようなセキュリティー関係等などで、大きなウェイトとなっており、使われる乱数の質、つまり一様性及びランダムネスが大きな問題となっている。乱数そのものは、小規模な実験で言うとサイコロを振って出る目を計測する方法から、様々な発生方法があり、またそれらを検定する統計的検定も多種多様である。そのような中で今回は、身近に存在する C 言語で使われる、擬似乱数が、実際に有用性のあるものかという疑問に対して、一様検定の多次元化に焦点をあて、検定を積み重ねることによって実験，考察をした。今回の実験は、C 言語と S 言語（統計解析ソフト R) の二つの言語を使用する事で、検証を行なった。

2　実行結果

2.1　カイ二乗検定 [2]

ここでは、カイ二乗検定の理論を基に作成した C 言語のプログラムによって、rand() では 6 次元、drand48() では 8 次元までの検定について述べている。rand() の際に高次元化を進める中で、6 次元まで検定を重ねていくと、rand() の周期の関係もあり棄却されてしまう。そこで今回は、6 次元までの検証とした。2 次元以降は、自由度が、大きくなり数表での近似が困難なためため、Wilson-Hilferty の近似式を使い、標準正規分布 N(0,1) に近似して検定を行っている。

表 1: rand() 関数の検定結果

	rand()		
次元	1	2	3
クラス	100	100	50
np_i	100	100	100
統計量	106.09	10077	125120.
正規近似		0.55626	0.24325

表 2: rand() 関数の検定結果 2

	rand()	
4	5	6
20	10	10
100	10	100
160068.	1001680.	1012724.
0.124667	0.37917	8.960834

表 3: drand48() の検定結果

	drand48()			
次元	1	2	3	4
クラス	100	100	50	20
np_i	100	100	100	100
統計量	91.45	10003.	125336.	160986.
正規近似		0.03642	0.67488	1.74375

表 4: drand48() の検定結果 2

	drand48()			
次元	5	6	7	8
クラス	10	10	5	5
np_i	100	100	1000	10
統計量	10090910.	1001851.	78185.	429759.
正規近似	2.03041	1.30958	0.15601	159.283

2.2　Kolmogorov-Smirnov 検定 [3]

次の結果は、S 言語による "ks.gof "、R による "ks.test " によって K-S 検定を行なったものである。掲載している結果は R によるもの。(S によるものとほぼ等しいため。) 検定はどちらも 2 次元の時点で大きな変化が見られ後の検定結果が推測できたため、4 次元までの検証結果を掲載した。また、実験は、C 言語で実行した rand()、drand48() 関数のカイ二乗検定の結果 100 個をサンプルとして行なっている。

表 5: rand() 関数による K-S 検定の結果

	K-S 検定			
次元	1	2	3	4
クラス	100	100	50	20
D 値	0.1348	0.2769	0.7739	0.8638
p-value	0.0528	4.394e-07	2.2e-16	2.2e-16
自由度	99	9999	124999	159999

表 6: drand48() 関数による K-S 検定の結果

	K-S 検定			
次元	1	2	3	4
クラス	100	100	50	20
D 値	0.0604	0.3656	0.8103	0.1364
p-value	0.8583	4.888e-12	2.2e-16	2.2e-16
自由度	99	9999	124999	159999

上記の結果より、rand() によるカイ二乗検定を検証すると、5 次元までは、np=100 程度で正規近似でき、良い値が検出されたが、周期の関係もあり 6 次元になると、棄却されてしまった。ここで、より一様性を追及するため、K-S 検定によって、検定の積み重ねを行い更に、検証してみたが、1、2 次元で、すでに棄却されてしまい、多次元での一様性に問題があると言える結果となった。同様に、drand48() で検証すると、周期の関係もあるが np=1000 にて、7 次元まで良い値が検出され、8 次元での棄却という形となった。しかし、K-S 検定を重ねて行なうと、2 次元で棄却されてしまい、rand() に比べると多少、多次元への対応が可能と言えるが、速度は落ちてしまうため、どちらも、高次元での使用は好ましくないと言える一つの結果となった。

2.3 OPSO 検定 [1]

ここでは次のような手順で、OPSO 検定を行った。X_i：1 ビットの乱数列 (生成法として、整数乱数 rand() を% 65536(2^{16}) することによって 16 ビットごとの乱数に分割して用いている。5 ビットずつ取り出す作業には、ビット演算を用いている。)

1. 数列$< X_i >$を 16 ビットごとに区切る。
2. 各 16 ビットから 5 ビットを取り出し、5 ビットの整数からなる新しい列 Y_i を作り、Y_i を二個ずつ組にする。2 つ組は、$(Y_1, Y_2), (Y_2, Y_3)$ のように 1 個ずつずらして重ねながら作る。
3. 2. の組を 2^{11} 組作る。
4. 3. で作った 2^{11} 組の中に、5 ビットの 2 つ組として現れていない組の個数 S を数える。
5. 1〜4 を 12 回くり返す。各回の違いは、16 ビットから取り出す 5 ビットの位置で、1 回目は 12 ビット目から 5 ビット、2 回目は 11 ビット目から 5 ビット、…、12 回目は 1 ビット目から 5 ビットと行なう。

ここでいう分割数 d は、2^5(5 ビット)、点の個数 n は、$2^{11} \times 12$ となる。

表 7: rand() 関数による OPSO 検定の結果 (個数)

一回目	二回目	三回目	四回目
146	132	132	133
五回目	六回目	七回目	八回目
141	144	140	150
九回目	十回目	十一回目	十二回目
134	131	133	142

表 8: drand48() 関数による OPSO 検定の結果 (個数)

一回目	二回目	三回目	四回目
147	117	145	154
五回目	六回目	七回目	八回目
153	144	138	135
九回目	十回目	十一回目	十二回目
136	142	130	113

以上の各結果 (表 7 および表 8) について、乱数列の一様性の検定を行うとよいのだが残念ながら、すぐに使える数値表がないので、実行できない。そこで、便宜的に、二つの検定結果が同じ分布をしているかどうか (統計解析ソフト)R によって二標本の K-S 検定を行なった。ここでは、次のような仮説を立てて検定を行った。

H_0：2 群のデータは同一分布をしている。

H_1：2 群のデータは同一分布をしていない。

今回は、有意水準を 5 % ($\alpha = 0.05$) として両側検定を行い、帰無仮説の採否を決めた。検定の結果、$D_n = 0.25$、p-value = 0.8475 となった。この結果を n=12、α=0.05 の統計数値表にて検定すると、両側 5 %点は、±0.37543 なので、$D_n \leq D_n(\alpha)$ となり、分布に差がないといえる結果となった。

3 結論とこれからの発展

本研究についてまとめると、次のようになる。今回述べた C 言語における擬似乱数の発生関数である rand()、drand48() を、まず計算時間という点で比べると、計算に使うビット長の違いという構造上の差、また実際に計測した結果も含め drand48() が、rand() に比べて計算時間が遅くなるという結果となった。これは、発生させる乱数の個数にもよるが、10^9 や 10^{10} などにもなると、数時間以上の差が出るため、どちらの関数を使うかという点で非常に大きな要素の一つと言える。しかしその反面、周期の長さがあるがゆえに rand() に比べて drand48() の方が、多次元への対応が可能となったと言える。このような事より、小規模な実験など低次元での実験の際には、それぞれの特徴に合わせて、擬似乱数を発生させればよいが、共に線形合同法による乱数の発生ということもあり高次元空間になるほど点の密度が疎になる多次元疎結晶構造となるため、高次元での検定や特に大規模なシミュレーションでは、C 言語で使われる擬似乱数を用いる事は、危険だと言える一つの結論となった。また、今回このような形で研究を行った事で、多々存在する検定をモデルに合わせて使うという点だけではなく、一つのモデルに対していくつかの検定を重ねる事や、一様検定を多次元化する事が、モデルを検定する上で、重要だと言えることがわかった。今後の課題として、他の擬似乱数発生アルゴリズムの検定を進めると共に松本眞、西村拓士両氏が開発されたメルセンヌツイスタに関しても研究をし、またそれを用いた実験を行なっていきたいと思う。

4 参考文献

[1] 伏見正則；乱数，東京出版 (1989)

[2] 白旗慎吾；統計解析入門，共立出版株式会社 (1992)

[3]R.A. ベッカー,J.M. チェンバース,A.R. ウィルクス，渋谷政昭+柴田里程訳；S 言語，共立出版株式会社 (1991)