CRYPTREC Report 2005
(CRYPTREC-RP-2000-2005)
付録 6

擬似乱数検定のための CRYPTREC ミニマムセット仕様書

平成 18 年 2 月

CRYPTREC 事務局

1 背景

電子政府推奨暗号リストにおける擬似乱数系では、SHA-1 を使った擬似乱数生成器が例示されている。しかし、用途によっては、例示された擬似乱数生成器以外でも、暗号学的に安全性が確認できれば、用途によっては利用できる場合もある。そこで電子政府で使用される擬似乱数生成器が、少なくとも高い乱数性を持つことを検証するためのツールが必要と考えられている。
また、乱数の検定法には様々な観点からの検定法が存在しており，それらを複数集めて検定ツールとしてまとめられたものもいくつか存在する。代表的なものとしては、NIST FIPS PUB 140-2、NIST Special Publication (SP) 800-22、DIEHARD、”The Art of Computer programming 準数値算法”D.Knuth 著に記載されたものなどが知られている。しかし、これらの検定ツールを比較検討すると、

1) 検定ツールごと採用されている検定法が異なり、検定法の選択基準が明確になっていない

2) 同じ検定手法でも検定ツール毎に閾値等の設定値が異なる場合がある

など問題点が存在する。特に、NIST FIPS PUB 140-2 と NIST SP 800-22 には、いくつかの検定法に不具合があることを指摘した学術論文があり、さらに、2002 年度版暗号技術評価報告書においても同様の指摘がなされている。
他方、暗号モジュール委員会で検討中の暗号モジュール評価においても乱数検定が必要になるという背景があり、2003 年度に擬似乱数生成系調査ワーキンググループ設置し、CRYPTREC としての乱数検定ミニマムセットの策定を目標に、擬似乱数生成系の調査および検定法の調査を開始した。
2003 年度の調査の結果、CRYPTREC の乱数検定ミニマムセットにおいては理論的な裏付けが無いものについては、積極的には採用しない方向に決まった。また、理論的裏付けがあったとしても、計算機実験が行なえる程度の実際的な追試を行い、閾値等の設定値が適切かどうかの確認が必要であるとの判断となった。ただし乱数検定法は全部で 250 種類ほど知られており、全部を確認することは非現実的であるので、2004 年度以降は調査の範囲を絞ることも課題となった。その結果、2004 年度以降の活動方針を以下のように定めた。

1) FIPS 140-2、SP 800-22、DIEHARD で採用されている各検定法の調査範囲の絞り込みと理論的根拠の確認及び計算機実験による検証

2) 1) の結果を踏まえての CRYPTREC 擬似乱数検定ミニマムセットへの導入の判断を行う

3) CRYPTREC 擬似乱数検定ミニマムセットの暗号モジュール評価ツールへの組み込みの検討を行う

2004 年度としては、2003 年度に定めた活動方針に添って以下の活動を行なった。

1) FIPS 140-2、SP 800-22、DIEHARD で採用されている各検定法の調査範囲の絞り込みと理論的根拠の確認及び計算機実験による検証

2) 1) の結果を踏まえての CRYPTREC 擬似乱数検定ミニマムセットへの導入の判断を行う

特に，1) の検定方法に関しては、離散フーリエ変換検定するための数学理論を構築するための予備調査と 2003 年度未解決であった分散値の理論的確認と、擬似乱数生成系に対する各種検定方式の理論的根拠の確認及び計算機実験による検証を実施した。

2005 年度には、2003 年度および 2004 年度の調査結果に基づき、CRYPTREC としての乱数検定のためのミニマムセットとして、NIST で公表している SP 800-22 の 16 種類の検定法の中から 14 種類を採択し、「乱数検定ミニマムセット仕様」として本仕様書の作成を行なった。

2　乱数検定のためのミニマムセット案

乱数検定のためのミニマムセットとして下記のものを定める

1. 頻度検定 (Frequency Test)
2. ブロック単位の頻度検定 (Frequency Test within a Block)
3. 連検定 (Runs Test)
4. ブロック単位の最長連検定 (Test for the Longest Run of Ones in Block)
5. 2 値行列ランク検定 (Binary Matrix Rank Test)
6. 重なりのないテンプレート適合検定 (Non-overlapping Template Matching Test)
7. 重なりのあるテンプレート適合検定 (Overlapping Template Matching Test)
8. Maurer のユニバーサル統計検定 (Maurer's "Universal Statistical " Test)
9. 線形複雑度検定 (Linear Complexity Test)
10. 系列頻度検定 (Serial Test)
11. 累積和検定 (Cumulative Sums (Cusum) Test)
12. ランダム回遊検定 (Random Excursions Test)
13. 変形ランダム回遊検定 (Random Excursions Variant Test)
14. 近似エントロピー検定 (Approximate Entropy Test)

3　検定に対する可否判断基準

ミニマムセットで採用されているすべての検定は、0 と 1 からなる乱数列を対象としている。また、ミニマムセットの検定では、各検定ごとに p-value が得られる。p-value とは、真の乱数生成器が検定を行っている系列よりも乱数らしからぬ系列を生成する確率である。例として、頻度検定の場合を考える。このとき、p-value は以下のように求める。試行としては 100 万系列を 1000 本考える。

1. $X_1, X_2, \cdots, X_n$ を $\{1, -1\}$ の中の値をとる n 個の確率変数とし、$S_n = X_1 + X_2 + \cdots + X_n$ とする。

2. 系列が真の乱数生成器からの出力ならば、

$$\begin{aligned} \mu &= 0 \\ \sigma^2 &= n \end{aligned}$$

   となるので、中心極限定理より、

$$\lim_{n\to\infty} P\left(\frac{S_n}{\sqrt{n}} \leq z\right) = \Phi(z)$$

   となる。なお、

$$\Phi(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx$$

   は標準正規分布の累積分布関数である。

3. 統計量 $s = |S_n|/\sqrt{n}$ を考える。このとき、

$$P(s \leq z) = 2\Phi(z) - 1$$

   が得られ、

$$p\text{-value} = 2[1 - \Phi(s)]$$

   である。なお、ミニマムセットでは、$\Phi(z)$ の代わりに誤差関数

$$erfc(z) = \int_{z}^{\infty} \frac{2}{\sqrt{\pi}} e^{-x^2} dx$$

   を用いれば

$$p\text{-value} = erfc(s/\sqrt{2})$$

   となる。

ミニマムセットでは、p-value< 0.01 のときに良い乱数生成器ではないと判断する。

なお、統計量がカイ 2 乗統計量の場合、p-value は、

$$\int_{z}^{\infty} \frac{1}{2\Gamma(\frac{N}{2})} \left(\frac{t}{2}\right)^{\frac{N}{2}-1} e^{-\frac{t}{2}} dt$$

により求める。ただし、N は χ^2 分布の自由度であり、

$$\Gamma(\alpha) = \int_{0}^{\infty} x^{\alpha-1} e^{-x} dx$$

である。

ミニマムセットでは、1 本の標本系列に対する仮説検定で乱数生成アルゴリズムを評価するのは無意味であるという考え方から、複数の標本系列 (NIST では 1000 程度を推奨している) に対し検定を行い、

1. p-value の一様性
2. p-value が 0.01 より大きくなる割合

から乱数列の評価を行う。1. では、得られた p-value が区間 $[0, 1)$ で一様に分布しているかどうかを調べるために、$[0, 1)$ を 10 の区間に分割し、分割した区間ごとの頻度が一様になっているかどうかをカイ 2 乗検定にて検定する。カイ 2 乗検定により得られた p-value が 0.0001 以上ならば、乱数列は良い乱数生成器であると判断する。また、2. では、標本の数を m としたとき、0.01 以上となる p-value の数の割合が

$$0.99 \pm 3\sqrt{\frac{0.99 \times 0.01}{m}}$$

の範囲に入っている場合は、乱数列は良い乱数生成器であると判断する。

4　検査の概要と可否判断

4.1　頻度検定

乱数列のビットの出現頻度を求めその度数分布が一様になっているかどうかを標準正規分布の誤差関数にて検定する。

入力	$\{\varepsilon_i\}$	:	0 か 1 の値をとる乱数列
	n	:	乱数列の長さ（乱数の個数）100 万系列を 1000 本 100 万系列を 1000 本
出力	$erfc(s_{obs})$	:	標準正規分布の誤差関数からの p-value
処理内容	ステップ 1	:	系列を $\pm$ に変換し、$S_n = X_1 + X_2 + \cdots + X_n$ $(X_i = 2\varepsilon_i - 1)$ を計算する。
	ステップ 2	:	統計値 $s_{obs} = \lvert\frac{S_n}{\sqrt{n}}\rvert$ を計算する。
	ステップ 3	:	p-value $= erfc(s_{obs}/\sqrt{2})$　を計算する。

4.2 ブロック単位の頻度検定

乱数列の文字の出現頻度を求めその度数分布が一様になっているかどうかをカイ 2 乗検定にて検定する。

入力	M	:	各ブロックのビット長
	$\{\varepsilon_i\}$	:	0 か 1 の値をとる乱数列
	n	:	乱数列の長さ（乱数の個数）100 万系列を 1000 本
出力	$\chi_0^2(obs)$	:	カイ 2 乗統計量からの p-value
処理内容	ステップ 1	:	入力系列を $N = \lfloor \frac{n}{M} \rfloor$ の重なりの無いブロックに分割する。使わないビットは無視する。
	ステップ 2	:	$\pi_i = \frac{\sum_{j=1}^{M} \varepsilon_{(i-1)M+j}}{M}$ を $1 \leq i \leq N$ に対して計算する。
	ステップ 3	:	カイ 2 乗統計量 χ^2 を $\chi^2(obs) = 4M \sum_{i=1} N(\pi_i - 1/2)^2$ により計算する。
	ステップ 4	:	p-value $=$ igamc$(N/2, \chi^2(obs)/2)$ を計算する。ここで igamc は不完全なガンマ関数である。

4.3 連検定

区間 $[0,1)$ 上を分布する乱数列を上昇連、下降連で分割し、部分列の長さでクラスを定義する。部分列を長さに応じてクラスに割り当て、その度数をカイ 2 乗検定にて検定し、連の偏りを調べる。なお、乱数列が 0 と 1 からなる乱数の場合は、区間 $[0,1)$ 上に分布する乱数列への変換が必要である。

入力	n	:	ビット列の長さ
	$\{\varepsilon_i\}$	:	0 と 1 の c 中から値をとる乱数列
出力	p-value	:	カイ 2 乗統計量からの p-value（
処理内容	ステップ 1	:	事前に 1 が出力される確率 $\pi = \frac{\sum_i \varepsilon_i}{n}$ を計算する
	ステップ 2	:	頻度テストをパスするかどうか決定する。もしも $\tau \leq \|\pi - 1/2\|$ が示されるならば連検定を行う必要は無い。検定を行う必要が無い場合は p-value は 0.0000 に設定される。ここで $\tau = \frac{2}{\sqrt{n}}$ は事前に定義されていなければならない。
	ステップ 3	:	検定値 $V_n(obs) = \sum_{k=1}^{n-1} r(k) + 1$ を計算する。ここで、$\varepsilon_k = \varepsilon_{k+1}$ ならば $r(k) = 0$ で、さもなければ $r(k) = 1$ である。
	ステップ 4	:	p-value $= erfc(\frac{\|V_n(obs) - 2n\pi(1-[pi)\|}{2\sqrt{2n\pi(1-\pi)}})$ を計算する

4.4 ブロック単位の最長連検定

0 と 1 からなる乱数列を M ビット単位で分割し、最長連の長さに応じて各部分列を 7 個のクラスに割り当てその度数をカイ 2 乗検定にて検定し最長連の長さの偏りを調べる。なお、M は乱数列の長さによって決まる。

入力　$\{\varepsilon_i\}$　：　0 と 1 の中から値をとる乱数列
　　n　：　乱数列の長さ（乱数の個数）100 万系列を 1000 本
　　M　：　各ブロックのビット数

出力　p-value　：　カイ 2 乗統計量からの p-value

処理内容　ステップ 1　：　乱数列を長さ M のブロックに分割する。ただし、ブロックサイズは、$128 \leq n < 6272$ のときは $M = 8$、$6272 \leq n < 750,000$ のときは $M = 128$、$750,000 \leq n$ のときは $M = 10000$ とする。

ステップ 2　：　$M = 8$ のときは、各ブロックを 1 の最長連の長さに応じて、f_0 から f_3 の $K = 4$ 個のクラスに分割し、各クラスの度数を求める。クラスに対応する長さおよびクラスの確率は

クラス	f_0	f_1	f_2	f_3
度数	〜1	2	3	4〜
確率	0.2148	0.3672	0.2305	0.1875

となる。また、$M = 128$ のときは、以下のような $K = 6$ 個のクラスに割り当てる。

クラス	f_0	f_1	f_2	f_3	f_4	f_5
度数	〜4	5	6	7	8	9〜
確率	0.1174	0.2430	0.2493	0.1752	0.1027	0.1124

となる。また、$M = 10000$ のときは、以下のような $K = 7$ 個のクラスに割り当てる。

クラス	f_0	f_1	f_2	f_3	f_4	f_5	f_6
度数	〜10	11	12	13	14	15	16〜
確率	0.0882	0.2092	0.2483	0.1933	0.1208	0.0675	0.0727

ステップ 3　：　カイ 2 乗統計量 $\chi^2(obs) = \sum_{i=0}^{K} \frac{(f_i - N\pi_i)^2}{N\pi_i}$ を求める。

ステップ 4　：　$\chi^2(obs)$ は、自由度 $K-1$ の χ^2 分布に従うので、この分布から p-value を計算する。

最長連検定は、長さ 20000 ビットの 1 ブロックに対する検定である。

4.5　2 値行列ランク検定：(32×32) の 2 値行列ランク検定

0 と 1 からなる乱数列の 1024 ビットの部分列から (32×32) の 2 値行列を構成し、各部分列を行列のランクに応じてを 3 個のクラスに割り当て、その度数をカイ 2 乗検定にて検定しランクの偏りを調べる。

入力　$\{X_i\}$　：　0 と 1 の中から値をとる乱数列

出力　p-value　：　カイ 2 乗統計量からの p-value

処理内容　ステップ 1　：　行列のランクを、29 以下、30, 31, 32 の 4 個のクラスに分ける。

ステップ 2　：　X_i を列の先頭から順に 32 ビットずつ取り出し、32 ビットの整数からなる新しい列 $\{Y_i\}$ を作る。

ステップ 3　：　連続する 32 個の Y_i を順に並べて $GF(2)$ 上の 32×32 行列を作る。32 個の組は $(Y_1, Y_2, \cdots, Y_{32}), (Y_{33}, Y_{33}, \cdots, Y_{64})$ のように重ね合わせずに作っていく。

ステップ 4　：　ステップ 3 で得られた行列のランクを計算し、ランクに応じてクラスの度数に加えていく。各クラスの度数を f_0, f_1, f_2, f_3 とする。

ステップ 5　：　ステップ 2〜4 を 40,000 回くり返す。

ステップ 6　：　カイ 2 乗統計量 χ_0^2 を計算する。各クラスの確率の値は、次の表の値を用いる。χ_0^2 は自由度 3 の χ^2 分布に従うので、この分布から p-value を計算する。

〜28	29	30	31
0.0052854502	0.1283502644	0.5775761902	0.2887880952

4.6 重なりの無いテンプレート適合検定

0 と 1 からなる乱数列を 8 つのブロックに分割し、各ブロックごとに m ビットの窓を先頭からスライドさせ、窓と m 文字のテンプレートが適合する回数を調べる。8 ブロックそれぞれの適合回数をカイ 2 乗検定にて検定し適合回数の偏りを調べる。

入力	$\{X_i\}$	:	0 と 1 の中から値をとる乱数列
	n	:	乱数列の長さ（乱数の個数）100 万系列を 1000 本
	m	:	テンプレートの長さ
	B	:	m ビットのテンプレート
出力	p-value	:	カイ 2 乗統計量からの p-value
処理内容	ステップ 1	:	乱数列を長さ $M = 131,072 = 2^{17}$ の $N = 8$ 個のブロックに分割する。
	ステップ 2	:	ブロック j のテンプレートの適合回数を W_j とする。m ビットの窓をブロックの先頭にセットし、テンプレートが適合しないときは、窓を 1 ビットずらし、適合するときは窓を m ビットずらす。
	ステップ 3	:	中心極限定理より、各ブロックの適合回数はは平均 $\mu = (M-m+1)/2^m$、分散 $\sigma^2 = M\left(\frac{1}{2^m} - \frac{2m-1}{2^{2m}}\right)$ の正規分布に従うことを用いて、カイ 2 乗統計量 $\chi^2(obs) = \sum_{j=1}^{N} \frac{(W_j - \mu)^2}{\sigma^2}$ を求める。
	ステップ 4	:	$\chi^2(obs)$ は、自由度 N の χ^2 分布に従うので、この分布から p-value を計算する。

ミニマムセットでは、テンプレートの長さを 2 から 10 まで選択できるが、9 または 10 が推奨されている。なお、NIST のツールでは、乱数列の長さが 1,000,000 の場合のみ検定することができる。

4.7 重なりのあるテンプレート適合検定

0 と 1 からなる乱数列を 968 個のブロックに分割し、各ブロックを m 文字のテンプレートが適合する回数により 6 個のクラス割り当て、その度数をカイ 2 乗検定にて検定し適合回数の偏りを調べる。

入力	$\{X_i\}$	:	0 と 1 の中から値をとる乱数列
	n	:	乱数列の長さ（乱数の個数）100 万系列を 1000 本
	m	:	テンプレートの長さ
	B	:	m ビットのテンプレート
出力	p-value	:	カイ 2 乗統計量からの p-value
処理内容	ステップ 1	:	乱数列を長さ $M = 1032$ の $N = 968$ 個のブロックに分割する。
	ステップ 2	:	ブロック j のテンプレートの適合回数を W_j とする。m ビットの窓をブロックの先頭から 1 ビットずつずらして行き、テンプレートと窓の適合回数を数える。
	ステップ 3	:	適合回数 0 のクラスを f_0、1 のクラスを f_1、2 のクラスを f_2、3 のクラスを f_3、4 のクラスを f_4、5 以上のクラスを f_5 とし、ブロックの適合回数 W_j から各クラスの度数を求める。
	ステップ 4	:	カイ 2 乗統計量 $\chi^2(obs) = \sum_{i=0}^{5} \frac{(f_i - N\pi_i)}{N\pi_i}$ を求める。クラス f_i に対応する確率 π_i は、 $\pi_0 = e^{-\eta}$ $\pi_1 = \frac{\eta}{2} e^{-\eta}$ $\pi_2 = \frac{\eta e^{-\eta}}{8}[\eta + 2]$ $\pi_3 = \frac{\eta e^{-\eta}}{8}\left[\frac{\eta^2}{6} + \eta + 1\right]$ $\pi_4 = \frac{\eta e^{-\eta}}{16}\left[\frac{\eta^3}{24} + \frac{\eta^2}{2} + \frac{3\eta}{2} + 1\right]$ $\pi_5 = 1 - (\pi_0 + \pi_1 + \pi_2 + \pi_3 + \pi_4)$ となる。ただし、$\eta = \lambda/2$, $\lambda = (M - m + 1)/2^m$ である。
	ステップ 5	:	$\chi^2(obs)$ は、自由度 5 の χ^2 分布に従うので、この分布から p-value を計算する。

4.8 Maurer のユニバーサル統計検定

0 と 1 からなる乱数列における長さ L ビットのパターンの間隔を調べることにより乱数列の一様性・圧縮可能性を調べる。

入力　$\{X_i\}$　：　0 と 1 の中から値をとる乱数列
　　n　：　乱数列の長さ（乱数の個数）100 万系列を 1000 本
　　L　：　ブロックの長さ
　　Q　：　初期系列のブロック数

出力　p-value　：　正規分布統計量からの p-value

処理内容　ステップ 1　：　乱数列を長さ L のブロックに分割し、先頭から Q ブロック分を初期セグメントとし、残りの K ブロック分をテストセグメントとする。ただし、$K = \lfloor (n - QL)/L \rfloor$ とする。

ステップ 2　：　T_j $(0 \leq j \leq 2^L)$ の初期値を $T_j = 0$ とし、「i 番目の L ビットブロックを 2 進数とみなしたときの値が j のときに、$T_j = i$ とする」という処理を初期セグメントの先頭ブロックから初期セグメントの最終ブロックまで順に行う $(1 \leq i \leq Q)$。

ステップ 3　：　sum の初期値を 0 とし、「i 番目の L ビットブロックを 2 進数とみなしたときの値が j のときに、$sum = sum + \log_2(i - T_j)$ とし、さらに $T_j = i$ とする」という処理をテストセグメントの先頭ブロックからテストセグメントの最終ブロックまで順に行う $(Q+1 \leq i \leq Q+K)$。また、$f_n = sum/K$ を求める。

ステップ 4　：　f_n は下記の表にある平均、分散の正規分布に従うので、この分布から p-value を計算する 。

L	平均	分散
6	5.2177052	2.954
7	6.1962507	3.125
8	7.1836656	3.238
9	8.1764248	3.311
10	9.1723243	3.356
11	10.170032	3.384
12	11.168765	3.401
13	12.168070	3.410
14	13.167693	3.416
15	14.167488	3.419
16	15.167379	3.421

4.9 線形複雑度検定

0 と 1 からなる乱数列を長さ M のブロックに分割し、ブロックごとの線形複雑度を求めることにより乱数列の周期性を調べる。

入力　$\{X_i\}$　：　0 と 1 の中から値をとる乱数列
　　n　：　乱数列の長さ（乱数の個数）100 万系列を 1000 本
　　M　：　ブロックの長さ

出力　p-value　：　カイ 2 乗統計量からの p-value

処理内容　ステップ 1　：　乱数列を長さ M のブロックに分割し、$N = \lfloor n/m \rfloor$ とする。
　　ステップ 2　：　Berlekamp-Massey アルゴリズムを用いて、ブロック i の線形複雑度 L_i を求める $(i = 1, \cdots, N)$。
　　ステップ 3　：　$\mu = \dfrac{M}{2} + \dfrac{(9 + (-1)^{M+1})}{36} + \dfrac{(M/3 + 2/9)}{2^M}$ を求め、$T_i = (-1)^M \times (L_i - \mu) + 2/9$ を求める $(i = 1, \cdots, N)$。
　　ステップ 4　：　T_i の値により 7 つのクラス $\nu_0, \nu_1, \cdots, \nu_6$ を定め、各クラスの度数を求める。

クラス	T_i の範囲
ν_0	$T_i \leq -2.5$
ν_1	$-2.5 < T_i \leq -1.5$
ν_2	$-1.5 < T_i \leq -0.5$
ν_3	$-0.5 < T_i \leq 0.5$
ν_4	$0.5 < T_i \leq 1.5$
ν_5	$1.5 < T_i \leq 2.5$
ν_6	$2.5 < T_i$

　　ステップ 5　：　カイ 2 乗統計量 χ_0^2 を計算する。クラス i の確率の値 π_i は、次の表の値を用いる。χ_0^2 は自由度 6 の χ^2 分布に従うので、この分布から p-value を計算する。

クラス	確率
π_0	0.01047
π_1	0.03125
π_2	0.125
π_3	0.5
π_4	0.25
π_5	0.0625
π_6	0.02078

4.10 系列頻度検定

0 と 1 からなる乱数列における長さ m ビットのパターン長さ $m-1$ ビットのパターン、長さ $m-2$ ビットのパターンが一様に出現しているかを調べることにより乱数列の一様性・圧縮可能性を調べる。

入力	$\{X_i\}$	:	0 と 1 の中から値をとる乱数列
	n	:	乱数列の長さ（乱数の個数）100 万系列を 1000 本
	m	:	ブロックの長さ
出力	p-value	:	カイ 2 乗統計量からの p-value(2 個)
処理内容	ステップ 1	:	重なりのある m ビットブロックの列を $(X_1, X_2, \cdots, X_m), (X_2, X_3, \cdots, X_{m+1}) \cdots$ のように定める。同様に、重なりのある $m-1$ ビットブロックの列、重なりのある $m-2$ ビットブロックの列を定める。
	ステップ 2	:	m ビットパターン $i_1 i_2 \cdots i_m$ の出現頻度 $\nu_{i_1 i_2 \cdots i_m}$、$m-1$ ビットパターン $i_1 i_2 \cdots i_{m-1}$ の出現頻度 $\nu_{i_1 i_2 \cdots i_{m-1}}$、$m-2$ ビットパターン $i_1 i_2 \cdots i_{m-2}$ の出現頻度 $\nu_{i_1 i_2 \cdots i_{m-2}}$ を求める。
	ステップ 3	:	$\Psi_m^2 = \frac{2^m}{n} \sum_{i_1 i_2 \cdots i_m} \left(\nu_{i_1 i_2 \cdots i_m} - \frac{n}{2^m}\right)^2 = \frac{2^m}{n} \sum_{i_1 i_2 \cdots i_m} \nu_{i_1 i_2 \cdots i_m}^2 - n$、$\Psi_{m-1}^2 = \frac{2^{m-1}}{n} \sum_{i_1 i_2 \cdots i_{m-1}} \left(\nu_{i_1 i_2 \cdots i_{m-1}} - \frac{n}{2^{m-1}}\right)^2 = \frac{2^{m-1}}{n} \sum_{i_1 i_2 \cdots i_{m-1}} \nu_{i_1 i_2 \cdots i_{m-1}}^2 - n$、$\Psi_{m-2}^2 = \frac{2^{m-2}}{n} \sum_{i_1 i_2 \cdots i_{m-2}} \left(\nu_{i_1 i_2 \cdots i_{m-2}} - \frac{n}{2^{m-2}}\right)^2 = \frac{2^{m-2}}{n} \sum_{i_1 i_2 \cdots i_{m-2}} \nu_{i_1 i_2 \cdots i_{m-2}}^2 - n$ を求める。
	ステップ 4	:	$\nabla \Psi_m^2 = \Psi_m^2 - \Psi_{m-1}^2$ および $\nabla^2 \Psi_m^2 = \Psi_m^2 - 2\Psi_{m-1}^2 + \Psi_{m-2}^2$ を求める。
	ステップ 5	:	$\nabla \Psi_m^2$ は自由度 2^{m-1} の χ^2 分布に従うので、この分布から p-value を計算する。同様に、$\nabla^2 \Psi_m^2$ は自由度 2^{m-2} の χ^2 分布に従うので、この分布から p-value を計算する。

4.11 累積和検定

0 と 1 からなる乱数列 $X_1, X_2, \cdots X_n$ に対し、$S_i = \sum_{j=1}^{i}(2X_j - 1)$ および $S'_i = \sum_{j=n-i+1}^{n}(2X_j - 1)$ $(1 \leq i \leq n)$ の絶対値の最大値を求め、その偏りを調べる。

入力	$\{X_i\}$	:	0 と 1 の中から値をとる乱数列
	n	:	乱数列の長さ（乱数の個数）100 万系列を 1000 本
出力	p-value	:	正規分布統計量からの p-value(2 個)
処理内容	ステップ 1	:	$S_1 = 2X_1 - 1$ とし、$S_k = S_{k-1} + 2X_k - 1$ を求める $(2 \leq k \leq n)$。さらに、$z = \max_{1 \leq k \leq n} \lvert S_k \rvert$ を求める。[Mode 0]
	ステップ 2	:	$p - \text{value} = 1 - \sum_{k=(-n/z+1)/4}^{(n/z-1)/4} \left[\Phi\left(\frac{(4k+1)z}{\sqrt{n}} \right) - \Phi\left(\frac{(4k-1)z}{\sqrt{n}} \right) \right] + \sum_{k=(-n/z-3)/4}^{(n/z-1)/4} \left[\Phi\left(\frac{(4k+3)z}{\sqrt{n}} \right) - \Phi\left(\frac{(4k+1)z}{\sqrt{n}} \right) \right]$ を求める。ただし、Φ は標準正規分布の累積分布関数である。
	ステップ 3	:	$S_1 = 2X_n - 1$、$S_k = S_{k-1} + 2X_{n-k} - 1$ とし、ステップ 1、2 を行う。[Mode 1]

4.12 ランダム回遊検定

0 と 1 からなる乱数列 $X_1, X_2, \cdots X_n$ に対し、$S_i = \sum_{j=1}^{i}(2X_j - 1)$ $(1 \leq i \leq n)$ を求め、$S_i = 0$ から次に 0 になるまでを 1 つのサイクルとみなし、-4〜-1、および、1〜4 の 8 種類の状態ごとにサイクルの出現度数の偏りを調べる。

入力　$\{X_i\}$　：　0 と 1 の中から値をとる乱数列
　　n　：　乱数列の長さ（乱数の個数）100 万系列を 1000 本

出力　p-value　：　カイ 2 乗統計量からの p-value(8 個)

処理内容　ステップ 1　：　$S_1 = 2X_1 - 1$ とし、$S_k = S_{k-1} + 2X_k - 1$ を求める $(2 \leq k \leq n)$。さらに、$0, S_1, S_2, \cdots, S_n, 0$ という新しい数列 S' を定める。

ステップ 2　：　0 から始まり次に現れる 0 までを 1 サイクルとし、 数列 S' からサイクルを求める。なお、数列 S' のサイクル数 J が $J < 500$ ならば、検定を中止する。

ステップ 3　：　8 種類の状態値を-4〜-1、および、1〜4 とし、各サイクルごとに状態値の出現度数を求める。

ステップ 4　：　状態値 x の出現度数が k となるサイクルの数を $\nu_k(x)$ とし、8 種類のすべての状態に対し $\nu_k(x)$ を求める $(1 \leq k \leq 5)$。ただし、$\nu_5(x)$ は状態値 x の出現度数が 5 回以上のサイクル数とする。

ステップ 5　：　8 種類のすべての状態値に対し、カイ 2 乗統計量 $\chi^2(obs) = \sum_{k=0}^{5} \dfrac{(\nu_k(x) - J\pi_k(x))^2}{J\pi_k(x)}$ を求める。ただし、

	$\pi_0(x)$	$\pi_1(x)$	$\pi_2(x)$	$\pi_3(x)$	$\pi_4(x)$	$\pi_5(x)$
$\|x\| = 1$	0.5000	0.2500	0.1250	0.0625	0.0312	0.0312
$\|x\| = 2$	0.7500	0.0625	0.0469	0.0352	0.0264	0.0791
$\|x\| = 3$	0.8333	0.0278	0.0231	0.0193	0.0161	0.0804
$\|x\| = 4$	0.0875	0.0156	0.0137	0.0120	0.0105	0.-733

である。

ステップ 6　：　$\chi^2(obs)$ は自由度 5 の χ^2 分布に従うので、この分布から 8 種類のすべての状態値に対する p-value を計算する。

4.13 変形ランダム回遊検定

0 と 1 からなる乱数列 $X_1, X_2, \cdots X_n$ に対し、$S_i = \sum_{j=1}^{i}(2X_j - 1)$ $(1 \leq i \leq n)$ を求め、-9〜-1、および、1〜9 の 18 種類の状態の出現度数の偏りを調べる。

入力	$\{X_i\}$	:	0 と 1 の中から値をとる乱数列
	n	:	乱数列の長さ（乱数の個数）100 万系列を 1000 本
出力	p-value	:	正規分布統計量からの p-value(18 個)
処理内容	ステップ 1	:	$S_1 = 2X_1 - 1$ とし、$S_k = S_{k-1} + 2X_k - 1$ を求める $(2 \leq k \leq n)$。さらに、$0, S_1, S_2, \cdots, S_n, 0$ という新しい数列 S' を定める。
	ステップ 2	:	0 から始まり次に現れる 0 までを 1 サイクルとし、 数列 S' からサイクルを求める。なお、数列 S' のサイクル数を J とする。
	ステップ 3	:	18 種類の状態値を-9〜-1、および、1〜9 とし、状態値 x の出現度数 $\xi(x)$ を求める。
	ステップ 4	:	18 種類のすべての状態値に対する p-value を $p - \text{value} = erfc\left(\dfrac{\lvert\xi(x) - J\rvert}{\sqrt{2J(4\lvert x\rvert - 2)}}\right)$ により求める。ただし、$erfc$ は標準正規分布の誤差関数である。

4.14 近似エントロピー検定

0 と 1 からなる乱数列における長さ m ビットのパターン長さ $m+1$ ビットのパターンが一様に出現しているかを調べることにより乱数列の一様性・圧縮可能性を調べる。ただし、$m < \log(n) - 7$ とする。

入力	$\{X_i\}$	:	0 と 1 の中から値をとる乱数列
	n	:	乱数列の長さ（乱数の個数）100 万系列を 1000 本
	m	:	ブロックの長さ
出力	p-value	:	カイ 2 乗統計量からの p-value
処理内容	ステップ 1	:	重なりのある m ビットブロックの列を $(X_1, X_2, \cdots, X_m), (X_2, X_3, \cdots, X_{m+1}) \cdots$ のように定める。同様に、重なりのある $m+1$ ビットブロックの列を定める。
	ステップ 2	:	すべての m ビットパターンに対して、出現頻度 $\#i$ および $C_i^m = \#i/n$ を求める。ただし、i は m ビットブロックを 2 進数とみなしたときの値とする。さらに、$\phi^{(m)} = \sum_{i=0}^{2^m-1} \pi_i \log \pi_i$ を求める。ただし、$\pi_i = C_j^m,\ j = \log_2 i$ とする。
	ステップ 3	:	すべての $m+1$ ビットパターンに対して、出現頻度 $\#i$ および $C_i^{m+1} = \#i/n$ を求める。ただし、i は $m+1$ ビットブロックの値を 2 進数とみなしたときの値とする。さらに、$\phi^{(m+1)} = \sum_{i=0}^{2^{m+1}-1} \pi_i \log \pi_i$ を求める。ただし、$\pi_i = C_j^{m+1},\ j = \log_2 i$ とする。
	ステップ 4	:	カイ 2 乗統計量 $\chi_0^2 = 2n[\log 2 - (\phi^{(m)} - \phi^{(m+1)})]$ を求める。
	ステップ 5	:	χ_0^2 は自由度 2^m の χ^2 分布に従うので、この分布から p-value を計算する。

5　可否判定条件

以下に p-value の判定条件を定める。条件を満たさない場合はランダムでないと判定する。

表 1: 可否判定条件

検定法	判定条件
頻度検定	p-value ≥ 0.01 ならばランダムと判定
ブロック単位の頻度検定	p-value ≥ 0.01 ならばランダムと判定
連検定	p-value ≥ 0.01 ならばランダムと判定
ブロック単位の最長連検定	p-value ≥ 0.01 ならばランダムと判定
2 値行列ランク検定	p-value ≥ 0.01 ならばランダムと判定
重なりのないテンプレート適合検定	p-value ≥ 0.01 ならばランダムと判定
重なりのあるテンプレート適合検定	p-value ≥ 0.01 ならばランダムと判定
Maurer のユニバーサル統計検定	p-value ≥ 0.01 ならばランダムと判定
線形複雑度検定	p-value ≥ 0.01 ならばランダムと判定
系列頻度検定	p-value ≥ 0.01 ならばランダムと判定
累積和検定	p-value ≥ 0.01 ならばランダムと判定
ランダム回遊検定	p-value ≥ 0.01 ならばランダムと判定
変形ランダム回遊検定	p-value ≥ 0.01 ならばランダムと判定
近似エントロピー検定	パラメータの範囲を狭め、 $\log(n) - 7$ 以下ならばランダムと判定