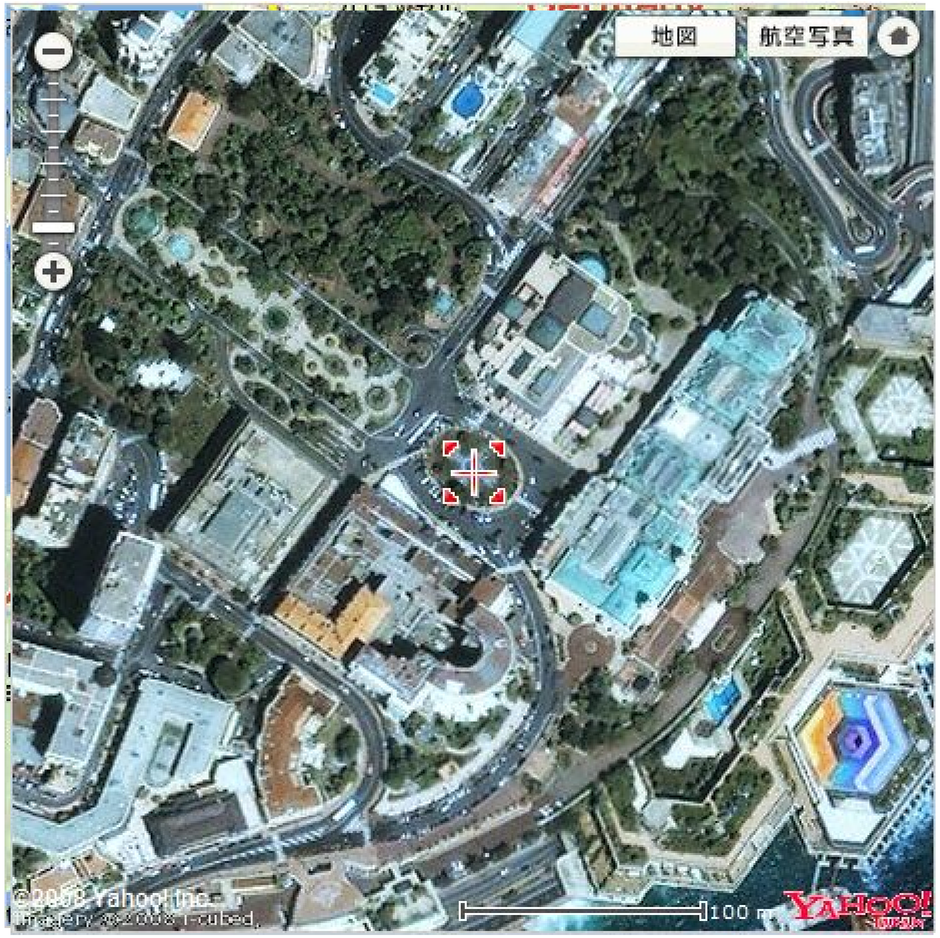
コンピュータと数学

さらにそれを超えて
モンテカルロへ


地図
航空写真
©2008 Yahoo! Inc.
Imagery ©2008 i-cubed.
100 m

モンテカルロ法 (**Monte Carlo method**, **MC**)とは
シミュレーションや数値計算を乱数を用いて行なう手法の総称。
元々は、中性子が物質中を動き回る様子を探るために
ジョン・フォン・ノイマンにより考案された手法。
カジノの都市国家モナコ公国の4つの地区（カルティ）の一つである
モンテ・カルロから名づけられた。

読んで楽しいカルチャー講座 パート2

宇部工業高等専門学校

◇５◇

ちょ～っとおもしろい 円周率3.14…の計算方法

電気工学科助教授　岡村　好庸

円周率とは円周の長さとその円の直径の比である。これをπ（パイ）と表し、近似値として三・一四が実際の計算によく用いられている。この三・一四を求める方法にはいろいろあるが、今回はコンピューター・ルーレットを用いた近似的な計算方法を述べてみよう。

ルーレットとは、賭博で一から三十六までの数字を書いた円盤の所要のところに溝けて、円盤を高速に回転させ玉を入れて円盤が止まったときの玉の位置によって勝負を争う賭博用具のひとつである。このギャンブルはどの数字にも、等しく玉が止まる、すなわちどの数字が採られる確率も等しいことが前提である。

コンピューター ルーレットの応用

コンピューターでソフトウエア（プログラム）をつくるとき、高級言語と呼ばれるプログラミング言語を用いることが多い。例えばベーシック、フォートランやCと呼ばれるものである。これら言語には通常〇から一までのある数、例えば〇・一二三四五六や〇・六二五七三九などを発生させる機能が用意されている。そこでこのような数をたくさん発生させてその分布をみると、〇から一までの間に一様に分布していることが分かる。これは〇から一までのどの数も等しい確率で発生することを意味する。このような数のことを一様乱数という。

この機能はルーレットと非常に似ている。コンピューター上で乱数を発生させて、これがある区間、例えば〇・一と〇・二の間の数となることは、円盤を回してある数が選ばれるということに対応している。異なるのは、ルーレットでは数の種類が限られているが、コンピューター上では区間をかなり狭くしてもかまわない点にある。そこで、このような乱数を発生させる機能をコンピューター・ルーレットと名づけよう。

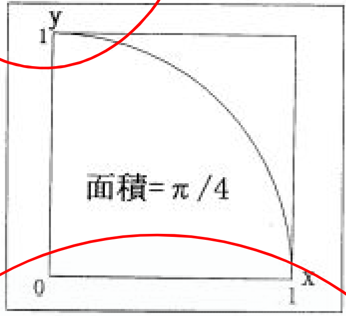
このルーレットは確率的な現象のシミュレーションに用いられたり、決定的な多次元積分の近似計算などに幅広く用いられているが、ここでは分かり易い例として円周率の計算にどのように利用されるかを説明する。

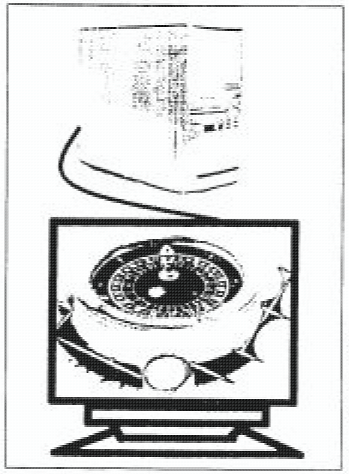
コンピューター・ルーレットにより乱数をたくさん発生させて、奇数番目の数をx、その次の偶数番目の数をyとして、図の（x，y）の位置に次々と点を打つことにする。たくさん発生させると、これらの点がその一辺の長さ一の四角を埋め尽くすようになる。ある程度埋め尽くされると、図の斜線の領域に打った点の数を数え、四角の中の点の数で割ると良い。これを四倍すれば、円周率の値が求まる。

これは半径一の円の面積が、円周率となることを利用している。すなわち、図の斜線の面積は半径一の円の四分の一なので、円周率の四分の一・π／4である。さきほどの割った数はこれに相当する。したがってその数を四倍すれば円周率の値が求められることになる。プログラムを作り、百万個の乱数を発生させ計算すると、三・一四〇一という値が得られた。このような方法には統計誤差が生じるので、誤差を最小にする様々な試みも工夫されている。

私自身はカジノのルーレットをしたことはないし、ロシアンルーレットも当然したことはないが、コンピューター・ルーレットに関してはいろんなところで利用している。原子内電子の輸送、励起、工程管理の問題等一見無関係に見える問題にも応用できる。このコンピューター・ルーレットを利用した計算技術は賭けで有名なモンテカルロにちなみ、一般にモンテカルロ法と呼ばれている。

ギャンブルは大多数の人にとり儲からないものと決まっているが、ひょっとしたらという夢を与える。コンピューター・ルーレットは様々な場面でその有用性を示しているが、ひょっとしたらまだ私達が知らない使い方があるのかもしれない。皆さんもなにか利用法を考えてみるとおもしろいのでは。でもギャンブルは控えめに。





岡村好庸（おかむら・よしのぶ）一九五〇年生まれ。専門・情報処理と非平衡統計物理。趣味・マンガの古本収集、温泉めぐり。

ひとこと…数年前に「製品の保証期間一年はなぜある」を連載しましたが、今回はコンピューター・ルーレットの話です。いつも思うのですが、身近なことに疑問を抱くことが科学する一歩です。なぜなぜ疑問をもちましょう。

様々な確率事象を乱数を利用してシミュレーションや数値積分をおこないます。
シミュレーションを通して、コンピュータ乱数の有用性や汎用性を理解することが目的です。

1) 誕生日が同じ人はどれぐらいいる? (20分)
数学の問題として解く
シミュレーションをする[birthday.exe]

2) 赤胴鈴之助問題 (40分)
実験をする
シミュレーションをする[akadou.exe]

3) 乱数 (20分)
乱数についての説明
乱数の様子をシミュレーションする[ransuu.exe]

4) 円周率の計算 (20分)
円周率、定積分のプログラム説明
プログラムによる計算実行[pi.exe]

5) 試作データから量産の可否を判断する (20分)
正規分布とその乱数の作成方法の説明
シミュレーションをする[speccheckS.exe]

6) 土砂崩れ (30分)
説明とシミュレーション[SelfOrganization.exe]

1) 誕生日が同じ人はどれぐらいいる？

学生40人のクラスで、生まれた月も日も同じという人は どれぐらいの確率でいるのだろうか。
一般にn人ではどうか。

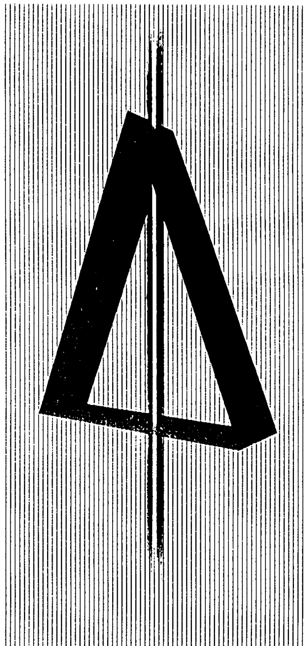
皆さんの予想は？

| 61 啓林館 | 数 I 628 |

文部省検定済教科書 高等学校数学科用

高等学校 数学 I 改訂版

山本芳彦 編



啓林館

発展 誕生日の同じ人がいる確率

高校の1学級のわずか40人くらいの中に生まれた月も日も全く同じ人がいるというのは，珍しいことのように思われる。ところが実際に調べてみると，意外にそういう人がいるものである。

このことを，確率の面から調べてみよう。

生まれた年はうるう年ではないものとし，1年のどの日に生まれることも同様に確からしいとすると，40人の生徒の誕生日のあり方は，全部で 365^{40} 通りある。

このうち，生徒の誕生日がすべて異なる場合の数は，${}_{365}P_{40}$ 通りである。

したがって，40人の生徒の誕生日がすべて異なる確率は，

$$\frac{{}_{365}P_{40}}{365^{40}} = \frac{365 \times 364 \times 363 \times \cdots\cdots \times 326}{365^{40}} \fallingdotseq 0.1088$$

よって，40人の生徒の中に，誕生日の同じ人がいる確率は，

$$1-0.1088 = 0.8912$$

となり，かなり大きいものになる。

人数をいろいろ変えて，少なくとも2人が同じ誕生日である確率を求めると右の表のようになり，人数の増加とともに1に近づいていく。

100人の場合には，この確率は

0.999999

よりも大きくなる。

人数	確 率	人数	確 率
5	0.0271	40	0.8912
10	0.1169	45	0.9410
15	0.2529	50	0.9704
20	0.4114	55	0.9863
25	0.5687	60	0.9941
30	0.7063	70	0.9992
35	0.8144	80	0.9999

生まれた年はうるう年ではないものとし，1年のどの日に生まれることも同様に確からしいとすると，40人の生徒の誕生日のあり方は，全部で 365^{40} 通りある。

このうち，生徒の誕生日がすべて異なる場合の数は，${}_{365}P_{40}$ 通りである。

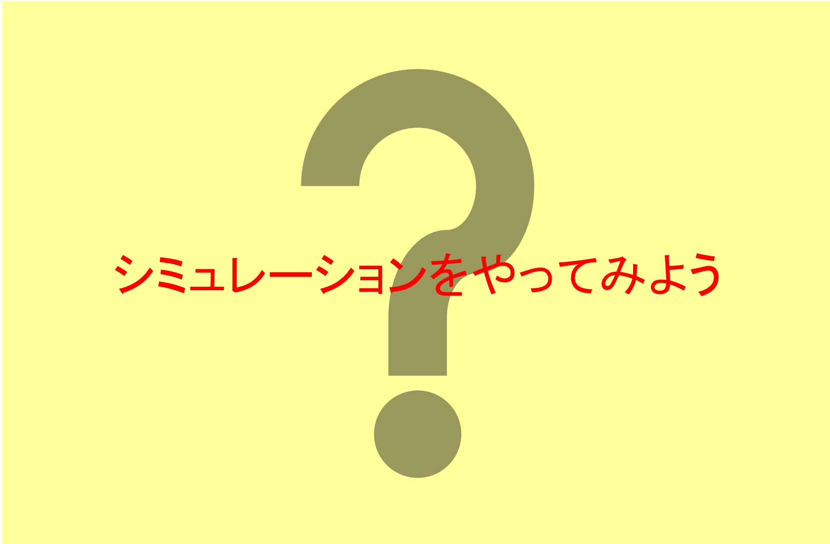
したがって，40人の生徒の誕生日がすべて異なる確率は，

$$\frac{{}_{365}P_{40}}{365^{40}} = \frac{365 \times 364 \times 363 \times \cdots\cdots \times 326}{365^{40}} \fallingdotseq 0.1088$$

よって，40人の生徒の中に，誕生日の同じ人がいる確率は，

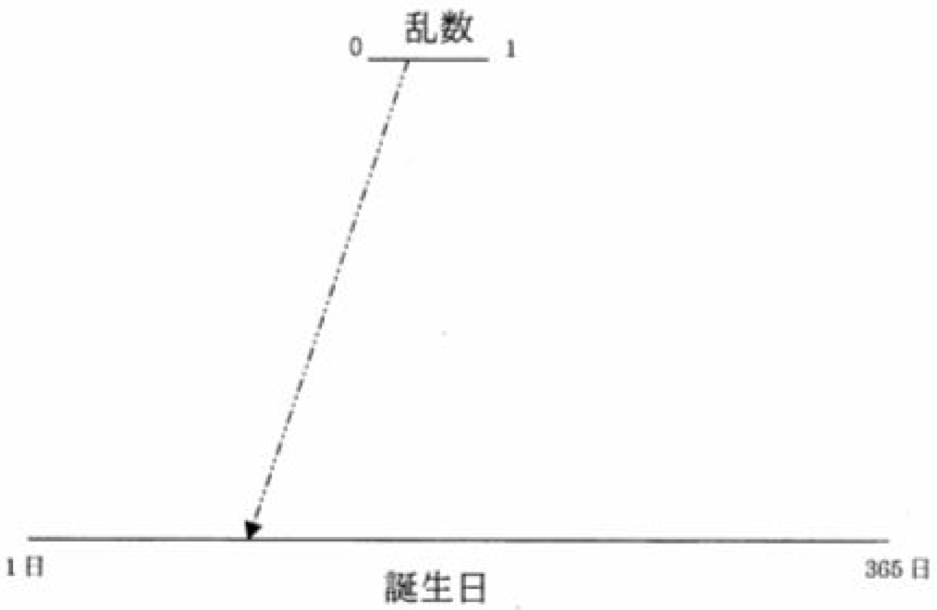
$$1 - 0.1088 = 0.8912$$

となり，かなり大きいものになる。


シミュレーションをやってみよう

Rnd 関数 (Visual Basic)

0 以上 1 未満の値を返す関数



```
cnt = 0                          '同じ誕生日の人がいるクラス数を数えるカウンター
      For j = 1 To numOfClass
         'j番目のクラスの人の誕生日をシミュレートする
         '365日のどの日にも誕生日の人はいないと初期化する
         For i = 1 To 365
            a(i) = 0         '第i日目の誕生日の人の数
         Next i

         Randomize()
         For k = 1 To ninzu
            r = Rnd() * 365 + 1     'k番目の人の誕生日をrにより決定する
            nr = Int(r)        'k番目の人の誕生日はnr日目
            a(nr) = a(nr) + 1   'nr日目の誕生日の人の数をカウントする
            If a(nr) = 2 Then   'もしすでに同じ誕生日の人がいれば
               cnt = cnt + 1   'j番目のクラスには同じ誕生日の人がいるということで、そのクラス数を増やす
               GoTo Label      '(j+1)番目のクラス人の誕生日をシミュレートするステートメントに移る
            End If
         Next k
Label:
      Next j
x = cnt / numOfClass
```

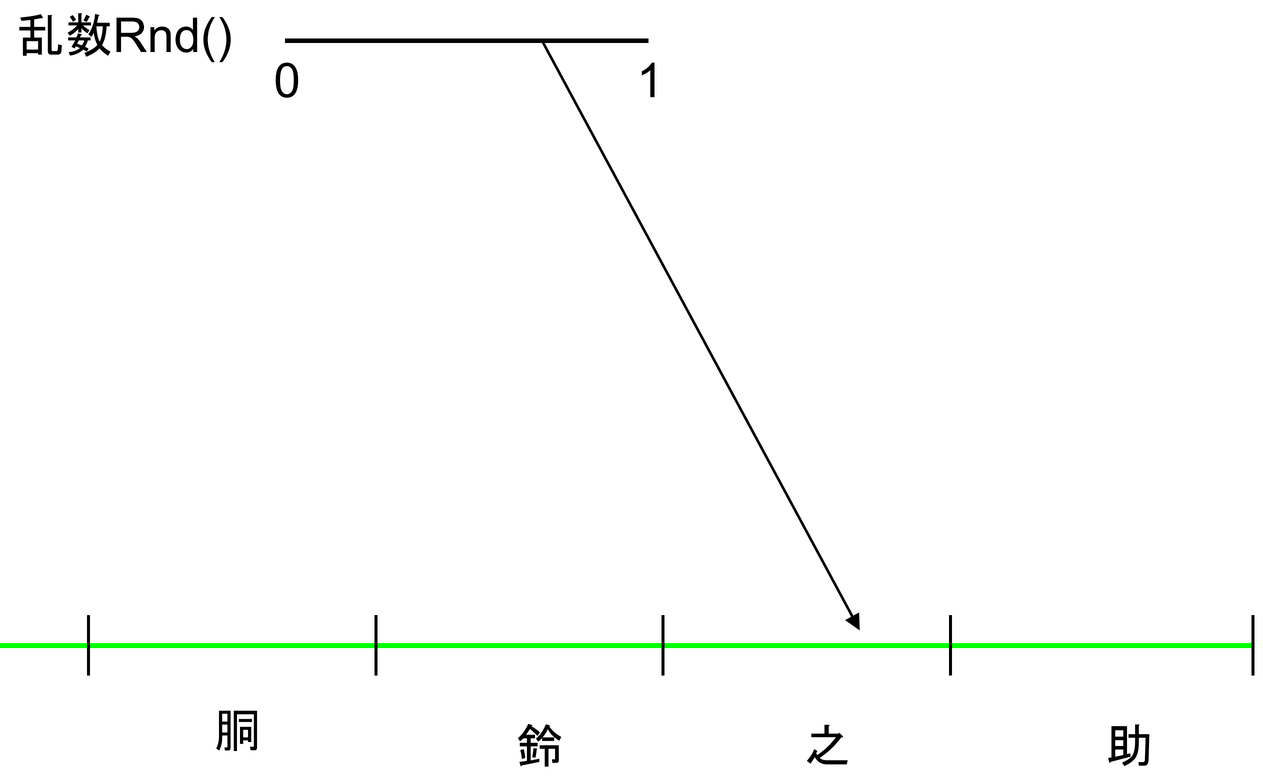
2) 赤胴鈴之助問題

赤胴鈴之助問題とは

　キャラメルを1箱買うと赤胴鈴之助のどれか１文字が箱の中にはいっている。赤胴鈴之助の5文字を集めると賞品がもらえる。そこで、賞品を手にいれるためには平均何箱買わなければならないか。
(各文字は等確率で入っているとする）

！！実験をやってみよう！！

シミュレーション実行


乱数Rnd()
0
1
赤
胴
鈴
之
助

```
R = Rnd()
Call Haitteitamono(t, A, R, R1, R2, R3, R4)
```

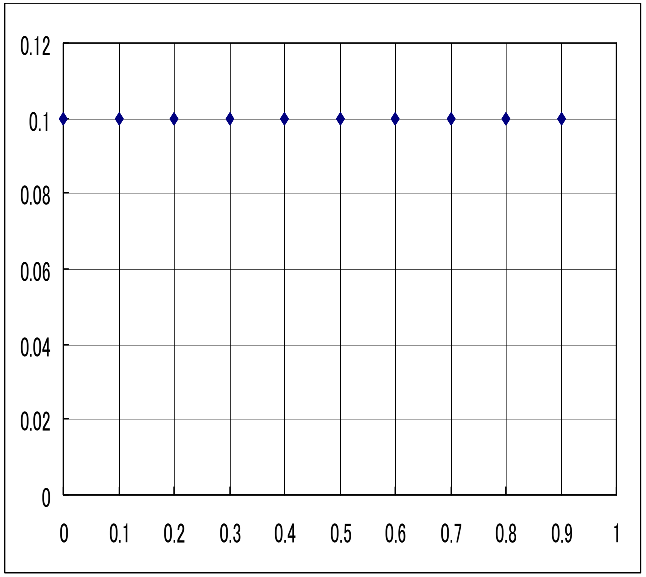
```
Sub Haitteitamono(ByVal t, ByVal A(), ByVal R, ByVal R1, ByVal R2, ByVal R3, ByVal R4)
        '乱数RによりA()を"赤""胴""鈴""之""助"に分類
        If R < R1 Then
            A(t) = 1 '"赤"
        Else
            If R < R1 + R2 Then
                A(t) = 2  '"胴"
            Else
                If R < R1 + R2 + R3 Then
                    A(t) = 3 '"鈴"
                Else
                    If R < R1 + R2 + R3 + R4 Then
                        A(t) = 4 '"之"
                    Else
                        A(t) = 5 '"助"
                    End If
                End If
            End If
        End If
End Sub
```

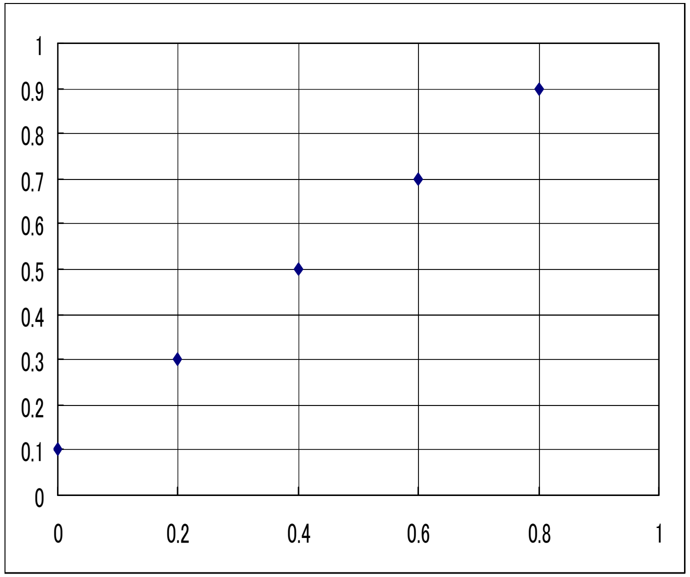
1	1	3	5	2	5	3	3	2	4
A(1)	A(2)	A(3)	A(4)	A(5)	A(6)	A(7)	A(8)	A(9)	A(10)

3) 乱数(Random Numbers)

乱数とはでたらめな数列のこと

でたらめ=出現がほぼ同じ割合（一様乱数）＋規則性がない





乱数列　0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, の分布ヒストグラムとペア打点図

乱数の様子を調べるプログラム実行

参考：**Visual Basic の Rnd**

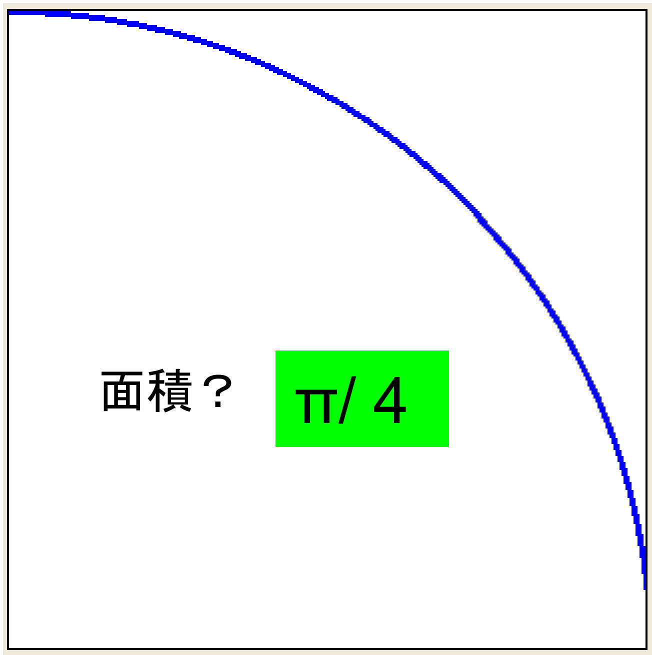
```
static long x=327680;
float Rnd(void)
{
x=(x*16598013+12820163)%16777216;
return x*(1.0/16777216.0);
}
```

24ビット線形合同法(周期は2^{24}＝16777216。精度24ビット)。
Randomize は与えられた種を16ビットの整数に変換して初期化（65536の系列）

4) 円周率の計算と定積分

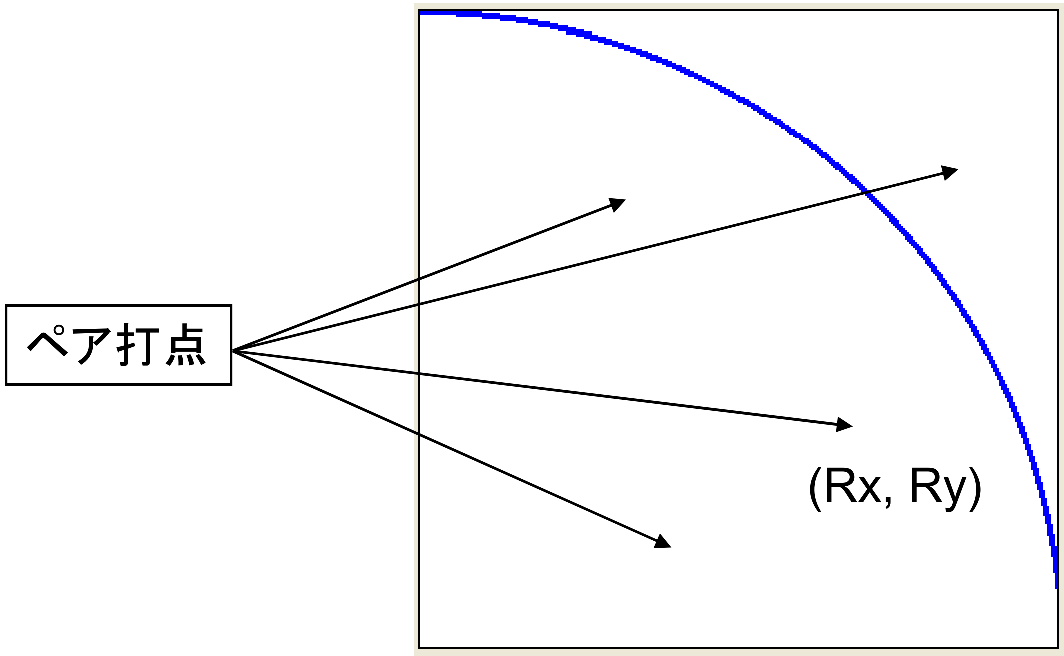
円周率π=(円周の長さ) / (円の直径)

値はいくらでしょうか？



```
'乱数を発生
        For I = 1 To N
           Rx = Rnd()
           Ry = Rnd()
'円周内の点を数える
           If Ry < Math.Sqrt(1 - Rx ^ 2) Then Cnt = Cnt + 1

        Next I
Pai = 4 * Cnt / N      '円周内の点を4倍すると円周率となる
```



定積分

$$\int_0^1 \sqrt{1-x^2}dx$$

π/4

積分値は？

$$\int_0^1 \frac{1}{1+x^2}dx$$

$$\int_0^1 \frac{3}{(1+z^2)(4-2z)}dx$$

$$z = 2 - \sqrt{4-3x}$$

$$\int_0^1 \sqrt{1-x^2}dx$$

$$\int_0^1 \frac{1}{1+x^2}dx$$

$$\int_0^1 \frac{3}{(1+z^2)(4-2z)}dx$$

$$z = 2 - \sqrt{4-3x}$$

```
For I = 1 To N
        R = Rnd()

‘aaを被積分関数とする
If RadioButton1.Checked = True Then
        aa = Math.Sqrt(1 - R ^ 2)
ElseIf RadioButton2.Checked = True Then
        aa = 1 / (1 + R ^ 2)
ElseIf RadioButton3.Checked = True Then
        x = 2 - Math.Sqrt(4 - 3 * R)
        aa = (1 / (1 + x ^ 2)) / ((4 - 2 * x) / 3)
End If

'aaの値をすべて加える
        Cnt = Cnt + aa
Next I

Pai = 4 * Cnt / N
```

円周率と定積分

参考: 多重定積分

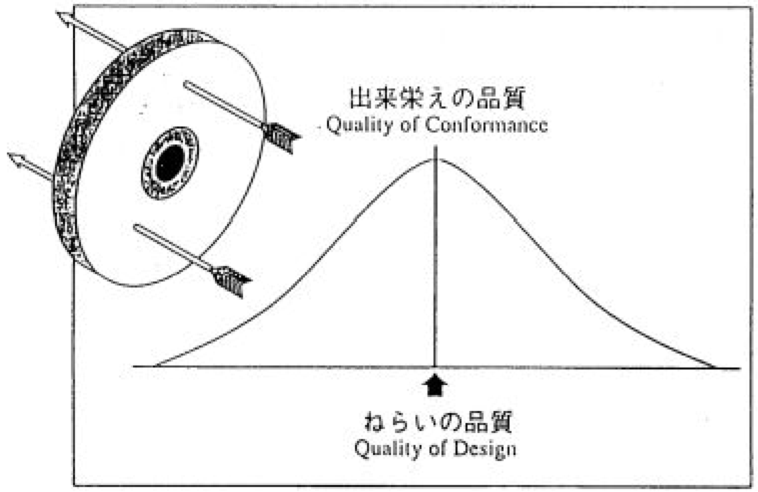
積分式は

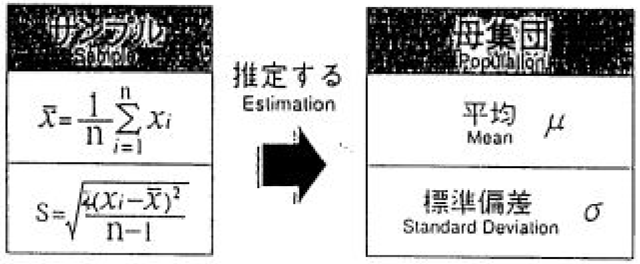
$$S = \int_0^1 dx_1 \int_0^1 dx_2 \int_0^1 dx_3 \int_0^1 dx_4 \int_0^1 dx_5 \exp[-\{x_1 x_3 + \sqrt{x_1 x_2 x_4} + \sin(x_2 x_3) + \cos(x_4 x_5) + x_5^2\}]$$

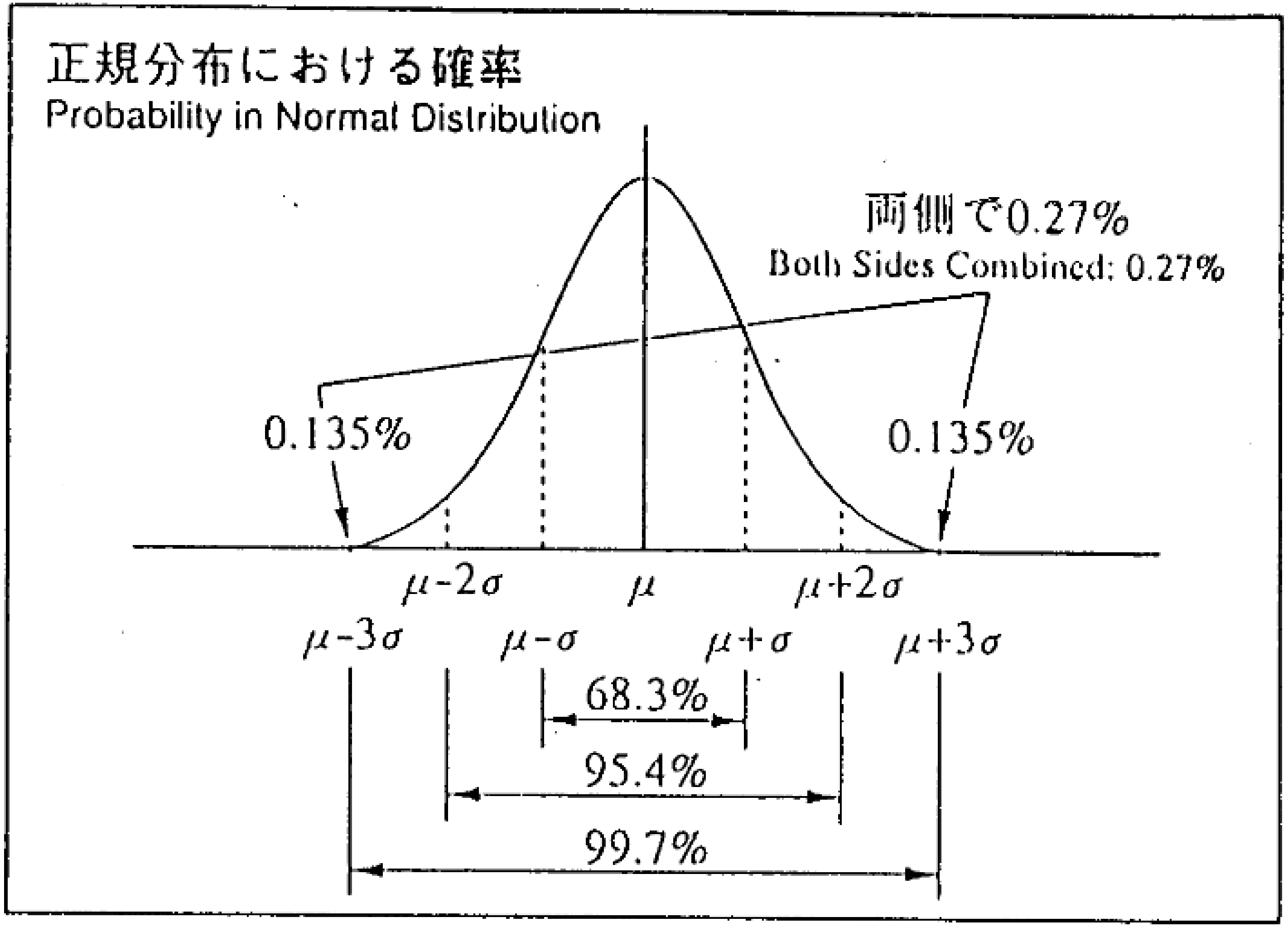
乱数を用いて S を求める．その値を s_i として $i = 1, 2, \cdots, n$ 回行う．これに対して統計処理を行う．

多重定積分をモンテカルロ法で

5) 試作データから量産の可否を判断する

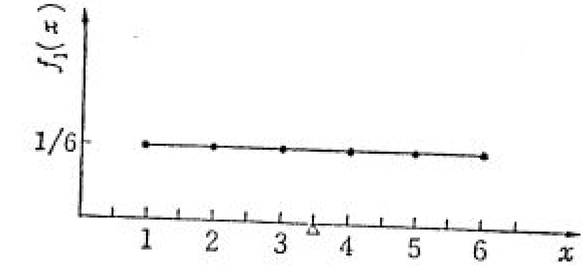




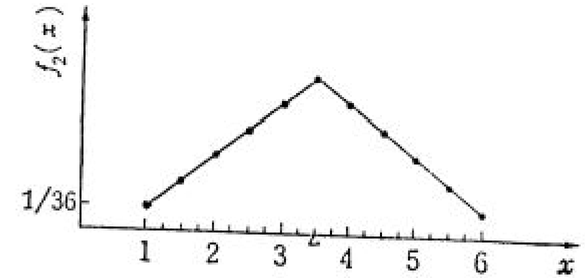


正規分布における確率
Probability in Normal Distribution
両側で0.27%
Both Sides Combined: 0.27%
0.135%
0.135%
μ-2σ
μ
μ+2σ
μ-3σ
μ-σ
μ+σ
μ+3σ
68.3%
95.4%
99.7%

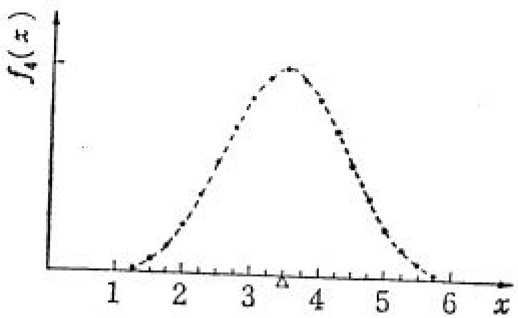
中心極限定理



1個のサイコロを振ったとき出る目の確率



2個のサイコロを振ったとき出る目の和を
2で割った値の出る確率



4個のサイコロを振ったとき出る目の和を
4で割った値の出る確率

平均 aveP、標準偏差　stdP　の正規分布（ガウス分布）乱数を発生する関数
normalRnd(aveP , stdP)

```
Function normalRnd(ByVal aveP, ByVal stdP)
        Dim j As Integer
        Dim r, sumNum, z As Double
        r = 0 : sumNum = 10
        For j = 1 To sumNum Step 1 : r = r + Rnd() : Next j
        z = (r - (sumNum / 2)) / Math.Sqrt(sumNum / 12)
        Return z * stdP + aveP
    End Function
```

一様乱数を足している

問題例

　ある製造ラインのサンプル強度を測定すると、平均値９０、標準偏差１２のデータを得た。
６０以上の強度を持つものを合格とすると、歩留まりがいくらになるかを求めよ。
（80％以上ならばこの製造ラインは合格とする）

歩留まり

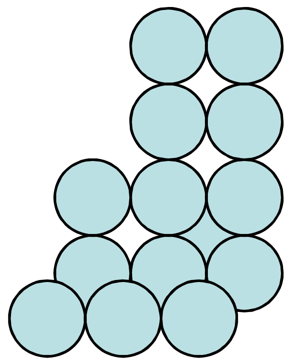
6) 土砂崩れ

自己組織的臨界状態

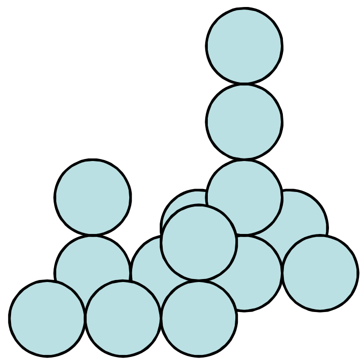
相互作用のある大規模系は臨界状態へと常に自己を組織化しており
その状態でのちょっとした出来事が連鎖反応を起こして破局に至るという理論

Per.Bak, Kurt A. Weisenfield, Chang Tang (1988)

1回目の崩壊



2回目の崩壊



$$\ln n(E) = -a \ln E + b$$

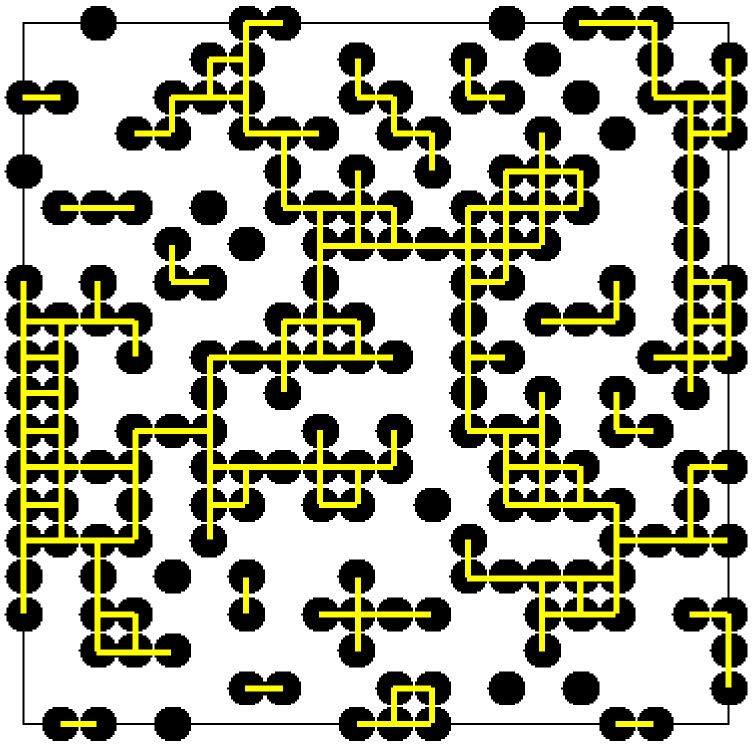
そのような崩壊
（発生エネルギーE）
が起こる頻度

連続崩壊の回数
（発生するエネルギーE）

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) _
                                                            Handles Timer1.Tick
    Dim x, y As Double
    Dim j, intx, inty, i, k As Integer
    x = Rnd()
    y = Rnd()
   ‘対応する整数値に変換 x->intx, y->inty
    c(intx, inty) = c(intx, inty) + 1
    count = 0
    Do
        flag = False
        For i = 1 To xbunkatu
            For k = 1 To xbunkatu
                If c(i, k) >= maxz Then
                    flag = True
                    c(i, k) = c(i, k) - 4  '最高の高さになれば周囲に散らばらして4個減る
                    '端からこぼれ落ちてなくなる場合（表面効果）
                    If i < xbunkatu Then c(i + 1, k) = c(i + 1, k) + 1
                    If i > 1 Then c(i - 1, k) = c(i - 1, k) + 1
                    If k < ybunkatu Then c(i, k + 1) = c(i, k + 1) + 1
                    If k > 1 Then c(i, k - 1) = c(i, k - 1) + 1
                    count = count + 1
                End If
            Next k
        Next i
    Loop While flag = True
    If count >= 1 Then frequency(count) = frequency(count) + 1
End Sub
```

クリック

おまけ　山火事の問題



本日はお疲れ様でした

気をつけてお帰りください