

モンテカルロ法とその応用

森脇義雄・田淵謹也

近年数学的に取り扱うことが困難な複雑で大規模なシステムの解析にシミュレーションの方法がしばしば用いられる。高速大型電子計算機を利用すれば、この方法は比較的簡単に実行することができ、システム中に確率過程を含み解析的方法が不可能な問題では特に有効である。ここではシミュレーションの基本的道具である乱数の作り方とその統計的性質について検討し、応用の一例として待ち行列モデルのシミュレーションについて解説する。

1. 緒言

1946年 Los Alamos において von Neumann と Ulam が核反応の計算をしたときに始まるモンテカルロ法は確率現象を数値実験として机上に再現し、その様相を観測しようというもので、考え方は単純であるが、応用面が広く、その後本来非確率的な計算、たとえば逆行列、偏微分方程式、積分方程式の解法にも適当な確率過程と結びつけることにより応用できることが指摘された¹⁾。しかし多くの場合、モンテカルロ法を使わない直接の数値計算の方がはるかに能率のよいことが経験的に示されている。特に、モンテカルロ法で精度を上げようとすると必要な実験回数が急激に増加し、時間のかかるのが難点である。ただ高次の多重積分では通常の積分公式よりモンテカルロ法の方が効率がよいといわれている。モンテカルロ法でこの種の問題を解くのは一次の近似解を求めため精密解はこれをもとにして通常の数値解法によって求めるのが普通である。シミュレーションの初期の応用は物理現象の解析、たとえば原子炉内の核反応、中性子の経路のシミュレーション²⁾ 宇宙線シャワーの研究等に用いられた。現在では数理統計学におけるモデルサンプリング (統計実験)、オペレーションズ・リサーチ (OR) の分野における待ち行列、ゲームのシミュレーション等に広く用いられている。特に待ち行列モデルで取り扱える例は非常に多く、電話交換、道路交通網、多重プログラミングおよび多重処理方式の電子計算機システム等のような交通、通信サービス施設およびリアルタイムの情報処理システムの設計、運用のためのシミュレーション、生産ライン、人員配置、在庫管理、建設計画等の生産企業諸部門の設計、運用を目的としたシミュレーションなど広範にわたっている。モンテカルロ法とは種々の複雑な問題に対して必要なパラメータを持つ確率過程を対応させることによって、数値的に所要の統計量を有限回の試行によって計算する方法であるといえる。たとえば所要の量 x が確率変数 ξ の母平均である

とすれば、 ξ の任意標本として $\xi_1, \xi_2, \dots, \xi_n$ なる系列をとることにより

$$\bar{\xi} = \frac{\xi_1 + \xi_2 + \dots + \xi_n}{n}$$

を計算して x の不偏推定量とすればよい。確かに十分大きな n に対しては、大数の法則によりほぼ確率 1 で $\bar{\xi} \approx x$ が成立することが保証されている。このようにモンテカルロ法には実際の問題をある確率過程に対応させるモデル化と任意標本の抽出 (所要の乱数による試行) という二つの手順があるわけであるが、有限回の試行によって解を推定する以上、推定値の分散はさげられない。現在では確率過程を忠実に再現することに主として重点が置かれ、シミュレーションをいかに能率よく (少ない試行で高い精度を得よう) 設計するかについては、まだまだ考慮されていない。モンテカルロ法で偏微分方程式の解や定積分の値を求める場合には標本抽出の効率を高める各種の方法が研究されており、それらは条件付モンテカルロ法 (Conditional Monte Carlo Method)³⁾ と呼ばれている。しかし待ち行列のシミュレーションに対してはごく最近になって取り上げられてきたにすぎない⁴⁾ モンテカルロ法が広く利用されるようになるにつれて必要な乱数の個数も飛躍的に増大した。またモンテカルロ法の実用性が認められるに至ったのも高速電子計算機の発達により容易に大容量の乱数を発生できかつ高速でアクセスできるようになったからである。計算機のサブルーチンとして発生させる乱数は、計算機の語 (word) の取りうる状態が有限であるから必ず周期を持つ。すなわち本当の意味の乱数ではないので擬似乱数 (pseudo random number) と呼ばれる。この周期を実用上さしつかえない程度にできるだけ長くするための方法は整数論を用いて詳しく議論されている⁵⁾ が、その統計的性質が所要の特性を満たすかどうかについては必ずしも明確ではなく、最近今までよく用いられてきた乱数列もその順序統計量 (order statistic) がよくないという報告⁶⁾ があり、実験的にもあまりよい結果が与えられない

ことがある^{6),7)}。したがって用いる乱数については前もって検定を行なう必要がある。

2. 乱数の発生

ある確率過程のシミュレーションには所要の分布法則に従う乱数系列を能率よく発生させることが必要となるが、これは通常一様乱数系列 (区間(0,1)上の一様でかつ独立に分布する点の系列) を変換公式によって指定された分布に変換することによって行なわれる。また特殊な分布たとえば指数分布, ガンマ分布, 正規分布等についてはいっそう能率的な乱数の変換法がある⁵⁾。ここでは二, 三の一様乱数の作り方とその検定結果を示す。

(1) 乱数表を活用する方法

普通乱数表といわれるものは多くは random digit table と呼ぶべきものであり、一様乱数を作るには必要な桁数を順次取り出して、正規化し、区間(0,1)上の点に対応させる。この方法は乱数の検定というやっかいな操作がはぶける大きな利点があるが主として次の二つの理由により従来あまり用いられていない。

(a) 大容量の乱数を計算機の主記憶装置にたくわえておくことは実際的ではなく、テープまたはドラムのような補助記憶装置に格納するとアクセス・タイムが長く演算速度が大幅に低下する。

(b) 乱数表には容量に限度があり、使い尽してしまふ可能性がある。

しかし最近の大型計算機では入出力制御部が中央演算装置と独立に動作し、高速の入出力バッファ・メモリを備える傾向にあるので、二つのバッファ・メモリに交互に乱数系列をたくわえておくことにより (a) の欠点は容易に解決できる。(b) の点については、大型の乱数表を用い、得られる全乱数系列を (3) で述べる合同式法により 1 回目と 2 回目では別の異なる系列に変換することが有効であるとの報告がある⁶⁾。

(2) 物理現象を利用する方法

これは放射線同位元素から出る放射線または熱雑音, ショット雑音のような物理的ゆらぎ現象を用いて乱数 (ただし random bits の系列) 発生装置を構成し、計算機と連動して所期の目的を達しようとするものである。この種の乱数発生装置は回路による特性の偏り, 不安定性がさげられないこと, 再現性がないことのために今ではあまり使われていない。この方法はむしろ機器, 装置のリアルタイム・シミュレータとして重要であろう。

(3) 算術的に発生させる方法

(a) 平方採中法 (Mid-Square Method)

最も古く考えられたもので、2進法または 10進法で S 桁の計算機では S 桁の数を平方し結果の 2S 桁からその中央部分 S 桁を取り出す。同様のことを以下繰り返して乱数系列を作る。この法では途中で具合の悪い値が生ず

ると、たちまち小さい cycle で回り始める。またどのような初期値から出発すればよいか理論的検討がむずかしく、演算時間もかなり長いので、現在ではほとんど用いられていない。

(b) 相加型合同式法 (Fibonacci Series Method)

次に示す合同式により乱数系列 $\{x_n\}$ を作る方法である。

$$x_{n+1} \equiv x_n + x_{n-1} \pmod{M} \quad (1)$$

$x_0=0, x_1=1$ なる初期値から出発するとき、 $\{x_n\}$ は Fibonacci 数列と呼ばれる。乱数系列を発生するのに乗法の代わりに加法を使うので演算速度が早く周期も非常に長い (初期値 x_0, x_1 として M と互いに素のものをとると $M=2^s$ のとき $3 \times 2^{s-1}$ $M=10^s$ の時は $15 \times 10^{s-1}$ ⁶⁾) が、独立性に関する統計的性質が良くないのであまり用いられていない。

(c) 乗算型合同式法 (Multiplicative Congruence Method)

これは次に示す合同式によって乱数系列 $\{x_n\}$ を作る方法であり、広く用いられている。

$$x_{n+1} \equiv \lambda x_n \pmod{M} \quad (2)$$

これは 1949 年 Lehmer によって提案された合同式法の最初のものであり、彼は $\lambda=23, M=10^8+1, x_0=47594118$ を用いた。すなわち $x_n \times 23$ の乗算結果の下位 8 桁より、下より第 9 位以上の部分を引くという方法である。この利点は下位の桁が短周期になるのを、上位の桁を操作することにより防げる点であり、周期は実現できる最大の 5882353 であることが証明されている。

この Lehmer の方法は現在でも広く用いられているが、最近この乱数系列の順序統計量に関する検定結果が良くないと指摘された⁶⁾。(2) 式で M としては 10 進法 S 桁の計算機では 10^s , 2 進法 S 桁なら 2^s に選べば、固定小数点演算で数値語一語の範囲から桁あふれした部分を無視すれば能率よく自動的に mod 演算が行なえるので便利である。 $M=2^s$ の場合は x_0 が奇数, $\lambda \equiv 3$ または $5 \pmod{8}$ ならば (2) 式の数列は可能な最大周期 2^{s-2} を持ち、 $M=10^s$ ならば x_0 が 2, 5 で割れず、かつ $\lambda \equiv \pm (3, 11, 13, 19, 21, 27, 29, 37, 53, 59, 61, 67, 69, 77, 83, 91) \pmod{200}$ ならば最大周期 $5 \times 10^{s-2}$ を持つことが証明されている⁶⁾。また乱数系列が互いに独立に生起するためには、 λ は小さすぎても大きすぎてもいけない。たとえば 2 進法の計算機を考えると、 $\lambda > 2^k$ とすれば、区間 (0,1) を 2^k 個の等区間に分割するとき、 x_n が第 i 区間 ($i=1, \dots, 2^k$) にあることが次にどの区間に x_{n+1} が生起するかについて、なんら情報を提供しないと考えられる。それは λ を乗ずることにより微小部分区間は少くとも (0,1) 区間以上に拡大されるからである。また (2) 式は $\lambda \delta \equiv 1 \pmod{M}$ を満たす δ によって $x_n \equiv \delta x_{n+1} \pmod{M}$ と変形することができ

るので、 $\lambda > 2^a$ かつ $\delta > 2^a$ であれば x_n は x_{n-1} , x_{n+1} のいずれとも相関が小さくなると考えられる。10 進法の計算機でも同様のことがいえるので、 λ としては $\delta \geq \lambda$ の成り立つ範囲でできるだけ大きくなるよう選べばよい。しかし $\lambda \delta \equiv 1 \pmod{M}$ を満たし、 $\delta \geq \lambda$ なるできるだけ大きく同時に最大周期をもたらし λ を選ぶ簡単な方法が見つかっていない。通常は

$$\lambda = \sqrt{M} \quad (3)$$

程度に選ばれることが多い⁸⁾。そうすれば上述の意味でおのおの x_n の上位 $\frac{S}{2}$ 桁は独立に生起すると考えられる。

(d) 混合型合同式法 (Mixed Congruence Method) これは次に示す合同式によって乱数列 $\{x_n\}$ を作り出す方法であり、これも広く用いられている。

$$x_{n+1} \equiv \lambda x_n + \mu \pmod{M} \quad (4)$$

上式で次の条件が満たされているなら周期は最長となり M である。すなわち $\{x_n\}$ は M 個のすべての値をわたることが証明されている⁹⁾。

- (i) μ と M は互いに素
- (ii) すべての M の素因数 p に対して $\lambda \equiv 1 \pmod{p}$
- (iii) 4 が M の因数ならば $\lambda \equiv 1 \pmod{4}$

したがって $M=2^a$ ならば μ が奇数で $\lambda \equiv 1 \pmod{4}$ であればよい。 $\lambda=2^a+1$ ($a \geq 2$) はすべて周期 2^a をもつ。 $M=10^a$ 場合は μ が 2 と 5 で割れず $\lambda \equiv 1 \pmod{20}$ ならばよい。すなわち $\lambda=10^a+1$ ($a \geq 2$) はすべて周期 10^a をもつ。現在 10 進法の計算機 (IBM 650, 7070 等) では

$$x_{n+1} \equiv 101 x_n + 1 \pmod{10^{10}} \quad x_0 = 0 \quad (5)$$

2 進法の計算機 (IBM 704, 709, 7090 等) では

$$x_{n+1} \equiv (2^7+1)x_n + 1 \pmod{2^{35}} \quad x_0 = 0 \quad (6)$$

等が用いられているが、(c) の方法と同様に順序統計量に関する独立性の検定結果があまり満足すべきものでないとの報告がある⁶⁾。

(e) 二つの合同式法を組み合わせる方法 乱数列の順序統計量に関する独立性を改善するために、互に独立に発生する二つの乱数列を組み合わせ、一方の乱数列を他方の乱数で一度シャフルすることにより新しい乱数列を作ることが考えられる⁶⁾。たとえば二つの乱数列 $\{x_n\}$, $\{x'_n\}$ を次に示す合同式により作り出すものとする。

$$x_{n+1} \equiv \lambda x_n \pmod{M} \quad (7)$$

$$x_{n+1}' \equiv \lambda' x_n' \pmod{M} \quad (8)$$

2 進法 S 桁の計算機を考えると、 $M=2^s$ とし、まずメモリ中の 2^n ($n < s$, $n=7$ 程度にとればよいだろう) 個の番地に (7) の乱数列 $(x_1, x_2, \dots, x_{2^n})$ を格納する。次に (8) 式により x'_1 を発生させ、その上位 n ビットによって指定される番地内容を求める乱数 X_1 とし、 X_1 を取り出した番地内容は (7) から得られる次の乱数で

置き換えておく。以下この操作を繰り返すことにより、(7), (8) のいずれの規則性にも関係しない新しい乱数列 $\{X_n\}$ を作ることができる。もちろん 10 進法の計算機にも容易にこの方法は適用できる。この方法によって作った乱数と検定結果を次章で示すが、乱数の独立性に関する統計的性質が単一の合同式法による乱数よりかなり改善されることがわかる。演算速度が遅くなるのが欠点であるが、計算機自体の演算速度が著しく高速化されている現在、正しい乱数列を作るためにはこの方法は有用であると思われる。

3. 乱数の検定

乱数の検定は大別して一様分布に従うことの検定と、独立性の検定の 2 種類に分けることができる。

(1) 一様分布に従うことの検定 (頻度検定 Frequency Test)

区間 $(0, 1)$ をいくつかの小区間に分割し、その各区間に落ちる乱数の度数の一様性を検定する χ^2 検定が普通行なわれるが、数理統計で適合度の検定と呼ばれているものはすべて「分布の一様性」の検定に利用できる。分割区間数は通常 10 または 100 程度にとり、標本の数も少なくとも 10000 個程度は取る必要があり、使用目的と同じ程度の大きさの標本についても検定してみることが望ましい。

(2) 独立性の検定

この方法についても非常に多くの検定法が提案され、それらの検出力についても定説があるとはいえない。ただ従来あまり顧みられていない点であるが乱数列を n -tuple と見て、検定法にも乱数列の順序統計量について敏感なものを採用しなければならない。

(a) 継次検定 (Pairs Test)

一様乱数の相続く 2 数はおのおのを (x, y) 座標にとると、単位正方形平面内の点に対応する。この単位正方形平面を 100 個の等しい小平面に分割し、おのおのの小平面に落ちる乱数 pair の頻度が等しいという仮説を χ^2 検定で検定する。標本の大きさは 10000~50000 程度は取る必要がある。

(b) 上下の連 (Runs up and down)

N 個の一様乱数列 x_1, x_2, \dots, x_N から次のようにして $(N-1)$ ビットの 2 進符号の系列を作る。すなわち第 n 項を $x_n < x_{n+1}$ であれば 1 に、 $x_n > x_{n+1}$ であれば 0 に対応させればよい。このような符号の系列は上下の連と呼ばれており、異符号で区切られた同符号列を連といい、相続く同符号の数を連の長さという。 N 個の乱数列より上下の連を作った場合長さ k の連の出現度数の期待値 $E(r_k)$ およびその分散は次式のように計算されている¹⁰⁾。

$$E(r_k) = 2 \left\{ \frac{N(k^2 + 3k + 1)}{(k+3)!} - \frac{k^3 + 3k^2 - k - 4}{(k+3)!} \right\} \quad (9)$$

$$\sigma^2(r_k) = 2N \left\{ \frac{k^2+3k+1}{(k+3)!} - \frac{2k^5+13k^4+24k^3+(3k+1)}{(k+3)!(k+3)!} \right. \\ \left. - \frac{(k-7)}{(2k+5)!} \frac{2(8k^3+36k^2+46k+14)}{(2k+5)!} \right\} \\ + 2 \left\{ \frac{k^4(3k+11)(k+3)-k(28k^2+101k+50)+4}{(k+3)!(k+3)!} \right. \\ \left. + \frac{2(16k^4+80k^3+116k^2+32k-19)}{(2k+5)!} \right. \\ \left. - \frac{k^3+3k^2-k-4}{(k+3)!} \right\}$$

ただし $k < N-1$ (10)

上下の連を作るにより乱数のサイクリックなふるまいは容易に見出すことができるので、この方法は乱数の検定によく用いられる。

(c) 0.5 上下の連 (Runs above and below the mean)

N 個の乱数の系列 x_1, x_2, \dots, x_N から、 $x_i > 0.5$ なら 1, $x_i < 0.5$ なら 0 ($i=1, 2, \dots, N$) と置くことにより N ビットの 2 進符号の系列を作ることができる。このような符号系列を 0.5 上下の連と呼ぶ。(b) と同様に連の長さを定義すれば長さ k の連の出現度数の期待値 $E(S_k)$ および任意の長さの連の期待値 $E(S)$ は次式で計算することができる⁸⁾。

$$E(S_k) = (N-k+3)2^{-(k+1)} \quad (11)$$

$$E(S) = \frac{N+1}{2} \quad (12)$$

0.5 上下の連を作ることにより、一様乱数列の平均値が急に異なるレベルへシフトするいわゆる population shift などは長さの長いものが多くなることから容易に検知できる。

Table 1 Results of Tests

1. Frequency Test

Test of goodness of fit to rectangular distribution:

(a) 10 intervals (degree of freedom=9).

No. 1 $\chi^2=2.881$ (of all 5,000 numbers)

$\chi^2=7.280$ (of all 10,000 numbers)

No. 2 $\chi^2=11.044$ (of all 10,000 numbers)

No. 3 $\chi^2=7.264$ (of all 10,000 numbers)

Prob. ($\chi^2 > 21.666$) = 0.01

Prob. ($\chi^2 > 16.916$) = 0.05

(b) 20 intervals (d. of f. = 19).

No. 1 $\chi^2=14.616$ (of all 10,000 numbers)

Prob. ($\chi^2 > 36.191$) = 0.01

Prob. ($\chi^2 > 30.144$) = 0.05

2. Confirmation of distribution of certain statistics to expectation:

Statistic	Observed average*	Expected average	Observed variance*	Expected variance
$\sum_i x_i / 128$		0.50000		0.0006510
No. 1	0.50049		0.0006275	
No. 2	0.50171		0.0006953	
No. 3	0.50034		0.0005043	
$\sum_i x_i^2 / 128$		0.33333		0.0006944
No. 1	0.33343		0.0007060	
No. 2	0.33350		0.0007579	
No. 3	0.33327		0.0005705	

* Based on 128 values.

3. Pairs Test

Successive pairs of uniform unumbers were taken as the coordinates of a point in the unit square. The unit square was divided into 100 equal cells. The occurrences in each cell were counted and the χ^2 statistic computed.

Degress of freedom=99.

Test of goodness of fit to rectangular distribution:

No. 1 $\chi^2=89.22$ (of all 5,000 numbers)

$\chi^2=93.45$ (of all 10,000 numbers)

No. 2 $\chi^2=108.5$ (of all 5,000 numbers)

$\chi^2=110.5$ (of all 10,000 numbers)

No. 3 $\chi^2=112.5$ (of all 30,000 numbers)

Prob. ($\chi^2 > 133.3$) = 0.01

Prob. ($\chi^2 > 122.4$) = 0.05

4. Runs up and down

Length of run	1	2	3	4	≥ 5	Any length
Observed average number of runs*						
No. 1	437.8	183.8	52.1	11.6	3.1	688.4
No. 2	418.1	188.9	54.3	12.2	3.1	676.6
No. 3	421.7	188.5	54.2	11.8	3.0	679.2
Expected average number of runs	426.8	187.5	53.9	11.7	2.4	682.3
Observed variance of number of runs*						
No. 1	254.3	75.8	26.0	7.2	5.0	90.1
No. 2	410.2	81.2	39.7	13.3	2.2	185.2
No. 3	484.1	107.4	54.9	9.0	1.4	204.5
Expected variance of number of runs	433.3	115.2	42.8	10.9	2.4	181.0

* Based on 16 sets of 1,024 numbers.

5. Runs above and below the mean

Length of run	1	2	3	4	5	6	7	8	9	≥ 10
observed avge. No. of runs*										
No. 1	257.8	121.5	65.7	33.2	14.9	7.9	3.8	2.3	1.6	0.8
No. 2	251.2	122.1	65.3	32.3	16.1	8.0	3.9	3.3	0.9	1.3
No. 3	255.4	131.0	65.1	30.5	15.4	9.2	3.6	2.3	1.4	0.9
Expected avge. No. of runs	256	128	64	32	16	8	4	2	1	1

* Based on 16 sets of 1,024 numbers.

Table 1 に擬似乱数の検定結果を示す。乱数の発生には比較的演算速度の早い乗算型合同式法を用い、使用した計算機が OKITAC 5090 C システムで数値語は 10 進 12 桁であるから、能率よく mod 演算を行なうために $M=10^{12}$ と定めた。Table 1 で No. 1, No. 2 の乱数列は (2) 式において初期値および乗数をおのおの次のように定めたものである。

$$\text{No. 1 } x_0 = 94496987207, \lambda = 7^{17} = 30513987207 \quad (13)$$

$$\text{No. 2 } x_0 = 1, \lambda = 10^6 + 3 = 1000003 \quad (14)$$

No. 3 の系列は 2 の (3) の (e) で述べたように、No. 1, No. 2 の乱数列を組み合わせて (7) 式には (13) に示した定数を選び (8) 式には (14) の定数を割り当てて作ったものである。まずメモリ中の 100 個の番地に No. 1 の乱数列 x_1, x_2, \dots, x_{100} を格納する。次に No. 2 の乱数列の第 1 項を作り、その上位 2 桁 (00~99 までをとる) によって指定される番地内容を求める乱数列の

X_1 とする. X_1 を読み出した番地内容は No. 1 乱数列から得られる次の乱数によって置き換えておく. 以下この操作を繰り返すことにより No. 3 の乱数列 $\{X_n\}$ を作ることができる. 1 の頻度検定よりいずれの乱数列も一様性はよい (ただ分割区間数をもっと多くする必要があるかもしれない). 2 は 16384 個の乱数列を発生させそれを 128 組のブロックに分け, おおのこのブロック内で平均値および 2 乗平均値とその分散を観測し, 理論値と比較したものである. いずれの乱数列もきわめて良い一致を示している. 3 の継次検定の結果も満足すべき結果を示している. 4, 5 は $16384=128^2$ 個の乱数列を 1024 個の乱数を含む 16 個のブロックに分け, おおのこのブロックの中で長さ k ($k=1, 2, \dots$) の上下の連および 0.5 上下の連を観測し 16 組の出現頻度の平均を取り, 上下の連については平均値の分散を測定したものである. 最後の桁はいずれも 4 捨 5 入してまらめてある. また理論値は (9)~(12) 式より計算した. 理論値と観測値の比較はだいたいよく一致しているが長さ k の上下の連の出現頻度平均値の分散が No. 1 乱数列でかなり理論値とずれが大きい. これが No. 1 を No. 2 の乱数列で一度シャフルして得られた No. 3 の乱数列でかなり改善されることがわかった. 以上の検定法の外に順序統計量に関するものとして次のようなものがある.

(d) Triples Test

一様乱数の相続く 3 個の数をおおのこの座標にとると, 3 乱数一組は単位立方体領域内の点に対応する. この立方体を 1000 個の等しい小立方体領域に分割し, おおのこの小立方体に落ちる乱数の組の出現度数が等しいという仮説を χ^2 検定する. この場合も標本の大きさは 10000~30000 程度は取る必要がある.

(e) Maximum of n Test

x_1, x_2, \dots, x_n がおおのこの独立な一様乱数であるとすれば, 確率変数 $W = \max(x_1, x_2, \dots, x_n)$ の分布関数 $F(x)$ は次式で表わされる¹¹⁾.

$$F(x) = \text{Prob}(W < x) = x^n \quad (0 < x < 1) \quad (15)$$

ゆえに W^n は区間 (0, 1) に一様に分布するはずである. この仮説に基づき, たとえば区間 (0, 1) を 100 等分し, 10000 個程度の W をとり, その各小区間に落ちる頻度の等しいことを χ^2 検定すればよい. n は 2 から 10 まで取れば十分であろう.

この外にもポーカー検定, ギャップ検定などもよく用いられる. 多数の検定法のうちでどれを採用するかは問題であるが, 使用目的 (使用する乱数の数, 求めたい統計量の性質など) によって適切なものをいくつか併用するのが普通である.

4. シミュレーションの精度

有限回の試行により能率よく高い信頼確率で計算値を

求めるためには試行回数と精度との関係をあらかじめ求めておく必要がある. モンテカルロ・シミュレーションの精度は通常中央極限定理を用いて論じられる. すなわちある確率空間に属する確率変数 x_i を定義し, x_i の集合 $\{x_i\}$ が互いに独立で同じ分布に従うときまたは $\{x_i\}$ が互いに異なる分布でもリンドバグ定理が成り立てば, $\{x_n\}$ の平均値を m , 分散を σ^2 とすると $n \geq 10$ なる範囲で次の中央極限定理が成立する.

$$\text{Prob.} \left(a \leq \frac{S_n - nm}{\sqrt{n} \sigma} \leq b \right) \approx \frac{1}{\sqrt{2\pi}} \int_a^b e^{-\frac{x^2}{2}} dx$$

$$\text{ただし } S_n = \sum_{i=1}^n x_i \quad (16)$$

いまある事象 E の生起確率の真の値を p , シミュレーションによる計算値を α とし, 次のように確率変数 x_i を定義する.

$$x_i = \begin{cases} 1 & \text{i th trial } \in E \\ 0 & \text{otherwise} \end{cases} \quad (17)$$

そうすると S_n は n 回の試行における事象の総和を表わし一方 $m=p, \sigma^2=p(1-p), \alpha = \frac{S_n}{n}$ であるから, (16) 式に代入すると

$$\text{Prob.} \left(\frac{|S_n - np|}{\sqrt{np(1-p)}} \leq a \right) \approx \frac{2}{\sqrt{\pi}} \int_0^a e^{-\frac{x^2}{2}} dx \quad (18)$$

ここで $a=2$ の時右辺は 0.950, $a=3$ のとき 0.997 である.

左辺 () 内を変形すると

$$|\alpha - p| \leq \sqrt{\frac{p(1-p)}{n}} a = \frac{\sigma}{\sqrt{n}} a \quad (19)$$

さらに計算値 α の真の値に対する相対誤差を ε とすると

$$\varepsilon \leq \sqrt{\frac{1-p}{np}} a \quad (20)$$

となり相対誤差 ε は試行回数の平方根に逆比例して減少する. また計算値 α より真の値の区間推定を行なうには (19) 式を p について解き得られる次式を用いればよい.

$$\frac{\alpha + \frac{a^2}{2n} - \sqrt{D}}{1 + \frac{a^2}{n}} \leq p \leq \frac{\alpha + \frac{a^2}{2n} + \sqrt{D}}{1 + \frac{a^2}{n}}$$

$$\text{ただし } D = \left(\alpha + \frac{a^2}{2n} \right)^2 - \alpha^2 \left(1 + \frac{a^2}{n} \right) \quad (21)$$

Fig. 1 には (20) 式で $a=3$ の場合 (three sigma rule という) すなわち信頼確率 99.7% で相対誤差をある値以下にするために必要な試行回数を p をパラメータとして示した.

ところが, 待合せ行列モデルのようにシミュレーションの途中の一部の現象から得られる結果がそれまでの系の履歴の影響を受ける場合には, シミュレーションの収束の程度は試行回数 n のすべてについて $\{x_i\}$ が独立であ

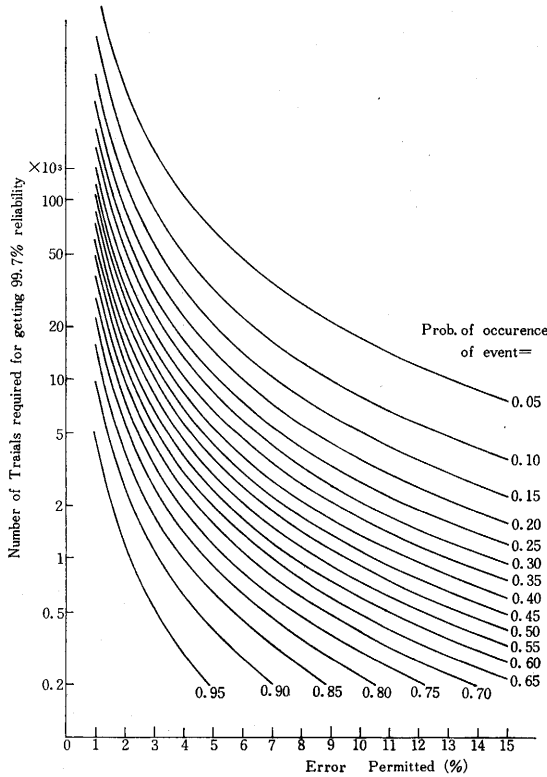


Fig. 1

るとして(16)式を適用したものよりも、緩慢であるといわれている^{4),7)}。したがって精度の推定には互いに相互影響を生じないと考えられる中間結果を独立な標本と考え、 t 分布により推定する実験的方法が採用されている¹²⁾。すなわち i 回目の結果を P_i とすれば、これらが互いに独立で正規分布に従うと仮定し、その k 回の平均を $\bar{P} = \frac{1}{k} \sum_{i=1}^k P_i$ とするとき $t = \frac{(\bar{P} - p)\sqrt{k-1}}{S}$ は自由度 $k-1$ の t 分布 (その確率密度関数を $g_{k-1}(t)$ とする) に従う¹¹⁾。ただし $S^2 = \frac{1}{k} \sum_{i=1}^k (P_i - \bar{P})^2$

$$\text{いま } \int_{|t| \leq t_0} g_{k-1}(t) dt = 1 - \int_{|t| > t_0} g_{k-1}(t) dt = \epsilon \quad (22)$$

によって t_0 を定義すると、次式によって真の値 p の区間推定を行なうことができる。

$$\epsilon = \text{Prob}(|t| \leq t_0) = \text{Prob}(\bar{P} - t_0 S / \sqrt{k-1} \leq p \leq \bar{P} + t_0 S / \sqrt{k-1}) \quad (23)$$

ただこの場合どの程度の試行回数をもって独立な正規分布に従う標本とするかについて理論的根拠が明確でなく、適合度の検定を行なって実験的に標本の大きさを決める必要があろう。

5. 待ち行列モデルのシミュレーションの一例

ここでは待ち行列モデルの一例として Fig. 2 に示すようなパルス分配式波高分析器¹³⁾を取り上げ、シミュレーションの方法について述べる。波高分析器とは、放射

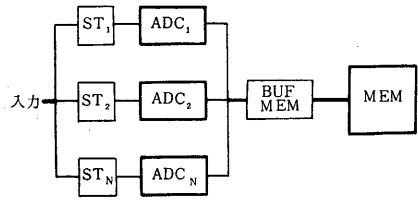


Fig. 2 Model 4

線検出器からのパルス入力の高値を A-D 変換し、同じ波高値を示すものの頻度を記憶装置に集計することにより放射線源のエネルギースペクトルを測定する装置であるが、半減期の短い線源の測定のように被測定量が刻々変化している場合は測定時間を短縮しなければならない。同時に測定値の統計的誤差を少なくするには短時間に大量のデータを処理しなければならない。パルス分配式波高分析器はこの要求を満たすために A-D 変換器 (ADC) を N 組並列に設け実効分解時間を大幅に改善しようとしたものである。次のようなモデル化を行えばシミュレーションによってこの波高分析器の計数損率を計算することができる。

(1) モデル化 (Fig. 2)

- a) 入力の発生は平均頻度 λ のポアソン分布に従い、先着順に処理される。
- b) A-D 変換時間 t_0 は一定または一様分布に従う。ここで一定というのは、比較形 A-D 変換器に相当し、また計数形 A-D 変換器に単一の線スペクトルを持つ (波高一定の) 放射線が入来する場合とも考えられる。一様分布をなすのは計数形 A-D 変換器に平坦な連続スペクトルを持つ放射線が入来する場合に相当する。
- c) 記憶装置 (MEM) の書き込み時間 t_w は一定とする。ここでは N 個の A-D 変換器 (ADC) による多重処理機能の外に、(i) MEM への書き込み中に ADC が引続き占有されないようにするため、バッファメモリを設け書き込みのさいは計数値をこれに移し直ちに A-D 変換器を切り離し、次の変換に移れるようにする。(ii) ADC が書き込み待合せ中には、伸長回路 (ST) を切り離し、独立に次の入来パルスを保持できる。という二つの機能の効果について次の四つの場合に分けて検討する。

- モデル 1 (i)(ii) の機能を共に持たない場合
- モデル 2 (ii) の機能のみを持つ場合
- モデル 3 (i) の機能のみを持つ場合
- モデル 4 (i)(ii) の機能を共に持つ場合

ただしここでモデル 1, 2, 3 は仮想的モデルである。

(2) シミュレーションの方法

シミュレーションを行なうさいには作成されたモデルがシステムを忠実に再現しうるものであると同時に実際のシステムより取扱いが容易でシミュレーション時間も

できるだけ短いものであることが望ましい。シミュレーションは多数回の試行の結果から問題のシステムに関する統計量を観察、記録することであるから状態の時間的な移り変わりだけが実際のシステムに合致するように、シミュレーションプログラム中の状態遷移ルーチンのアルゴリズムを定めればよい。ここで取り扱うシステムは入力が発生も含めて各レベルで各種の装置が並列な動作を行なっているので、これらの状態変化を時間的に直列な現象に変換しなければならない。Fig. 3 にモデル 4 シミュレーションのフローチャートを示す。ADC の取りうる状態は EMPTY (空き), CONV (変換中), WAIT (MEM への書き込み待合せ中), HOLD (待合せ中にさらに伸長回路で保持) の四つであり、MEM は BUSY, NOT. BUSY, の二つの状態を取る。ADC, MEM の状態変化は入力発生、変換終了、書き込み終了によって起こるので、入力発生、時間間隔、変換時間を所要の擬似乱数により指定すれば系の状態推移を観察でき、所要の統計量を求めることができる。すなわち状態変化の起こる時刻を記述する時刻表 (Time Chart) を作成し、この表から最初に起こる事象を探索しその事象に関連した操作を行ない、時刻表を更新しては同様の操作を繰り返せばよい。実際にはこの外に求める統計量 (ここでは計数損率) を集計し、その信頼区間を求め記録するためのルーチンを設け、あらかじめ定められた回数の試行を行なうとシミュレーションを停止するようにした。タイム・スケールは任意に選ぶことができるが本シミュレーションでは $0.1 \mu s$ を時間単位とした。ポアソン到着をするパルスの到着時間間隔は指数分布をなすので、入力発生時刻は (13) 式で決まる一様乱数列 $\{x_n\}$ を次式によって、指数乱数列 $\{y_n\}$ に変換して用いた。

$$y_n = -\frac{1}{\lambda} \ln(x_n) \tag{24}$$

本シミュレーションでは計算値の信頼区間の推定に (20) 式を一応の目安と考え、95% の信頼確率で誤差が 5% 以下になるよう試行回数を選んだ。所要時間は一様乱数の発生に 1 個当たり約 400 ms, 1 実験当たり生起パルス数 2000 で約 15 分であった。Fig. 4・5・6 に得られた結果の一部を示す。平均 A-D 変換時間を t_c とすると、入力トラヒック密度 ρ は $\rho = \lambda t_c$ となる。また 2 段目の影響を評価するために、MEM への書き込み時間と平均 A-D 変換時間との比を $\delta = \frac{t_w}{t_c}$ (通常 < 1) で表わす。

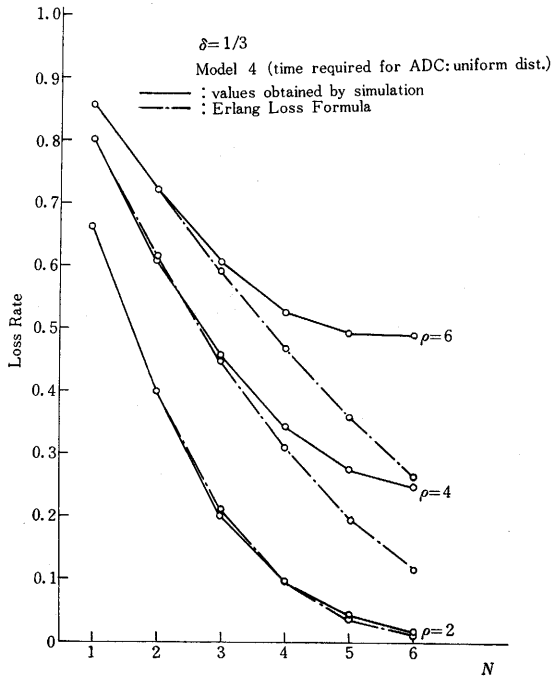


Fig. 4

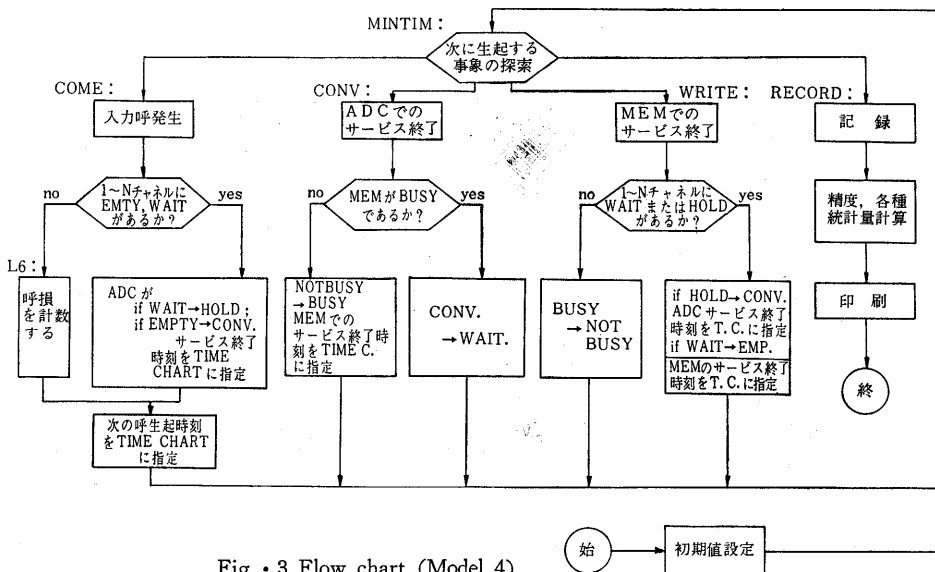


Fig. 3 Flow chart (Model 4)

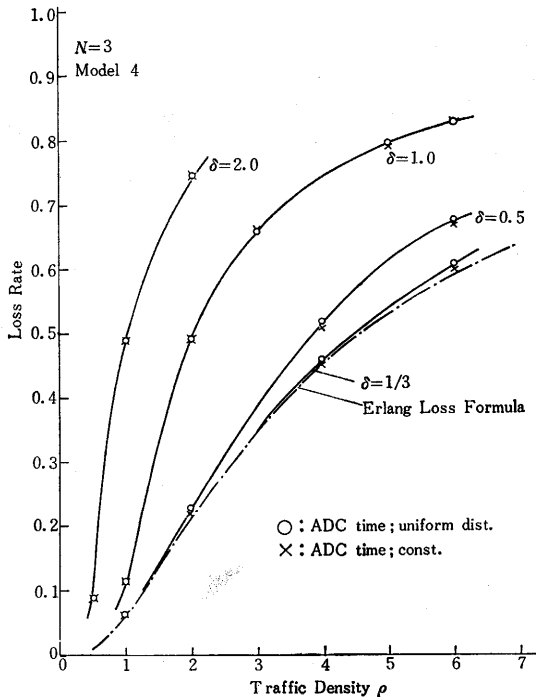


Fig. 5

Fig. 4は A-D 変換時間が一般分布に従うときの多重処理による計数損率 (Loss Rate) の改善効果を示す。N が 4 以下の範囲ではほぼ Erlang の式に従う改善効果を示すが、N が 4 以上になると書き込み待合せの頻度が多くなり改善率は飽和してくる。一般に $\rho\delta < 1$ あるいは $N\delta < 1$ の場合には Erlang の式に従う改善効果を示すが、 $\rho\delta > 1$ かつ $N\delta > 1$ になると 2 段目の影響により改善効果は飽和し、 $N \rightarrow \infty$ で計数損率は $1 - \frac{1}{\rho\delta}$ に漸近する。

Fig. 5 は A-D 変換時間が一定 (分散 0) および一般分布に従う場合を比較したものであり、両分布についての計数損率がシミュレーション精度の範囲内で完全に一致しているのは注目に値する。Fig. 6 は $N=3$ の場合にモデル 1~4 を比較したものであるが、前述の (i) および (ii) の機能を付加することにより計数損率は Erlang の式に近づき、2 段目の処理を待ち合わせることに伴う計数率の低下をきわめて少なくすることができるとわかる。

6. あとがき

モンテカルロ法の難点は計算精度を上げようとすると非常に多くの試行を必要とし、時間のかかる点である。モンテカルロ法を数値解析に応用する場合には少ない試行で高い精度を得るための手法がいくつか提案されているが、待ち行列のシミュレーションをいかに能率よく設計するかについては最近取り上げられてきたにすぎず⁴⁾ 今後の課題であるといえよう。また待ち行列のシミュレーションは多くの共通した特徴を持っているので、プロ

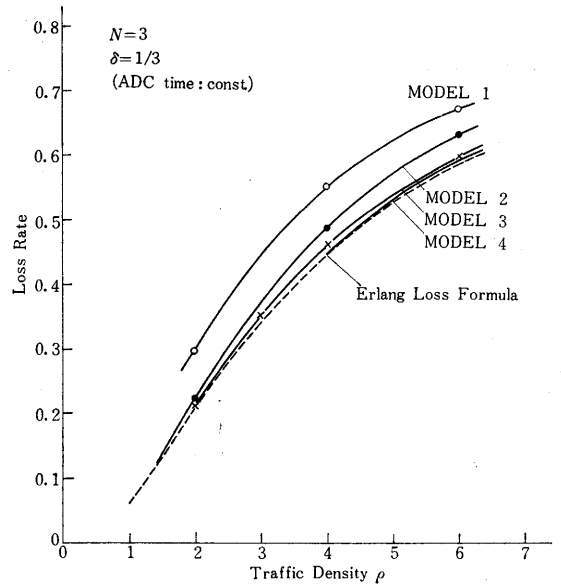


Fig. 6

グラミングを能率よく行なうために多くのシミュレーション用語 (たとえば GPSS, SIMSCRIPT, SOL, DYNAMO 等) が作られている。しかしいずれもそれほど一般的とはいえず、理解も必ずしも容易でなく、今後さらに一般的、実用的なシミュレーション用語の研究が待たれる。

(1966年3月8日受理)

文 献

- 1) Y. A. Shreider: Method of Statistical Testing (Monte Carlo Method) Elsevier Publishing Comp. 1964
- 2) E. D. Cashwell, C. J. Everett: A practical manual on the Monte Carlo Method for random walk problems; Pergamon Press 1959
- 3) H. Kahn: Use of Different Monte Carlo Sampling Techniques; Symposium on Monte Carlo Method, Wiley 1956
- 4) E. S. Page: On Monte Carlo Method in Congestion Problems II (Simulation of Queueing Systems) Journal of OR of America 1965
- 5) 一松, 森口, 山内: 電子計算機のための数値計算法 I 培風館 1965
- 6) M. D. Mac Laren, G. Marsaglia; Uniform Random Number Generators; J. of ACM vol. 12 No. 1 1965
- 7) 藤原, 風間: 電気通信学会, 交換研資 65.4-3 1965
- 8) IBM Corp: Random Number Generation and Testing: Reference Manual C-20 8011
- 9) T. H. Hull, A. R. Dobell: Random number generators; SIAM Review vol. 4, 1962 p. 235-254
- 10) H. Levene: On the Power Function of Test of Randomness based on Runs up and down; AMS vol. 23 1952
- 11) S. S. Wilks: Mathematical Statistics; Princeton Univ. P. 1944
- 12) 鈴木, 高木: クロスバリンク方式についての電子計算機による擬似トラフィック実験; 通研実用化報告 vol. 14 No. 5 1965
- 13) 森脇, 高羽, 瀬沢: パルス分配式多チャネル波高分析器; 電気4学会連大 No. 547 1964