シミュレーションソフトと統計的側面

若山　邦紘

1. はじめに

今回のシミュレーションソフト特集はORソフト特集の第2弾として企画されたものである．これに筆者がOR誌編集委員長のときに，森戸　晋さんにオーガナイザーをお願いしたのであるが，そのつけがまわり，ミイラとりがミイラになった次第である．

シミュレーションモデル，それも特に離散系モデルを計算機のプログラムとして記述しようとするとき，このプログラムが一般の数値計算用プログラムとかなり異なった形式のプログラムとなるため，1950年代の後半にシミュレーションプログラムが簡単に書けるようにGASPという離散系シミュレーションのための FORTRAN サブルーチンが Pritsker によって発表され，まもなくGPSS, SIMSCRIPT といった本格的な離散系のシミュレーション言語が，それぞれIBM, RAND で開発された．しみじみと「オブジェクト指向言語が新しいなんていったって，30数年前に SIMCSCRIPT はオブジェクト指向だったんだもんねえ」とは，当時SIMCSCRIPで大規模なシミュレーションを実施していた小池將貴さん（元三菱電機，現筑波技術短期大学）の言である．小池さんは古い恋人を想い出すようにまだ惚れ込んでいるのです．ただしSIMSCRCIPT は玄人好みであるので素人にはあまり利用されてこなかったのが実情である．

このマニュアルに，シミュレーション言語にはまず問題をモデル化するための世界観があることが重要だとされ，

①シミュレーションの進行を制御するためのタイミングルーチン

②種々の分布に従う乱数の発生

③結果の統計と報告書作成

の機能を備えていなければならないとある．

また離散系の基本的コンセプトが，

①システムの状態変化が離散的な時刻に起こり，

②システムの状態を記述している変数の値が離散的に変化する，

ものであることを認識しておいていただきたい．状態変化をもたらす出来事を事象とかイベントと呼ぶが，事象はある時刻に瞬間的に起こり，その瞬間に状態が変化する，という前提でモデルは記述されるのである．

2. なぜ乱数か？

たとえば，ほかほか弁当屋に客がランダムにやってきて，何人分かの弁当を注文し買って帰るとする．ある客の到着から次の客の到着までの時間間隔が一定ではなくランダムだとすると，その時間間隔をランダムに決めるためのランダムメカニズムが必要になる．また何個買うかを決めるのにもランダムメカニズムがいる．注文をして金を払い弁当を受け取るまでの所要時間もランダムメカニズムで決めたい．このようなランダム要因を含んだシミュレーション分析に乱数を使用するのは無作為にデータを抽出し，偏りの少ない結果を推定しようとする標本調査の応用である．

具体的にはランダムメカニズムをそれぞれの要因について，実績データをもとにして確率分布を決定したり，要因の構造から推察される理論的分布を当てはめたりし，これらの分布に従う乱数を発生させて，ランダムに起こる現象を模擬してゆくのである．

弁当屋にやってくる客の到着時間間隔が指数分布に従っているとか，その客が買ってゆく品物の数がポアソン分布に従っているとか，そのサービス時間がアーラン分布に従っているとかいった場合，シミュレーションの実行中にそれらの分布に従う乱数を引きながら，事象の発生時刻をコントロールしたりシステムの状態を表わしている変数の値を変化させる．現在のところ不確定要素のシミュレーションにおけるランダムメカニズムは乱数が基本である．

そこでシミュレーションソフトは各種の乱数を使いやすい関数の形で用意している．比較的新しい SLAMⅡ と WITNESS で利用できる確率分布を表1に示す．ちなみにSLAMⅡは FORTRAN をベースに，WITNE

わかやま　くにひろ　法政大学　工学部
〒184 小金井市梶野町3－7－2

表 1　用意されている乱数発生用関数

確率分布の種類	SLAMⅡ	WITNESS
[0, 1] 一様分布（実数）	○	○
[a, b] 一様分布（実数）	○	○
（整数）		○
三角分布	○	○
指数分布	○	○
アーラン分布	○	○
ガンマ分布	○	
ワイブル分布	○	
ベータ分布	○	
正規分布	○	○
対数正規分布	○	○
二項分布		○
ポアソン分布	○	○
ユーザー定義の離散型分布	○	○
ユーザー定義の連続型分布		○

SS は C をベースにした言語である．

しかし「乱数を用いてシミュレーションを実行したが結果がどうもおかしい．乱数は信用できるのか」といった問いかけが多い．これは確かに乱数がおかしい場合もあるが，実験のやり方が悪い場合も考えられる．

伏見[2]には16ビット程度のパソコンでは発生される乱数の性質に欠陥があることが解説され，これに対してM系列を用いた乱数発生法がよいことが述べられている．関根，高橋，若山[4] にはいろいろな乱数発生アルゴリズムが詳しく述べられている．いずれにしても，32ビットで演算する計算機なら，通常のシミュレーションでは普通の乱数プログラムの基本として用いられている合同法による乱数に問題はないと考えてよい．

3.　シミュレーションソフトの結果の統計機能

シミュレーションソフト，とりわけ離散系シミュレーションソフトは待ち行列のシミュレータであると言いきっても叱られまい．そこで，システムの中で発生する待ち行列に関する滞留数と滞留時間に関するヒストグラムや平均値，分散，最大値，最小値などの基本的な統計量をプログラムをわざわざ書かなくても自動的に集計してくれるような仕組みが組み込まれている．実際，シミュレーションのプログラムを自分で書くと半分以上がこのような結果の統計を出力するためのステートメントになってしまう．シミュレーションソフトはモデルの基本構造を組み立てるだけで，このような煩雑さからユーザーを解放してくれるのである．

ランダムな実験は，やるたびに結果が異なるであろうから，より高い精度を求めるならば多数回のシミュレーションが必要であろう．また，いろいろなパラメータを変えての実験となるとその組合せは膨大になる．むかし1セットのシミュレーションが終わると何百ページかのアウトプット用紙が出てくるのをよく見たものである．これを見て結果を解析するのは大変な仕事となる．さらに，結果を生のデータそのままで見せるのではなく，誰が見てもわかりやすい形のレポートにまとめる機能が備わっているとありがたい．この稿ではふれないが出力形式のひとつとしてシミュレーションの実行経過を視覚的に訴えるアニメーション機能を含んだシミュレーションソフトが時流となっている．ワークステーションの普及によりモデルの動きの可視化が実用的になってきた．

4.　シミュレーションの観測データのバラツキ

誰しもが観測データをもとにして何らかの推定値を求めるためにシミュレーションを行なっている．簡単な$M/M/1$のシミュレーションから平均待ち時間を次のように計算してみた．

①初期状態は空の状態から始める．

②客の待ち時間を記録し，N人までの待ち時間を平均する．

③Nを5万人まで増やしてゆきながら平均待ち時間を計算する．

④乱数の初期値を変えて実験をくりかえす．

その結果が**図1**である．太線はこの実験から得られた95%信頼区間を示す．つまり「平均待ち時間はこの幅の中に入るだろう」といえば，95%はあたること（ひっくり返せば当たらない確率が5%）を意味する．信頼区間の幅が広すぎて満足できなければ人数をもっと増やさなければならない．実験回数を増やせないとすれば，あたる確率を犠牲にした信頼区間を示すしかない．

図2はこの$M/M/1$の理論的解析からわかっている待ち時間の平均μと分散σ^2をもつ母集団から互いに独立

なデータ $x_1, x_2, \cdots, x_N$ を抽出し，この実験をくりかえした場合のグラフである．周知のように

$$\bar{x}=(x_1+x_2+\cdots+x_N)/N$$

$$s^2=\frac{1}{N-1}\sum_{i=1}^{n}(x_i-\bar{x})^2$$

としたとき，と s^2 は μ と σ^2 の不偏推定量であるから，95%信頼区間は

$$\bar{x}\pm 2s/\sqrt{N}$$

と表わされる．つまり，信頼区間の幅はデータ数のルートに反比例する．別の言い方をすれば，推定値の精度を現在より1桁あげようとするならば，現在の100倍の実験が必要となる．これをルート法則という．この事実から，シミュレーションであまり精度にこだわると莫大な計算時間を覚悟しなければならない．

グラフを一見すれば，平均待ち時間の信頼区間の幅は独立なデータの平均の場合に比べてはるかに広い．じつはわれわれのシミュレーションにおいて，i 番目の客の待ち時間を w_i，サービス時間を s_i，$t_{i,i+1}$ を次の客との到着間隔とすると，

$$w_{i+1}=\max\ (w_i+s_i-t_{i,i+1},\ 0)$$

という関係があるので，自分の前の客の待ち時間が長いと自分の待ち時間も長くなることが予想され，事実 w_i と w_{i+1} には強い正の相関がある（図3参照）このように1つの時系列のデータの間に相関があるとき，この時系列には自己相関があるという．w_i と w_{i-k} 相関をこの実験のデータから計算すると図4のようなグラフになる．このカーブを自己相関関数と呼び，$E(w_i)=\mu$，$V(w_i)=\tau^2$ とし，自己相関のある待ち時間を $w_1, w_2, \cdots, w_N$ とすると，自己相関関数は $r(k)=\mathrm{Cov}(w_i, w_{i+k})/\tau^2$ と書くことができる．そして，平均値は独立な場合と同じく

$$\bar{w}=(w_1+w_2+\cdots+w_N)/N$$

によって推定できるが，分散の推定量は τ^2/N とはならず，

$$V(\bar{w})\doteqdot\frac{\tau^2}{N}\{1+2\sum_{k=1}^{N-1}(1-\frac{k}{N})r(k)\}$$

と表わされる．$r(k)=0$ ならば独立な場合そのものであるが，正の自己相関がある場合は $r(k)>0$ であり，{ }の中の値を g と書くと相関のない場合に N 個のデータから得られるものと同じ精度を実現するためには，gN 個の

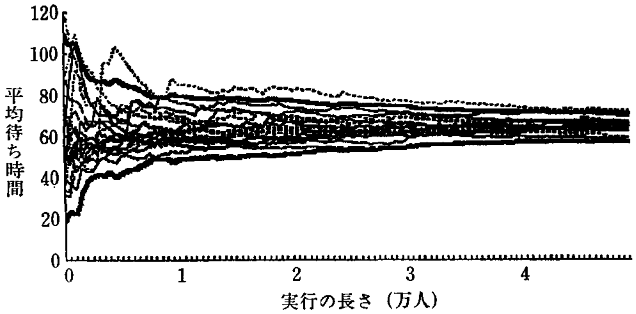

図 1　平均待ち時間の変化の様子

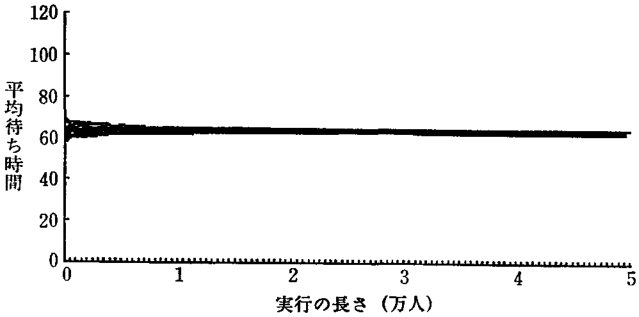

図 2　独立なサンプルの場合

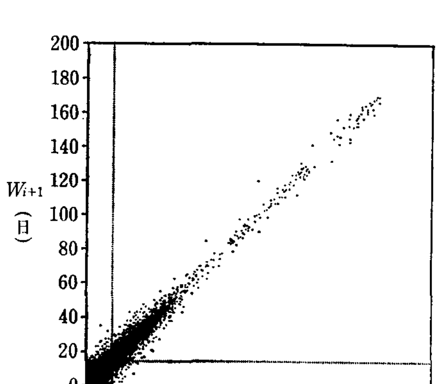

図 3　引き続く客の待ち時間の相関関係

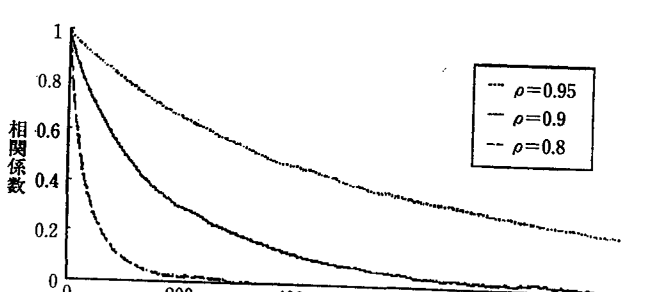

図 4　$M/M/1$ の自己相関関数

表 2　相関による標本数への影響

混雑率　ρ	g
0.8	75.09
0.9	330.85
0.95	1268.48

データが必要となる．待ち行列のシミュレーション結果から得られたgの値は表2のような結果になる．混雑がはげしい場合にはその影響の大きさに驚く．

われわれは標本調査におけるランダムサンプリングの考えをとり入れ，乱数を各所に使ったのであるが，実行の途中でシステムの状態が一度偏ったところにいってしまうと，しばらくはその影響が続き，観測データは独立性を失ってしまうというまことに厄介なもの，それが確率的なシステムシミュレーションの宿命といえる．

5.　独立なデータの抽出法

このようなシミュレーションのデータの観測法に工夫をこらして，独立なデータを得るための考えについてふれよう．

簡単な方法でブロック形成法という方法がある．自己相関の影響を打ち消すために観測データをまとめてブロックにして，それぞれのブロックの平均値をあらたなデータと見れば，これらは独立な観測値となる．ブロックの作り方にはいろいろ考えられているが，いちばん簡単な方法は，乱数の初期値を変えてその実験ごとに平均待ち時間を計算したものをデータすれば，これらのデータは互いに独立になる．しかし，この実験では窓口が空の状態からシミュレーションを始めているから$w_0=0$である．したがって，はじめのうちの待ち時間の値は$w_0=0$の影響を受けているはずだから小さい値に偏ったものであろうことが予想される．そこで，この影響が無視できるくらいまでの待ち時間データは捨てて，それ以後の観測値から平均待ち時間を計算するのがよいといわれている．これをシステムのウォームアップという．それぞれの実験でウォームアップが必要で捨てるデータ量が馬鹿にならない．

そこで，1回目の実験の終了時の状態は定常状態からのランダムなサンプルと考え，これを2回目の実験の初期状態とすればウォームアップの無駄が省ける．これは

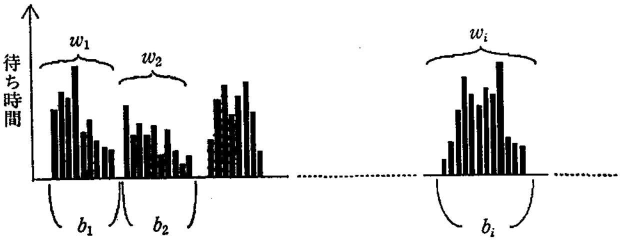

図 5　再生的過程を利用したサンプリング

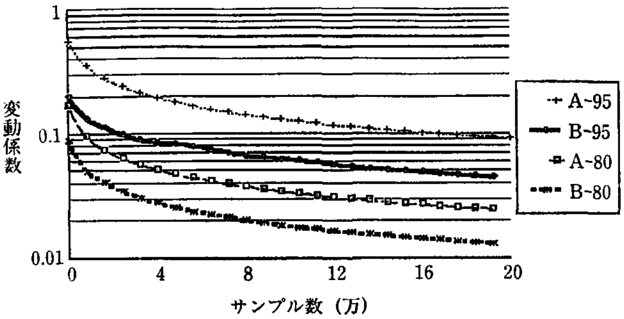

図 6　再生過程を利用したサンプリング方法の精度（Aが自己相関関数による方法，Bが再生的過程を利用した方法）

1回の長い実験を分割したものと考えてよい．ただし，第1ブロックの終了状態と第2ブロックから得られるデータは相関をもってしまうことになるので，独立と見なせるようにするには，ブロックの大きさを十分に大きくとるか，ブロック間のデータを捨てるなどしてブロック間の独立性をできるだけ保つようにするなどが考えられる．しかし，全体の実験の長さが一定だとした場合，ウォームアップもブロック間のデータを捨てることもやめて，ブロック間の独立性が保てるようなできるだけ小さいブロックに分けてできるだけ数多くのデータを得るのが賢明な策である．

ブロック形成による方法では，自己相関の高い系列に対してはブロックサイズを大きくしなければならないから，独立なデータの個数が少なくなるために推定値の分散を小さくしようとした当初の目的が果たせなくなる．

待ち行列はしばしば空になる．この時の待ち時間は0である．これ以前の状態はこれ以後の状態には影響を与えないから，待ち時間ゼロを境にブロックを形成すれば各ブロックの挙動は互いに独立である．これを「再生過程を利用したサンプリング」と呼ぶ（図5参照）．この場合，ブロックサイズは異なるから，平均値の推定には加重平均を計算する．

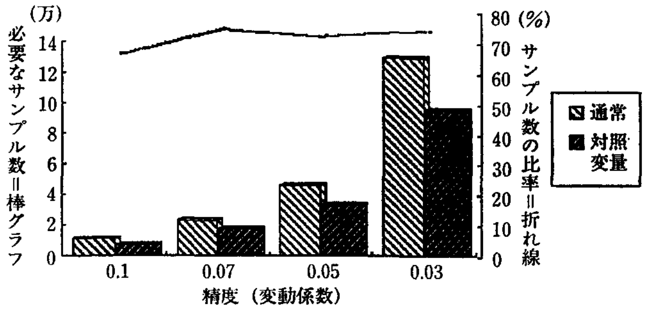

図 7　精度の達成のために必要なサンプル数の比較（$\rho=0.8$）

$$\bar{w}=(b_1w_1+b_2w_2+\cdots+b_Mw_M)/N$$

によって推定できるが，分散の推定量は，

$$V(\bar{w})=\sum_{i=1}^{M}\frac{b_i\tau^2}{N}\{1+2\sum_{k=1}^{b_i-1}(1-\frac{k}{b_i})r(k)\}$$

となる．

図 6 は再生過程を利用した場合の精度（変動係数で表わしている）を自己相関関数による方法と比較したものである．この方法は大変に小さい分散の推定値を与える．同じ精度でよいのなら実験回数は約 1 ／ 3 ですむ．しかし，複雑なモデルでは再生点を定義しても，システムの状態が再生点になかなか戻ってこないことになってしまうので，うまく使える場合が少ないのである．

6.　分散減少法の適用

$M/M/1$ のシミュレーションにおける対照変量法（負相関法）の適用を考えてみよう．この方法はモデルへ負の相関をもつような入力をいれたとき，モデルから負の相関をもつ出力が得られるならば，これらの出力の平均値は分散が小さくなることを利用したものである．すなかち，相関をもつ 2 つの確率変数の平均値の分散は

$$V(\overline{x+y})=\frac{2}{\sigma^2}(1+\rho)$$

となるので，$\rho<0$ の場合は独立なデータを平均したものより推定値の精度が高い．

もっとも簡単なのは，もとの乱数列が $\{r_1, r_2, \cdots, r_N\}$ としたとき，$\{r_1, r_2, \cdots, r_N\}$ を乱数列とするシミュレーションを実行すると結果には負の相関が期待できるかも知れない．しかし，モデルの中で乱数同士が複合的に作用して，出力の値に負の相関が出てこないことがあるので必ずしもうまくゆくとは限らない．$M/M/1$ の待ち時間シミュレーションでは，

$$w_{i+1}=\max(w_i+s_i-t_{i,i+1}, 0)$$

において，$u_i=s_i-t_{i,i+1}$ の分布に従う乱数を発生させるようにすれば，乱数を 1 種類しか使わないので対照変量法が効果的に働く．**図 7** は通常の 2 種類の乱数を使った場合と比較したものである．

7.　おわりに

数多いシミュレーションソフトが出回って，計算機プログラミングを知らない人でもモデルが作れるようになってしまった．だからシミュレーション結果はどんどん出てくる．しかし，シミュレーションの結果から「何がわかるのか」[1]については，相変わらず何の面倒も見てはくれずユーザーまかせ.「シミュレーション公害が出なければよいが」と危惧するのは筆者ばかりではあるまい．

参 考 文 献

［1］　逆瀬川浩孝「シミュレーションで何がわかるのか」ＯＲ誌**32**，11

［2］　伏見　正則「出力結果の解析」ＯＲ誌**35**，2

［3］　伏見　他「特集：乱数とその応用」ＯＲ誌**36**，12

［4］　関根，高橋，若山「シミュレーション」日科技連，1976