論　文

並列処理によるモンテカルロ法の実行

— 金融工学分野への応用の試み —

譚　康融

1．まえがき

モンテカルロ法とは，数値シミュレーションの1つの方法で，多くの研究分野で用いられている。特に，すっきりした解析式や・解（Explicit expression）などが求められにくい場合，その優れた力が発揮される。要は，様々な確率過程をコンピュータ上で生成させることによって，システムの状態を少しずつ更新していき，十分な回数で更新を繰返したあと，システムの最終状態から，平均値を用いてシステムを評価するという手法である。その応用は，統計物理から，数値計算や，原子炉の設計最適化などの最適化問題まで，多くの分野に適用されている。

しかしながら，モンテカルロ法の実行にあたっては，コンピュータリソースが大量に食われて，コンピュータに膨大な計算負荷がかかり，結果を得るまでの待ち時間が相当長いといった問題点もよく指摘され，モンテカルロ法の応用のボトルネック（Bottle neck）となっている。本論文は，これらの問題を解決するために，並列処理を導入し，並列処理による実行時間の短縮法を考案し，特に金融工学分野の実践問題を取り上げて，並列処理法に適用する。さらに従来の非並列処理に比べるため，例題では，同じ時間を費す並列処理による計算結果との精度比較を行う。並列処理の効率性について考察した。本論文は，以下のように構成さ

れている。2，3においては，それぞれ，モンテカルロ法，並列処理のコンセプトについて述べ，4においては，計量ファイナンスの実問題を取り上げ，並列処理によるモンテカルロ法の実行（Parallel processing）の手法を提案し，それらの結果を示す。

2．モンテカルロ法

2.1. モンテカルロ法

モンテカルロ法において，1次元の積分の近似計算という例は，よく知られているが，特に積分の解析式がとっても煩わしい場合や，簡単に解析的な結果が得られない場合などにおいて，モンテカルロ法はその力を発揮し，計算結果を高い精度で近似·推定することができる。もちろん，1次元の積分問題だけではなく，n次元の積分問題まで拡張しても適用する。要するに，n次元の積分の空間を多数個の多次元空間に分けて，各多次元空間において，独立，且つ均一分布のランダムなサンプルを生成して，それらを用いて関数値を計算し，それらの平均を積分の近似値として用いる。

積分問題以外にも，いろんな応用例が挙げられる。例えば，金融工学の分野でも取り上げる問題として，確率関数のパースをシミュレーションし，その値を推定する場合がある。以下の例題では，ヨーロッピアンコールオプションの評価をはじめ，いくつかのモンテカルロ法に適用できる金融工学の問題を取り上げる。

［例題1］ヨーロッピアンコールオプションの評価問題。すなわち，

$$C = e^{-rT}\hat{E}\{f(S_T)\}$$
$$C = e^{-rT}\{\max\{S_T - X, 0\}\}$$

となり，ここで，Eは期待値であり，Xは行使価格，Tを満期期間，rをリスク

レス利息率となる。この問題を評価するため，ここで，ランダムな乱数を生成して，価格変化のパースを模擬し，それらの平均値をオプションの価格として近似することができる。

［例題2］バリュー･アット･リスク（Value at Risk）の推定。実際の投資問題として，ポートフォリオのバリュー･アット･リスクを推定しなければならない。理論上は，正規分布の仮定は，もっとも都合がいいが，実際の金融市場の収益率の分布は，ファットテール（Fat tail）の分布を呈しているケースが数多く観察されている。しかも，正規分布ではなく，Student-t ではない場合がある。さらに，資産間の相関もありうる。そのとき，ポートフォリオのバリュー･アット･リスクを推定するには，モンテカルロ法が用いられると，より精度の高い結果が得られる。

［例題3］アジアオプションのプライシング。アジアオプションは，パースに依存したオプションである。そのペイオフは，いままで経過したアセット価格の平均値に依存する。理論上で陽的な数式を用いて表現できないが，モンテカルロ法を適用することによりオプションへの評価ができる。

金融工学におけるモンテカルロ法の利用により，変数の取りうる範囲が広くなり，線形，あるいは非線形リスク，相関問題，非正規分布など，あらゆる問題に取り込め，シミュレートすることができる一方，モデルリスクについて，他の方法で確認して補完する必要性がある。また，シミュレーションの結果について，その解釈は，解析式のように素直に解釈できない場合が多々あることも承知しなければならない。

2.2. 乱数の生成法

モンテカルロ法を実施する際に当たっては，もっとも大事なことは，いかに効率的にしかも質の高い乱数を作り出すかという点に集約する。一般的にコンピュー

タ乱数は，予め決められていた数学のメカニズムによって，作られているものであり，擬似乱数（Pseudo random numbers）と呼ばれる。従って，一定周期をもつことになる。乱数生成によく使われている方法は，以下のいくつかの方法であろう。1）線形同歩法，2）反転法，3）受容—拒否法などの方法が挙げられる。いうまでもないが，周期が長ければ長いほどよい乱数となり，より精度の高い結果が得られる。

また，同一生成法において，同じパラメータで違う乱数列を得たい場合は，毎回，違う乱数のシーズ（Seeds）をセットする必要がある。シーズとなるものは，コンピュータで乱数を生成する場合は，コンピュータに内蔵しているクッロクを採用している場合がほとんどである。FORTRAN，PASCAL，SAS，SPLUS などのコンピュータ言語，あるいは統計パッケージでは，利用者が自らシーズをセットする機能が設けられている。

さらに，質のよい乱数を作り出すために，最初生成した乱数の分散や，相関関係などを調整するテクニックが求められる。

一般的に，1）対照変量法（Antithetic variates），2）共同乱数法（Common random numbers），3）制御変量法（Control variates），4）層化抽出標本法（Stratified sampling），5）加重サンプリング法（Importance Sampling）などがよく採用されている。ここで，次の例題で利用される対照変量法について簡単に説明する。

いま，2組の乱数 (X_i^1, X_i^2) があるとし，

$$X_i = \frac{1}{2}(X_i^1 + X_i^2)$$

とすると，

$$Var(X_i)/n = (Var(X_i^1) + Var(X_i^2) + 2Cov(X_i^1, X_i^2))/4n$$

となり，ここで，共分散の項をマイナスにすれば，全体の分散が小さくなることが明らかである。

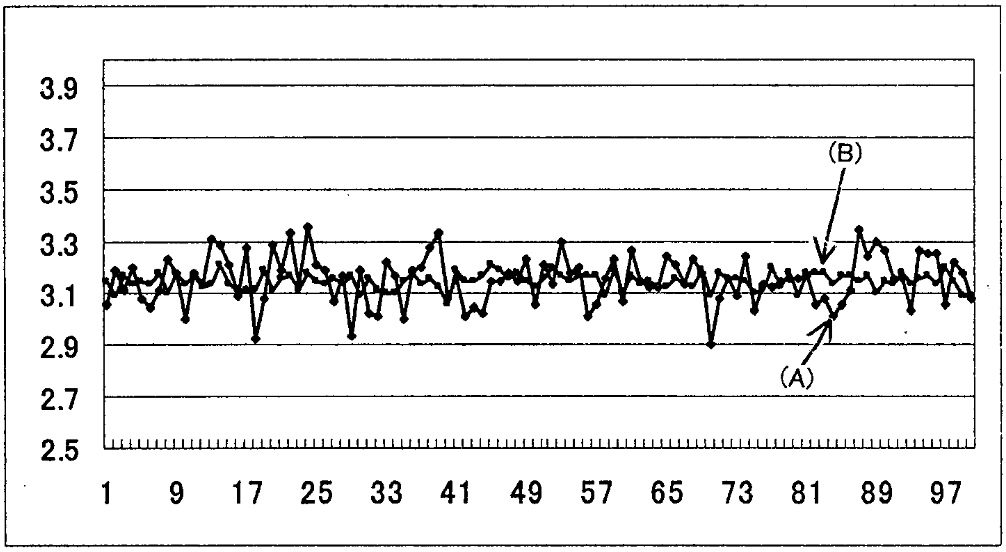

図1　円周率の値の比較

ここで，簡単な計算をした結果を図1にまとめている。対照変量法を使わないとき(A)と使うとき(B)において，それぞれ得られた円周率の値の比較である。図1から，分かるように，対照変量法を用いた円周率への推定のばらつき（標準偏差）は，対照変量法を使わなかったほうよりも，だいぶん小さくなっていることが明らかである。

2.3. 準モンテカルロ法

準モンテカルロ法（Quasi-Monte Carlo Method）は，従来のモンテカルロ法に比べて，擬似乱数（Pseudo random numbers）の代わりに準乱数列（Quasi-random sequences，あるいはLow-discrepancy sequences）を生成してそれを用いてシミュレーションをするというところがモンテカルロ法と違う。実際にこの準乱数列の値は，すべて決まっており，モンテカルロ法のようなランダム的なも

のではない。いままでの研究から，準モンテカルロ法はシミュレーションの精度を向上させ，あるいは計算コストを低下させるなどといった利点が確認されている。ファイナンスの例は，ヨーロッピアンコールオプションの問題を準モンテカルロ法を用いて解くことができる。経験的な結果から見ると，準モンテカルロ法は，モンテカルロ法よりも速く収束し，且つより高い精度の解を得ることができる。

3．並列処理

並列処理（Parallel processing，あるいは Parallel computation）とは，複数の PE（Processor element）による計算機ジョブ（Job），あるいはタスク（Task）の実行のことである。本論文では，モンテカルロ法，あるいは準モンテカルロ法の結果は，この並列処理によって得られたものである。従来のコンピュータは，シリアル処理はほとんどであるため，大量計算に向いていないのに対して，スーパーコンピュータは，PE 間の通信をやり取りしながらも，各 PE が独立で計算タスクを完成することができる。現段階では，並列処理に向いているプログラミング言語は，C や，FORTRAN95 などが挙げられる。一般的には，プログラムの中で使用する PE 数を指定しなければならない。

さらに，並列処理を行いたいループについても，文で示さなければならない。すべての例題において，並列プログラムを示すとかなりのスペースが必要になるため，ここでは，FORTRAN でつくったプログラムの一例だけを示しておく。

［例題］円周率の推定

```
      PROGRAM MAINF90
      PARAMETER (NMAX=10000)
      DOUBLE PRECISION A (NMAX, NMAX)
      REAL*8 SSUM, PI
!XOCL PROCESSOR P(4)
!XOCL INDEX PARTITION IP=(PROC=P, INDEX=1, NMAX, PART=
      BAND)
!XOCL LOCAL A(:, / IP)
      CALL DP_VRAU4 (IX, DA, IN, DWORK, NWORK, ICON)
!XOCL PARALLEL REGION
!XOCL SPREAD DO/IP
      DO J=1, NMAX
      DO I=1, NMAX
      IF (X (I) **2+Y (I) **2.LE. 1.0) SSUM=SSUM+1
      ENDDO
      ENDDO
!XOCL END SPREAD
!XOCL END PARALLEL
!XOCL UNIFY(ID)
!XOCL MOVEWAIT SUM(SSUM)
      PI=SSUM/NMAX
      WRITE (6, *) PI
      END
```

図2　並列処理プログラム

上記リストの中の !XOCL 文は，並列計算を行う場合を指定している。文 !XOCL PARALLEL REGION と !XOCL SPREAD DO/IP および !XOCL END SPREAD と !XOCL END PARALLEL にはさまれているプログラムの部分は，並列処理が施される。冒頭の !XOCL PROCESSOR P(4)は，プログラムのなかで使用する PE の数を指定している。

並列処理の長所としては，複数の PE の並行作業によって，プログラムの実行スピードが従来の非並列機より桁数の違うほど速くすることである。従来，何日間，何ヶ月間もかかる計算時間は，数時間ほど短くなるのである。いままで実用

に至らなかった理論・手法，例えば，モンテカルロ法を用いたバリュー・アット・リスクの推定とかの手法は実際の業務に応用することができるようになり，リスク管理のレベルを一層アップすることにつながる。一方，短所としては，並列処理を行うために，スーパーコンピュータを導入しなければならないが，予測精度の高いシミュレーション結果を得られるかわりに，運営コストも高くなる。また，多数のPCをネットワークに接続し，並列処理を行うことも可能であるが，リスク管理などのシミュレーションに使われている例は，まだ報告されていないし，オペレーティングシステム（Operating system）などのシステム安定性もスーパーコンピュータより劣り，迅速なリスク管理に向いていない。システムの導入に当たっては，計算コストとパフォーマンスとのバランスの取れたシステムが最も望ましい。さらに，並列処理のプログラムは，開発ツールを用いて自動的に生成することはまだできないため，ユーザが自ら書くことを前提としているのが現状である。

4．金融工学への応用

本節では，具体的な計量ファイナンスの問題を取り上げ，モンテカルロ法と準モンテカルロ法を用いてシミュレーションを行い，その結果のよさについて検討する。また並列処理と非並列処理の計算精度比較をも行う。

［例題４］幾何ブラウン運動（Geometric Brownian Motion)。このモデルを用いた，株価の変動は以下のような式で表せる。すなわち，

$$dS_t = \mu S_t dt + \sigma S_t dz \qquad (1)$$

ここで，S_t は株価で，μ 及び σ は常数項で，dz は標準ウィンナー過程である。ゆえに，

$$d\ln S = (\mu - \frac{1}{2}\sigma^2)dt + \sigma dz \qquad (2)$$

$$S(t+\delta t) = S(t)\exp(v\delta t + \sigma\sqrt{\delta t}\,\varepsilon) \qquad (3)$$

という式が得られる。この式に基づいて，シミュレーションすると，図3のような結果が得られた。

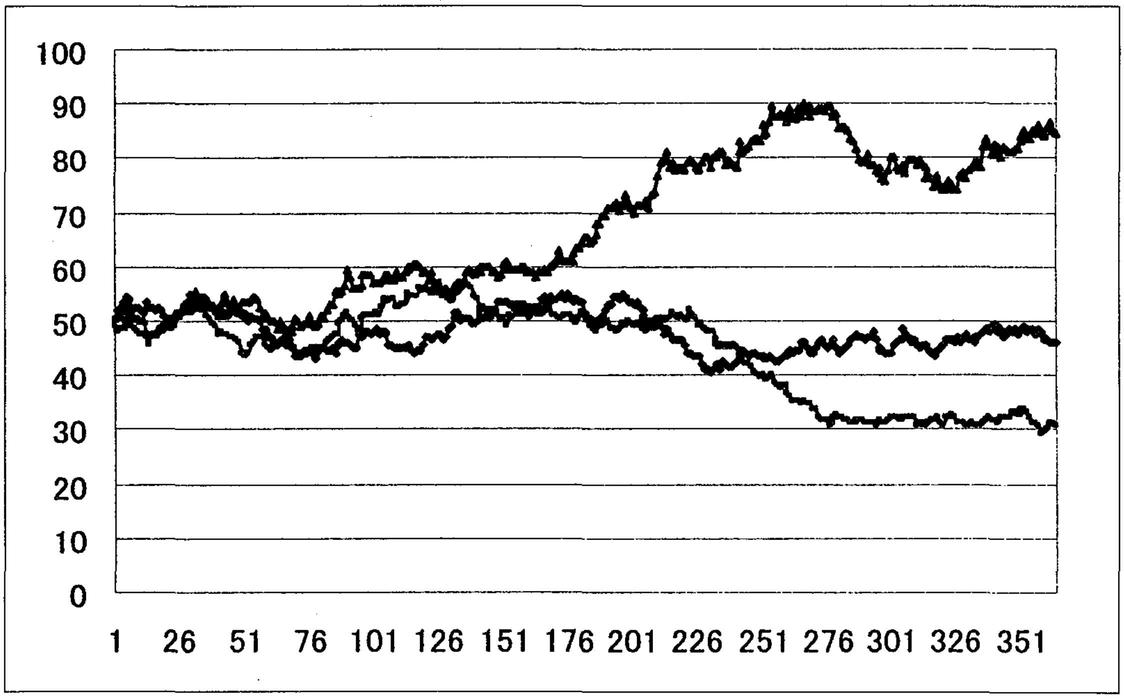

図3

［例題5］Black-Scholes公式の変わりにコール，乃至はプットオプションに適用できる。ここで，オプションの評価は，fの公式になるが，

$$f = E^*[e^{-rT}F(S_T)] \qquad (4)$$

すなわち，

$$\max\{0,\ S(0)e^{(r-\sigma^2/2)T+\sigma\sqrt{T}\varepsilon} - X\} \qquad (5)$$

となる。この式を用いたシミュレーションの結果は，BSの公式の結果は，

5.1911であるのに対して，5.1923となる。

［例題 6］アジアオプションの評価。ここで，Sが幾何ブラウン運動に従うとする。すると，N期になるときのペイオフは，次の式によって，計算される。すなわち，

$$\max\left\{\frac{1}{N}\sum_{i=1}^{N}S(t_i)-X,\ 0\right\} \tag{6}$$

となる。シミュレーションの結果は，100期経過したところ，評価値は135.12となった。

終わりに並列処理と非並列処理との比較例を挙げる。

［例題 7］コレスキー分解によるVaRの推定。いま，乱数Xを生成し，X～N（μ，Σ）ように制約を付ける。ここで $\Sigma = CC^T$ とすると

$$\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix} + \begin{bmatrix} c_{11} & & & \\ c_{21} & c_{22} & & \\ \vdots & \vdots & \ddots & \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_n \end{bmatrix} \tag{7}$$

ここで，

$$\begin{bmatrix} c_{11} & & & \\ c_{21} & c_{22} & & \\ \vdots & \vdots & \ddots & \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix} \begin{bmatrix} c_{11} & c_{21} & \cdots & c_{n1} \\ & c_{22} & \cdots & c_{n2} \\ & & \ddots & \vdots \\ & & & c_{nn} \end{bmatrix} = \Sigma \tag{8}$$

であるため，

$$c_{ij} = (\Sigma_{ij} - \sum_{k=1}^{j-1} c_{ik}c_{jk})/c_{jj}, \quad j < i \tag{9}$$

および

$$c_{ii} = \sqrt{\left(\Sigma_{ii} - \sum_{k=1}^{i-1} c_{ik}^2 \right)} \tag{10}$$

が得られる。

ここで，並列処理によるバリュー･アット･リスクの推定を試みる。一般的にVaRの定義は以下のようになる。すなわち，

$$\mathrm{P}\{\mathrm{L} > x_p\} = \mathrm{p} \tag{11}$$

となる。一般的に，この確率pの値が既知（Given）であり，1％あるいは5％などの値を取る。実際のシミュレーションにより，x_p を推定するケースがほとんどである。精確な x_p を推定するため，かなりの大量計算をしなければならないが，時間とコストの問題があり，効率的な手法ではない。ここで，まず加重サンプリングの手法をVaRの計算に適用し，その上で並列処理によってVaRの推定を行う。

加重サンプリングをVaRに適用すると，

$$\mathrm{P}\{\mathrm{L} > x\} = \mathrm{E}^*[\{\mathrm{L} > x\} \mathrm{l}(\mathrm{Z})] \tag{12}$$

となり，E^* は加重サンプリングの分布下の期待値である。ここで，

$$\mathrm{l}(\mathrm{Z}) = \frac{\exp\{-\frac{1}{2}\mathrm{Z}'\mathrm{Z}\}}{|\mathrm{B}|^{-\frac{1}{2}} \exp\{-\frac{1}{2}(\mathrm{Z}-\mu)'\mathrm{B}^{-1}(\mathrm{Z}-\mu)\}} \tag{13}$$

となり，l(Z) は尤度比である。シミュレーションに用いられた乱数は，コレスキー分解によるものである。同一ポートフォリオについて，非並列処理及び並列処理をそれぞれ20分にわたってシミュレーションを行った。その結果は表1に示されている。

表 1　非並列処理と並列処理との精度比較

形　　態	P{L ＞ x}	標準偏差比
並列処理（PE ＝ 4）	1 %	約　93
並列処理（PE ＝ 8）	1 %	約 128

以上の結果から分かるように，PE 数の増加によって，結果の精度（標準偏差）が向上されている。

また，準モンテカルロ法と従来のモンテカルロ法によるヨーロッピアンオプションの評価は以下のようになる。

表 2　シミュレーション誤差

方法＼回数	3000	5000	10000
QMC	−0.88%	−0.513%	−0.283%
MC	4.278%	−1.092%	−0.835%

表 2 から分かるように，準モンテカルロ法（QMC）とモンテカルロ法（MC）によるシミュレーションの誤差は，QMC のほうが MC に比べるとかなり小さい。同じ計算回数でも QMC が速く収束することが明らかである。

5．むすび

以上の分析から分かるように，モンテカルロ法・準モンテカルロ法を用いて，数量ファイナンスの実問題に適用することによって，従来の解析的な解が求められにくい，あるいは求められないハイブリッド金融商品への適切な評価ができるようになり，さらに並列処理によって，従来のモンテカルロ法の実行に所要計算時間が大幅に減少し，精度がアップすることが確認された。リスク管理などの実

問題に応用することによって，より精度の高い評価，あるいは予測値を得ることができる。ただ，シミュレーションにおいて収束であるかどうか，その判別はときには判断しにくい場合がある。それについては今後の研究に取り込みたいと思います。

参考文献

1) Feller, W., An Introduction to Probability Theory and Its Applications, 1968.
2) Ross, S., An Introduction to muthematical Finance, Cambridge University Press, 1999.
3) Campell, J., The Econometrics of Financial Market, Princeton University Press, 1997.
4) Nelson, C., "Predictable Stock Returns: The Role of Small Sample Bias", Journal of Finance, 48, 641-661, 1993.
5) Glasserman, P., "Asymtotically optimal importance sampling and stratification for pricing path-dependent options, Math, Finance 9, 117-152, 1999a.
6) Hammersley, J., Monte Carlo Methods, Methuen & Co., Ltd., Lodon, U.K., 1964.
7) Hull, J.C, Options, Futures and other derivatires, Prentice Hall, 1997.