

CDOプライシングの 離散高速アプローチ(2): ツリーを用いた準解析的プライシングの マルチ・ファクター・モデルへの適用

よこやしんや
横谷進弥

要 旨

横谷 [2007] では、CDO (Collateralized Debt Obligation) プライシングに必要な条件付同時デフォルト確率についてツリー法を用いた効率的な計算手法を構築し、1ファクター・モデルへの応用例を提示した。本稿では、ツリー法のマルチ・ファクター・モデルへの応用について議論する。特に、マルチ・ファクター・モデルで必要となる(1)多次元積分の計算をいかに効率よく行うか、(2)任意の相関行列をいかに少数のファクターで表現するかという2点に焦点を当てる。多次元積分の計算では、次元数(ファクター数)が3以内であれば数値積分、4以上の場合にはモンテカルロ法が効率的となる。後者の場合、モンテカルロ法を効率化する手法である分散減少法、準モンテカルロ法の活用が重要となる。両者の収束の速さを比較検討した結果、ファクター数が大きい場合はツリー法に準モンテカルロ法を用いると、既存のCDOプライシング手法より高速に計算できることがわかった。また、相関行列のファクター化手法を最適なファクター数の選択方法とあわせて紹介し、その計算事例を示す。

キーワード：CDO、ツリー法、モンテカルロ法、準モンテカルロ法、分散減少法、相関行列のファクター化手法

本稿は、2007年3月に日本銀行金融研究所で開催された「信用リスク評価の高速化手法」をテーマとするファイナンス研究会への提出論文に加筆・修正を施したものである。同研究会の参加者からは、貴重なコメントを多数頂戴した。記して感謝したい。ただし、本稿に示されている意見は、筆者個人に属し、日本銀行の公式見解を示すものではない。また、ありうべき誤りは、すべて筆者個人に属する。

横谷進弥 日本銀行金融研究所(現 みずほ証券クレジットトレーディング部、
E-mail: shinya.yokoya@mizuho-sc.com)

1 . はじめに

近年、CDO (Collateralized Debt Obligation) の取引は増加傾向にあり、プライシングやリスク管理実務において、正確性を保ちつつ高速な計算を可能とするプライシング・モデルや計算技術が求められている。

CDOプライシング・モデルにおいては、Li [2000] がコンピュータを応用してデフォルト時点をサンプリングするモンテカルロ法を提示し、これが現在の標準的なCDOプライシング手法となっている²。このモデルは、任意の資産相関行列を仮定することができ、かつ複雑な商品においても価格を正確に計算することができるという優れた特徴を有しているが、計算負荷が大きいという欠点がある。これに対し、Laurent and Gregory [2005] は、ファクター・モデルを用いて準解析的にCDOトランシェの期待損失率を求める手法を提示した。Li [2000] のモデルでは単一相関を想定した1ファクター・モデルであったとしても計算に数分要していたが、この解析的手法は瞬時に正確に計算するため、トランチド・インデックスの標準的なプライシング・モデルとして用いられるようになった。

このような準解析的な手法では、所与の共通ファクターのもとでの条件付同時デフォルト確率を求める必要があり、これが計算負荷の大きさを決める。Laurent and Gregory [2005] は、条件付同時デフォルト確率の導出法について、離散特性関数に高速フーリエ変換を使用した手法を推奨しており、Andersen, Sidenius, and Basu [2003] はリカーシブ法という格子を用いた手法を推奨している。これに対して、横谷 [2007] はオプションのプライシングなどに用いられているツリー法を応用した手法を提示した。同手法は、厳密な条件付同時デフォルト確率を導出する離散特性関数を用いた手法と理論的に同値であり、かつ既存手法よりさらに計算を高速化する優れた手法である。また、横谷 [2007] の手法は、既存の準解析的手法の主要対象である1ファクター・モデルだけではなく、参照銘柄間の任意の資産相関に対応したマルチ・ファクター・モデルにも応用可能であり、かつ、高速なプライシングを可能とする。

本稿では、マルチ・ファクター・モデルに応用する際に問題となる2点、(1)多次元積分の計算効率化と、(2)任意の資産相関情報を少数のファクターからなるモデルに集約する手法について検討する。

マルチ・ファクター・モデルを計算するに当たって、ファクター数が3以内であれば、数値積分を用いることで期待損失率を高速に計算することができる。一方、

1 ここでの「正確性」とは、所与のプライシング・モデルが正しいと仮定したうえでの価格計算の正確性を意味している。本稿では、所与の相関構造をコンピュータで表現できると仮定し、ファクター・モデルを前提として分析を行う。なお、本稿では正規コンピュータを用いているが、他のコンピュータ・モデルも応用可能である。

2 Li [2000] の手法はSchönbucher [2003]、小宮 [2003] を参照。

ファクター数が4以上の場合、数値積分を用いると計算負荷が大きくなってしまいうため³、モンテカルロ法を使って期待値を評価することになる。モンテカルロ法の精度を上げるには多くの試行回数を要するため、可能な限り少ない試行回数で効率的に計算する技術が必要となる。これには、(a)分散減少法、(b)準乱数を用いた準モンテカルロ法という2つのアプローチが考えられる。本稿では、計算事例を用いて両手法の計算効率化の効果を検証する。

また、1ファクター・モデルの場合、大抵、単一相関を仮定するためファクター係数は相関係数と1対1対応するが、任意の相関行列をマルチ・ファクター・モデルで扱う場合は、相関行列の情報をできるだけ損なわずにファクターの最適な数やファクター係数を求める工夫が必要となる。この相関行列のファクター化に関しては、Andersen, Sidenius, and Basu [2003] がスペクトル分解を応用した効率的な手法 (Iterative Spectral decomposition Method : 以下、ISM) を提唱している。もっとも、Andersen, Sidenius, and Basu [2003] はファクター数を所与としたときの計算アルゴリズムのみを示しており、ファクター数の選択やISMの理論的枠組みについての詳しい説明は行っていない。そこで、最適ファクター数の導出法 (以下、ISM+ α) や、そのアルゴリズムを提示し、ISMとISM+ α の理論的枠組みについて解説を行う。

以下、2節ではCDOトランシェの準解析的計算手法の概要を説明し、3節では横谷 [2007] で提示されたツリー法によるCDOの条件付同時デフォルト確率の求め方を示す。4節では、ツリー法のマルチ・ファクター・モデルへの応用に必要なモンテカルロ法および準モンテカルロ法について説明する。5節では、相関行列のファクター化手法としてISMおよびISM+ α のアルゴリズムを提示し、補論2でその理論的な枠組みを解説する。6節ではマルチ・ファクター・モデルの計算事例を示し、モンテカルロ法における収束の速さを比較検討する。最後に7節で分析内容をまとめる。

2 . 準解析的CDOプライシング

本節では、ファクター・モデルと条件付デフォルト確率の設定を行ったうえで、準解析的なCDOトランシェのプライシング手法を説明する。

(1) ファクター・モデルと条件付デフォルト確率の設定

参照銘柄数を m として、ファクター・モデルを以下のように定義する。

$$\mathbf{V} = \mathbf{A}\mathbf{Y} + \mathbf{B}\boldsymbol{\epsilon}. \quad (1)$$

.....
3 高次元の数値積分は分割数の次数乗の計算負荷がかかる。

ここで、 $\mathbf{V} = (v_1, \dots, v_m)^T$ 、 $\mathbf{B} = (b_1, \dots, b_m)^T$ 、 ϵ はおのおの m 次元ベクトル、 \mathbf{A} は各要素を a_{ij} とする $m \times z$ 行列であり、 z はファクターの数とする。また、 $\mathbf{Y} = (y_1, \dots, y_z)^T$ と ϵ は互いに独立な正規確率変数とし、 $b_i (i = 1, \dots, m)$ は

$$b_i = \sqrt{1 - \sum_{j=1}^z a_{ij}^2} \quad (2)$$

で与えられる。

v_i を各参照企業の資産価値を示す確率変数とすると、資産価値 v_i は z 個の共通ファクター \mathbf{Y} と、1個の独自要因（固有ファクター） ϵ_i によって決まる。

ここで、各参照企業の時点 t でのデフォルト確率を \hat{p}_i とする⁴。(1)式で v_i は標準正規分布に従うことから、

$$\hat{p}_i = P_r [v_i < \Phi^{-1}(\hat{p}_i)], \quad (3)$$

が成り立つ。ここで、 $\Phi(\cdot)$ は z 次元標準正規分布関数とする。

(1)式の共通ファクター \mathbf{Y} が与えられたもとの条件付デフォルト確率を p_i とすると、

$$p_i = \Phi \left(\frac{\Phi^{-1}(\hat{p}_i) - \sum_{j=1}^z a_{ij} y_j}{b_i} \right), \quad (4)$$

となり、任意の $i, j (i \neq j)$ に対して p_i, p_j は独立な事象に対する確率となる。参照銘柄全体（以下、プール）の条件付同時デフォルト確率は、この条件付デフォルト確率を用いて解くことができる。

(2) 共通ファクター係数

(1)式の共通ファクター係数 \mathbf{A} は、資産相関を表す何らかの情報に基づいて、外生的に与えなくてはならない。1ファクター・モデル($z = 1$)では、各参照銘柄間の資産相関が単一相関 $\rho (0 < \rho < 1)$ で表され、 \mathbf{A} はベクトル $a_{i1} = \sqrt{\rho} (i = 1, \dots, m)$ となる。これに対し、任意の相関行列を想定する場合、5節で説明するスペクトル分解を応用した手法を用いることで、少ない次元数（ファクター数）で共通ファクター係数 \mathbf{A} を表すことができる。

(3) デフォルト時損失率

トランシェごとのデフォルト時損失額とデフォルト時損失率を表現するための工

4 以下の議論は任意の t について成立するため、 t を省いて表記する。

夫を紹介する。まず、参照銘柄のデフォルト時損失額（額面×(1-回収率)）について考える。CDOをプライシングする際、(4)式によって求まる条件付デフォルト確率を用いて、プールの条件付同時デフォルト確率を計算する。そのときデフォルト時損失額が同じであれば、同時デフォルト確率は「 i 銘柄デフォルトする確率（ i は銘柄数の合計を最大とする自然数）」と定義できる。一方、デフォルト時損失額が銘柄間で異なれば、銘柄を単位として考えることは難しい⁵。そこで、本稿ではデフォルトを数える単位を細かくし、デフォルト時損失額の最大公約数を単位とする。このように設定することにより、各銘柄のデフォルトを「最大公約数を基準として k 単位デフォルトする（ k は自然数）」事象として捉えることができ、同時デフォルト確率もこの単位で表現できる。ここで、デフォルト時損失額を自然数で表せるとして⁶、それぞれの損失額を $\{a_1, a_2, \dots, a_m\}$ とする。このとき、 $\{a_1, a_2, \dots, a_m\}$ の最大公約数を H として、

$$b_i = \frac{a_i}{H}, i = 1, 2, \dots, m, \quad (5)$$

を要素とするベクトル $\mathbf{B} = \{b_1, b_2, \dots, b_m\}^T$ を定義する。 b_i は各銘柄のデフォルト時損失額が H を基準として何単位分かを示す自然数である。以下では、この \mathbf{B} を用いて同時デフォルト確率を考えていく。総単位数 n は $n = \sum_{i=1}^m b_i$ で与えられる。

また、トランシェの損失率を表すベクトルをサイズ $n+1$ のベクトル \mathbf{L} とし、その要素 l_k ($k = 0, \dots, n$) はプールが k 単位毀損されたときの、トランシェの損失率を表すものとする。

(4) 多次元積分の手法選択

CDOトランシェのプライシングにおいては、満期までの任意時点の期待損失率を求めることが中心的な課題となる。プロテクション・レグ（デフォルト発生時の支払額）の価値は、額面に期待損失率を乗じて得られ、プレミアム・レグ（期中プレミアムの支払額）の価値は、期待損失率の関数として考えることができる⁷。したがって、ここでは基礎となる期待損失率の求め方を説明する。

まず、時点 t までに n 単位中 k 単位だけプールが毀損する条件付同時デフォルト確率を

$$cp(t, k, n | \mathbf{Y}), \quad (6)$$

5 例えば、全ての銘柄間でデフォルト時損失額が違うのであれば、同時デフォルトの場合の数が2の銘柄数乗となる。

6 クレジット商品は各参照銘柄を億円単位で取引することが普通であり、回収率もパーセント表示で小数点以下まで細かく規定することは稀であるので、ほとんどの場合この仮定は成り立つ。

7 Laurent and Gregory [2005] が詳しい。

時点 t までに n 単位中 k 単位以上プールの毀損する条件付同時デフォルト確率を

$$CP(t, k, n | \mathbf{Y}), \quad (7)$$

とする。また、アタッチメント（トランシェの下限）を o 、デタッチメント（トランシェの上限）を O とし、プール全体の損失率が $[o, O]$ に収まる単位数を $[d, u]$ (d, u は自然数)⁸ とすると、時点 t のトランシェの条件付瞬間期待損失率 $g(t | \mathbf{Y})$ は、

$$g(t | \mathbf{Y}) = \sum_{j=d}^{u-1} l_j \frac{\partial cp(t, j, n | \mathbf{Y})}{\partial t} + l_u \frac{\partial CP(t, u, n | \mathbf{Y})}{\partial t}, \quad (8)$$

のように求まり、 t 時点の割引率を $D(t)$ （本稿では確定的関数とする）とすると、 \mathbf{Y} を条件としたときの 0 時点の条件付期待損失率 $EL^0(\mathbf{Y})$ は以下のように求まる。

$$EL^0(\mathbf{Y}) = \int_0^T g(t | \mathbf{Y}) D(t) dt. \quad (9)$$

最後に、 \mathbf{Y} で積分することにより、0 時点でのトランシェの期待損失率 EL^0 を以下のように求めることができる。

$$EL^0 = \int EL^0(\mathbf{Y}) \phi(\mathbf{Y}) d\mathbf{Y}. \quad (10)$$

ここで、 $\phi(\cdot)$ は多次元標準正規密度関数である⁹。

(9) 式の積分は時点 t についての 1 次元積分なので数値積分を用いて計算することができる¹⁰。

同様に、(10) 式の積分はファクター数が 3 以内 ($z \leq 3$) の場合は、数値積分の中でも Gauss = エルミート (Gauss-Hermite) 法 (Gauss 求積法の 1 つ) を用いることにより比較的速く計算することができる¹¹。もっとも、ファクター数 z が 4 以上になると、Gauss = エルミート法の分割数を d とすると計算のオーダーが $O(d^z)$ となり級数的に大きくなるため¹²、モンテカルル口法または準モンテカルル口法を用いる必要があ

8 総額面を TN とすると、 $d = \min(k | o < \frac{kH}{TN}, k \in \{1, \dots, n\})$, $u = \min(k | O < \frac{kH}{TN}, k \in \{1, \dots, n\})$ となる。

9 横谷 [2007] では積分の順序が違うが結果は同じである。

10 この種の数値計算には Gauss 求積法が望ましい。Gauss 求積法とその実装法については Press *et al.* [1994] が詳しい。本稿では Gauss 求積法の Gauss = ルジャンドル (Gauss-Legendre) 法を用いた。

11 Gauss = エルミート法の分析の結果、ファクター数が 3 の場合、1 次元につき 10 分割 (計 1,000 分割) で十分収束することがわかった。これに対し、後の節で説明する準モンテカルル口法は試行回数が 1,000 回では十分収束するとは言いえない。ちなみに、分割数と試行回数は計算過程上で被積分関数を呼ぶ回数という意味で同じである。ただし、Gauss = エルミート法ではファクター数が 4 以上でも、安定的な値を求めるには 1 次元につき 10 分割ほど必要になるが、それであると合計で 10,000 分割以上となってしまう。この場合は、6 節で説明するように準モンテカルル口法で計算する方が少ない試行回数で計算することができる。上述の結果は、CDO のさまざまな商品設定においても成立しており、後述する計算手法の性質から一般性をもって成立することが推測される。

12 この問題を解決する手法の 1 つとして、スパース・グリッド (Sparse Grids) という手法がある。スパース・グリッドを用いた多次元積分については Gerstner and Griebel [1998] を参照。Gerstner and Griebel

る。モンテカルロ法・準モンテカルロ法については4節で詳しく説明する。

3. ツリーを用いた条件付同時デフォルト確率の求め方

本節では、横谷 [2007] で提示されたツリー法によるCDOトランシェの条件付瞬間期待損失率 $g(t|\mathbf{Y})$ の計算手法について説明する。

$x_k (k = 0, \dots, n)$ を条件付同時デフォルト確率 ($= cp(t, k, n|\mathbf{Y})$)、デタッチメントより小さい最大の (前述した最大公約数 H を単位とした) 単位数を $detach-$ ($= u-1$) とすると、条件付瞬間期待損失率は、

$$g(t|\mathbf{Y}) = \sum_{k=1}^{detach-} (l_k - 1) \frac{\partial x_k}{\partial t}, \quad (11)$$

のように表すことができる¹³。条件付同時デフォルト確率 $x_k (k = 0, \dots, detach-)$ は、以下のように、ツリーによるフォワード・アルゴリズムで計算することができる (導出過程については横谷 [2007] を参照)。

1. $c_0(0)=1$ とする。
2. $i = 1, \dots, m$ の順序で、サイズ $(1 + \sum_{j=1}^i b_j)$ のベクトル C_i を用意する。ただし、 $1 + \sum_{j=1}^i b_j > detach-$ のときは、 C_i のサイズを $(detach-)+1$ とする。
 C_i の各要素は、 $k = 0, \dots, (\text{Size of } C_i - 1)$ について、

$$c_i(k) = p_i c_{i-1}(k - b_i) + (1 - p_i) c_{i-1}(k),$$

とする。ただし、 $k - b_i < 0$ の際は右辺第1項を0とする。逆に、 $c_{i-1}(k)$ がない場合は右辺第2項を0とする。

3. 最後の C_m の要素が $c_m(k) = x_k$ (条件付同時デフォルト確率) となる。

上記のアルゴリズムによって求まる条件付同時デフォルト確率は、プールの損失分布の離散特性関数を離散逆フーリエ変換して求めた値と同値であり、厳密な条件付同時デフォルト確率である¹⁴。また、アルゴリズムとして Andersen, Sidenius, and Basu [2003] のリカーシブ法よりも効率的な手法である¹⁵。

[1998] に、被積分関数が滑らかでないときにスパース・グリッドを用いるよりも準モンテカルロ法を用いた方が効率的であるという指摘がある。ちなみに、本稿のモデルでも同手法を試みたが、本稿で扱う被積分関数が滑らかでないためか、少ない分割数では収束しないことが多かった。

13 (11)式の導出法は横谷 [2007] を参照。

14 横谷 [2007] を参照。

15 横谷 [2007] の数値事例において、50銘柄のプールではツリー法の方がリカーシブ法より約2倍速く計算できることが示されている。

4. マルチ・ファクター・モデルのモンテカルル口法

本節では、マルチ・ファクター・モデルにおける期待損失率 EL^0 を与える(10)式を計算する手法について議論する。まず、4節(1)で基本的なモンテカルル口法のアルゴリズムについて説明し、計算の効率化のための手法として4節(2)で分散減少法を、4節(3)で準モンテカルル口法を説明する。なお、2節(4)で指摘したように、1ファクター・モデルとマルチ・ファクター・モデルの差は(10)式の計算手法の違いだけである。したがって、横谷 [2007] の4節(5)で説明した複数のトランシェの計算や感応度計算の効率的な手法(同5節)は、マルチ・ファクター・モデルにおいても応用可能である。

(1) 基本的なモンテカルル口法

基本的なモンテカルル口法は、以下のアルゴリズムで表せる。

1. メルセイヌ・ツイスターなどの擬似乱数により z 個の一樣乱数 $u_j (j=1, \dots, z)$ を発生させ、 $y_j = \Phi^{-1}(u_j) (j=1, \dots, z)$ により標準正規変数ベクトル $\mathbf{Y}_l (l=1)$ を生成する。
2. 3節のアルゴリズムにより $g(t|\mathbf{Y}_l)$ を計算する。
3. (9)式により $EL^0(\mathbf{Y}_l)$ を計算する。
4. 1.~3. を M 回 ($l=1, \dots, M$) 行った後、その和を試行回数 (M) で割ることで、(10)式の近似値 \hat{EL}_M^0 とする。

この近似値 \hat{EL}_M^0 は、

$$\hat{EL}_M^0 = \frac{1}{M} \sum_{l=1}^M EL^0(\mathbf{Y}_l), \quad (12)$$

で与えられる。 $EL^0(\mathbf{Y}_l)$ については、数値積分によりほぼ正確な数値が得られるため、 \hat{EL}_M^0 は試行回数 M が十分大きければ、大数の法則により EL^0 に収束する。

ただ、上記のモンテカルル口法は Li [2000] の手法¹⁶と比べ、1回の試行当たりの計算時間が長くなるため、CDO トランシェの計算の高速化を図るためには何らかの工夫が必要となる。その手法として分散減少法と準モンテカルル口法の概要を本稿のモデルに沿って示す¹⁷。

16 コピュラ関数を用いて、各参照組織のデフォルト時刻をサンプリングし、1試行ごとにトランシェの損失率を計算する手法。同手法の場合、1回の試行当たりの計算時間が短くて済む。

17 各手法の理論的な詳細については、湯前・鈴木 [2000]、Jäckel [2002]、Glasserman [2004] を参照。

(2) 分散減少法

基本的なモンテカルロ法は、単純に $EL^0(\mathbf{Y}_i)$ をサンプリングして平均をとる手法であるのに対し、分散減少法は(a)期待値のとり方を変更したり、(b)同じ期待値を持つ別の関数に置き換えてサンプリングすることにより、誤差分散を減少させて、より少ない試行回数で期待値に収束させる手法である。本稿の枠組みでは次のように表すことができる。(12)式の誤差分散は、 \mathbf{Y}_i が互いに独立であるとすると、

$$\begin{aligned} V(\hat{EL}_M^0) &= \frac{1}{M^2} \sum_{i=1}^M V[EL^0(\mathbf{Y}_i)] \\ &= \frac{1}{M} V[EL^0(\mathbf{Y})] \\ &= \frac{K_0}{M}, \end{aligned} \quad (13)$$

となる。ここで、 $K_0 = V[EL^0(\mathbf{Y})]$ とした。分散減少法を言い換えると、(13)式の分散 K_0 をより小さい分散 $K_i (K_i < K_0)$ とすることにより、 \hat{EL}_M^0 の分散減少を施す手法であるといえる。以下、代表的な分散減少法を説明する。

(a) 期待値のとり方を変更する手法

1) 条件付モンテカルロ法

「条件付モンテカルロ法」は(i)条件付期待値の期待値が無条件期待値と一致すること、(ii)無条件誤差分散より条件付誤差分散の方が小さいこと¹⁸を利用し、条件付期待値をサンプリングして平均をとる近似法である。実は、(12)式は $EL^0(\mathbf{Y})$ が条件付期待値であることから条件付モンテカルロ法に属しており、もともと誤差分散が小さくなる枠組みとなっている。これに対し、Li [2000]の手法は無条件モンテカルロ法に属する。

2) 層別化法

条件付モンテカルロ法を応用した手法の中で「層別化法 (Stratified Sampling)」という手法がある。この手法は1次元積分の場合、一様乱数の範囲[0, 1]をあらかじめ m 個の小区間

$$[c_1, c_2], [c_2, c_3], \dots, [c_m, c_{m+1}] (c_1 = 0, c_{m+1} = 1),$$

に区切る。ここで、各区間の幅は任意とする。次に、それぞれの小区間で個別にモンテカルロ法を行い、最後におのおのの区間幅をウエイトとする加重和をとる手法

18 この証明は補論1を参照。

である¹⁹。同手法は理論的には多次元にも応用ができるが、次元数を z とすると分割数が m^z と級数的に大きくなり、計算負荷が増すことになる。

3) ラテン・ハイパーキューブ法

多次元積分を計算する際、層別化法の計算量が級数的に増大してしまう問題に対応し、McKay, Conover, and Beckman [1979] は「ラテン・ハイパーキューブ法 (Latin Hypercube Sampling : 以下、LHC法)」という手法を提唱した。LHC法は各次元ごと m 個の小区間に均等に区切って m^z 個の層を作り、ランダム順列により m 個の層を選び、その各層内からランダムに多次元一様乱数をサンプリングする手法である。この手法は m が十分大きければ誤差分散が減少することがStein [1987] によって示された。

4) 重点サンプリング法

「重点サンプリング法 (Importance Sampling)」は、被積分関数の密度が高い箇所において多くサンプリングするように確率密度を変更する手法である。ただし、本稿の枠組みの場合、被積分関数に依存する重点サンプリング法は次の点から計算量削減効果が期待できない²⁰。

1. 分析の結果、被積分関数 ((10)式の $EL^0(\mathbf{Y})\phi(\mathbf{Y})$) が単峰型の場合、確率分布の平均を被積分関数の最大値に移動させる重点サンプリング法²¹の分散減少効果が大きいことが確かめられたが、被積分関数に局所最大値が複数ある場合、分散減少効果が望めない場合があることが確認された。本稿の設定では、共通ファクター係数 A の絶対値が大きい場合 (想定する相関が大きい場合) 局所最大値が複数存在する傾向があり、重点サンプリング法が分散減少につながらないことがある。
2. また、重点サンプリング法はGlasserman and Li [2005] に示されているように、被積分関数を最大化させる確率変数 (本稿でいえば \mathbf{Y}) を求めるという最適化問題を解く必要があるが、局所最大値が複数ある場合、グローバル最適解を短時間で求めることは容易ではない。
3. さらに、本稿の枠組みでは、各時点でツリーを構築しなくてはならないため、被積分関数を計算する負荷が大きく、必然的に最適解を計算する負荷が大きくなってしまう。
4. 横谷 [2007] で示したように、同一プールに複数のトランシェがある場合の計算や、各銘柄のデフォルト確率の感応度の計算においては、一度ツリーで

19 層別化法はGlasserman [2004] が詳しい。

20 一般的に、重点サンプリング法は非常に有効な分散減少法として知られている。特に、被積分関数が単峰型の場合には非常に強力な計算手法となる。重点サンプリングやその応用例、層別化法との組合せ手法等の詳細はGlasserman [2004] を参照。

21 Glasserman and Li [2005] の5節を参照。

条件付同時デフォルト確率を求めれば、それを繰り返し使用し、効率的に計算することができる。これは上記のLHC法や後述する負の相関法を使った場合においても成り立つが、重点サンプリング法では成り立たない。これは、重点サンプリング法は被積分関数に依存した手法であるため、複数のトランシェや感応度の計算に同じ条件付デフォルト確率を繰り返し用いることができないことに起因する。

(b) 同じ期待値を持つ別の関数に置き換えてサンプリングする手法

5) 負の相関法

「負の相関法 (Antithetic Sampling)」は上記の基本的なモンテカルロ法のアルゴリズム1.において、 Y_l と同時に $-Y_l$ を用意し、

$$\tilde{EL}_M^0 = \frac{1}{M/2} \sum_{l=1}^{M/2} \{EL^0(Y_l) + EL^0(-Y_l)\}, \quad (14)$$

で求まる \tilde{EL}_M^0 を(10)式の近似値とする手法である。(14)式の誤差分散は、

$$\begin{aligned} V(\tilde{EL}_M^0) &= \frac{4}{M^2} \sum_{l=1}^{M/2} V\{EL^0(Y_l) + EL^0(-Y_l)\} \\ &= \frac{1}{M} [V\{EL^0(\mathbf{Y})\} + \text{Cov}\{EL^0(\mathbf{Y}), EL^0(-\mathbf{Y})\}], \end{aligned} \quad (15)$$

と表すことができ、 $\text{Cov}\{EL^0(\mathbf{Y}), EL^0(-\mathbf{Y})\} < 0$ が成り立てば、 $\text{Var}(\tilde{EL}_M^0) < \text{Var}(\hat{EL}_M^0)$ となり、分散が減少する。

本稿の枠組みでは、(8)式の $g(t|\mathbf{Y})$ が条件付確率 $p_i (i = 1, \dots, m)$ の加法と乗法の組み合わせであり、 p_i に対して比例関係にあるということ、および(4)式の関係から、

$$\text{Cov}(g(t|\mathbf{Y}), g(t|-\mathbf{Y})) < 0, \quad (16)$$

が任意の時点 t について成り立つ。このため、

$$\text{Cov}(EL^0(\mathbf{Y}), EL^0(-\mathbf{Y})) < 0, \quad (17)$$

が成立し、本稿の枠組みにおいて負の相関法が分散減少効果を持つことがわかる。

(3) 準モンテカルロ法

一様乱数に対して準乱数 (Halton列、Faure列など) を用いる手法は「準モンテカルロ法」と呼ばれている。準乱数はlow-discrepancy列と呼ばれ、Discrepancyという一様性を示す指標がある一定基準を下回る数列のことである。したがって、準乱数は擬似乱数と違い、乱数としてのランダムさを満たす保証はない。擬似乱数を用いるモンテカルロ法の収束性は大数の法則により成り立つ議論であり、収束のオー

ダーは(13)式の平方根をとることで $O(1/\sqrt{M})$ であることがわかる。それに対し、準モンテカルロ法は準乱数の収束のオーダーは $O((\log M)^z / M)$ となる²²。したがって、準モンテカルロ法の方が基本的なモンテカルロ法で計算するより少ない試行回数で収束する可能性が高い。また、準モンテカルロ法は、擬似乱数を用いるモンテカルロ法のアルゴリズムの中で一様乱数を準乱数に取り替えるだけであり、手法の変更に伴う実装上の負荷が全くないこともこの手法の利点といえる。

また、Jäckel [2002] で指摘されているように、古典的な準乱数であるHalton列やFaure列、Sobol列等では高次元の計算が非効率となるが、近年、多次元においても効率的な計算が可能となる準乱数が開発されており、それらのコードも手軽に入手することができる。例えば、「QuantLib²³」というC++ライブラリーには古典的な準乱数であるHalton列やFaure列、Sobol列のほか、改良型Halton列や改良型Sobol列がある²⁴。

また、準乱数は計算対象となる関数形によって相性があり、一般的にどの準乱数が一番よいということはいえず、使用者はさまざまな準乱数を試す必要がある。本稿では、6節で改良型Halton列、Faure列、改良型Sobol列²⁵の比較を行う。

5 . 相関行列のファクター化手法

本節では(1)式のファクター・モデルについて、相関行列の情報をできるだけ損なわずに、かつ少ないファクター数で相関構造を表せるような共通ファクター係数Aの導出手法をAndersen, Sidenius, and Basu [2003] に沿って説明する。なお、Andersen, Sidenius, and Basu [2003] はファクター数を所与としたときの計算アルゴリズムの説明にとどまっていたが、本稿では最適なファクター数の導出法についても説明し、補論2では、その理論的な枠組みを示す。

まず、全参照銘柄*m*社の資産相関行列をΣとする。(1)式のAを用いて、 $m \times m$ 行列Cを

$$C = I - \text{diag}AA^T, \quad (18)$$

22 準モンテカルロ法の詳細はJäckel [2002] を参照。

23 QuantLib (<http://quantlib.org/>) は金融工学や金融実務で標準的に使われる関数・クラスが網羅された無料ライブラリーであり、数学関数も充実している。作成は世界のユーザが参加するメーリングリスト内での情報交換をもとに行われており、関数のテスト・コードもあるので信頼性も高いといえる。本稿の計算事例ではQuantLib Ver.0.4.0を用いた。

24 湯前・鈴木 [2000] 154頁において指摘されているように、一部の準乱数では特許が成立しているとのことなので、使用者はこの特許に配慮しなくてはならない。

25 Halton列、Faure列、Sobol列は、QuantLibのHaltonRsgクラス、FaureRsgクラス、SobolRsgクラスを用いた。HaltonRsgの設定は (dimensionality=5、seed=1000、randomStart=true、randomShift=true) FaureRsgの設定は (dimensionality=5)、SobolRsgクラスの設定は (seed=0、directionIntegers=Jaeckel) とした。

で与える。ただし、 \mathbf{I} は単位行列とする。

このとき、次の関係を満たす $m \times z$ 行列 $\tilde{\mathbf{A}}$ を考える。

$$\tilde{\mathbf{A}} = \arg \min_{\mathbf{A}} \{ \text{tr} (\Sigma - \mathbf{A}\mathbf{A}^T - \mathbf{C})(\Sigma - \mathbf{A}\mathbf{A}^T - \mathbf{C})^T \}. \quad (19)$$

(19)式²⁶から求まる $\tilde{\mathbf{A}}$ は、相関行列 Σ とファクター係数 \mathbf{A} の情報量 $(\mathbf{A}\mathbf{A}^T + \mathbf{C})$ の2乗誤差を最小にするファクター係数と捉えることができる。もっとも、 \mathbf{A} の要素を全て動かして(19)式を最適化することは困難であるので、以下のようにスペクトル分解を応用して $\tilde{\mathbf{A}}$ を求め²⁷。本稿では、以下の手法をISM (Iterative Spectral decomposition Method) と呼ぶ。

ここで、 z をファクター数、 $\tilde{\Gamma}$ を $\Sigma - \mathbf{C}$ の固有値を大きい順に並べたときの上位 z 個に対応する固有ベクトルの束 ($m \times z$ 行列) とし、 $\tilde{\Lambda}$ をその固有値の上位 z 個を対角要素とした行列とする ($z \times z$ 対角行列)。このとき、 $m \times z$ 行列 $\hat{\mathbf{A}}$ を

$$\hat{\mathbf{A}} = \tilde{\Gamma} \sqrt{\tilde{\Lambda}}, \quad (20)$$

とし、 $m \times m$ 行列 $\hat{\mathbf{C}}$ を

$$\hat{\mathbf{C}} = \mathbf{I} - \text{diag} \hat{\mathbf{A}} \hat{\mathbf{A}}^T, \quad (21)$$

とする。ここで、 Σ は既知であるが、 \mathbf{A} あるいは(18)式で与えられる \mathbf{C} は未知であるため、まず $\mathbf{C} = 0$ として $\tilde{\Gamma}$ や $\tilde{\Lambda}$ を求め、これらにより $\hat{\mathbf{A}}$ を計算する。その結果、(21)式で $\hat{\mathbf{C}}$ が得られるため、 \mathbf{C} を更新する。この繰り返しにより $\hat{\mathbf{A}}$ が収束するまでのアルゴリズムは以下のように表せる。最終的に得られた $\hat{\mathbf{A}}$ は(19)式の $\tilde{\mathbf{A}}$ となる。証明は補論2を参照。

1. Σ に対し固有値 $\tilde{\Lambda}_0$ 、固有ベクトル $\tilde{\Gamma}_0$ を求め、 $\hat{\mathbf{A}}_0 = \tilde{\Gamma}_0 \sqrt{\tilde{\Lambda}_0}$ とする。
2. $\hat{\mathbf{C}}_0 = \mathbf{I} - \text{diag} \hat{\mathbf{A}}_0 \hat{\mathbf{A}}_0^T$ を求める。 $s = 1$ とする。
3. $\Sigma - \hat{\mathbf{C}}_{s-1}$ に対し固有値 $\tilde{\Lambda}_s$ 、固有ベクトル $\tilde{\Gamma}_s$ を求め、 $\hat{\mathbf{A}}_s = \tilde{\Gamma}_s \sqrt{\tilde{\Lambda}_s}$ とする。
4. $\hat{\mathbf{C}}_s = \mathbf{I} - \text{diag} \hat{\mathbf{A}}_s \hat{\mathbf{A}}_s^T$ を求める。
5. $\text{tr} (\hat{\mathbf{C}}_s - \hat{\mathbf{C}}_{s-1})(\hat{\mathbf{C}}_s - \hat{\mathbf{C}}_{s-1})^T$ が一定水準(G)²⁸以下になったらストップし、 $\mathbf{A} = \hat{\mathbf{A}}_s$ とする。
6. 5.の条件が満たされない場合、 $s = s + 1$ として、3.から繰り返す。

この手法がAndersen, Sidenius, and Basu [2003]で示された共通ファクター \mathbf{A} の計算手法であり、所与のファクター数 z のもとで、最適なファクター係数を選ぶことができる。このアルゴリズムは大抵2、3回で収束するような非常に効率的な計算法

26 X を $m \times m$ 行列とすると、 $\text{tr}(X) = \sum_{i=1}^m X_{ii}$ と定義する。

27 スペクトル分解については補論2で説明する。

28 本稿の計算事例では $G = 1.0e-8$ とした。

である²⁹。本稿ではここまでの計算法を「ISM」と呼ぶ。

さらに、(19)式を用いて、最適なファクター数 z を決めることも可能である。 $\hat{A}(z)$ を、 z を所与としたときのISMで求まる共通ファクター係数とし、 $\hat{C}(z) = \mathbf{I} - \text{diag}\hat{A}(z)\hat{A}(z)^T$ となる対角行列 $\hat{C}(z)$ を設ける。このとき、 $z=1, 2, \dots$ の順で

$$\text{Er}(z) = \text{tr} \{ \Sigma - \hat{A}(z)\hat{A}(z)^T - \hat{C}(z) \} \{ \Sigma - \hat{A}(z)\hat{A}(z)^T - \hat{C}(z) \}^T, \quad (22)$$

を計算し、 $\text{Er}(z)$ が上記アルゴリズム5.の G 以下になると、その $\hat{z} = z$ が最適値となる。この計算手法を「ISM+ α 」と呼ぶ。

最適ファクター数 \hat{z} は Σ のランクに近いものを表しており³⁰、本稿の計算事例(表1)のようにカテゴリー別の相関行列を設定すれば、そのカテゴリー数が z の最適値になる(この点は後述する)。ただ、カテゴリー数を大きくした場合や、ヒストリカルな市場データから資産相関を求めた場合、 \hat{z} が大きくなる可能性がある。そして、 \hat{z} が大きいファクター・モデルに準モンテカルロ法を用いる場合、古典的な準乱数であるHalton列、Sobol列、Faure列を使用すると収束が遅くなることがあるが、改良されたHalton列やSobol列を用いると効率的に計算することができる³¹。

なお、最適ファクター数 \hat{z} が参照銘柄数 m と同じになってしまう場合は、注意が必要である。 $\hat{z} = m$ のとき、補論2の(A-2)式より、 $\Sigma = \hat{A}(\hat{z})\hat{A}(\hat{z})^T$ 、 $\hat{C}(\hat{z}) = 0$ になってしまうので、ファクター・モデルとして計算できない。そのようなときは、ISM+ α の繰り返し計算の停止判定に上記の G よりも大きな値 \hat{G} を適用すればよい。このとき、 \hat{G} を大きくし過ぎるとランクを落とすこと(最適な次元数 \hat{z} 以下で計算すること)によってトランシェ・スプレッドの誤差が拡大してしまうので、想定する相関水準と商品群から、許容誤差を判断して \hat{G} を定める必要がある。

また、最適ファクター数が3より大きい($\hat{z} > 3$)場合でも、ISMによりファクター数を3としたときに誤差が許容範囲に収まるようであれば、2節(4)で指摘したように、ファクター数を3とし、(10)式をガウス=エルミート法を用いて計算した方が速く計算することができる。

6. 計算事例

本節では、4節で説明したモンテカルロ法の計算事例を示す。最初にCDOトランシェや資産相関の設定を行い、次に分散減少法と準モンテカルロ法の効果を検証する。また、試行回数だけではなく計算時間を基準とした比較検証も行う。

29 50銘柄の相関行列から5ファクター係数(表4)を導出する計算時間は0.016秒であった。

30 正確には、 $\Sigma - \hat{C}(\hat{z})$ のランクを表している。

31 ファクター数が増えても一様乱数の生成による計算負荷が増えるだけなので、全体の計算負荷はあまり変わらない。

(1) 設定

計算対象とするCDOは50銘柄のCDS (Credit Default Swap) を原資産とする満期5年のシンセティックCDOであり、全銘柄の額面が均等なプールである。クーポンの支払いは年4回とし、デフォルト時に経過利子分を受け渡す契約とする。割引率は固定ショートレート関数 $\exp(-rt)$ を仮定し、 $r=1.34\%$ とする。各参照銘柄のCDSも満期は5年、クーポン支払いは年4回とし、デフォルト時に経過利子分を受け渡す契約とする。全銘柄のデフォルト時回収率は40%とする。満期5年のCDSスプレッドは1銘柄目が2bp (1bp = 0.01%)、2銘柄目が4bp、と2bpずつ上昇し、50銘柄目が100bpとする。全ての銘柄のデフォルト確率に対して固定ハザードレート関数 $(1 - \exp(-h_i t))$ を仮定し、各 h_i はCDSスプレッドから逆算した。

また、相関行列を表1のように3種類想定した。例えば「低相関」の場合、最初の1銘柄目～10銘柄目までは互いの資産相関が2%、11銘柄目～20銘柄目が3%とし、それらの区間外の銘柄間相関(例えば、1銘柄目と11銘柄目)は1%とした。「中相関」「高相関」も同様である。この想定は、格付会社等で用いられている業種・地域相関の仮定を簡素化したものである。

上記の設定のもと、Li [2000] の手法で試行回数を500万回と十分多くとったときのトランシェ・スプレッドを表2に示した。以下の分析では、このスプレッドを真の値と仮定する³²。なお、(9)式に対する数値積分は、ガウス＝ルジャンドル(Gauss-Legendre)法(40分割³³)を適用した³⁴。

表3は(22)式の $\text{Er}(z)$ を求めた結果であり、5節のISM + α を用いると³⁵、どの相関水準でも最適ファクター数は5となった。5節で指摘したように、表1のようにカテゴリー別の相関を仮定すると、カテゴリー数が最適ファクター数と同じになる。ファクター数を5とし、ISMを用いて導いた5次元のファクター係数を表4に示した。

表1 相関行列

| 銘柄 | 低相関 | 中相関 | 高相関 |
|-------|-----|-----|-----|
| 1～10 | 2% | 20% | 60% |
| 11～20 | 3% | 25% | 40% |
| 21～30 | 4% | 30% | 30% |
| 31～40 | 3% | 25% | 40% |
| 41～50 | 2% | 30% | 60% |
| 区間外 | 1% | 5% | 10% |

32 Li [2000] の手法で計算する際、正規乱数の生成に必要な一様乱数にはメルセイヌ・ツイスターを用い、相関のある正規乱数の生成にはコレツキー分解を使った。上記の設定のもとでは試行回数を500万回とすれば極めて安定的な値に収束したので、その値を真の値と仮定して比較を行うことにした。

33 各クーポン期間内(今回は0.25年を想定)で2分割の積分を行った。

34 プログラム言語はC++、インテル(R) C++コンパイラーでコンパイルした。使用したPCのスペックは、CPUがIntel(R) Celeron(R) 2.66GHzで、メモリは512MBである。

35 ISMのアルゴリズム5.の G を $1.0e-8$ とした。

表2 スプレッド水準

| トランシェ | スプレッド (bp) | | |
|--------|------------|----------|----------|
| | 低相関 | 中相関 | 高相関 |
| 0～3% | 2,075.87 | 1,706.33 | 1,405.76 |
| 3～6% | 336.46 | 415.26 | 438.46 |
| 6～9% | 25.92 | 99.63 | 162.94 |
| 9～12% | 1.42 | 25.17 | 66.66 |
| 12～22% | 0.01 | 1.90 | 10.38 |

表3 相関行列のファクター化誤差

| ファクター数 | 低相関 | 中相関 | 高相関 |
|--------|----------|----------|----------|
| 1 | 0.1307 | 16.4186 | 49.9036 |
| 2 | 0.0647 | 10.7453 | 26.9075 |
| 3 | 0.0276 | 6.1389 | 13.0415 |
| 4 | 0.0095 | 2.3860 | 4.6083 |
| 5 | 4.25E-10 | 2.47E-10 | 1.37E-10 |

表4 共通ファクター係数

| 銘柄 | ファクター1 | ファクター2 | ファクター3 | ファクター4 | ファクター5 |
|-------|--------|---------|---------|---------|---------|
| 低相関 | | | | | |
| 1～10 | 9.82% | -2.53% | 0.00% | -6.87% | -7.07% |
| 11～20 | 11.81% | -6.25% | 10.00% | 4.63% | 0.00% |
| 21～30 | 14.83% | 13.31% | 0.00% | 1.73% | 0.00% |
| 31～40 | 11.81% | -6.25% | -10.00% | 4.63% | 0.00% |
| 41～50 | 9.82% | -2.53% | 0.00% | -6.87% | 7.07% |
| 中相関 | | | | | |
| 1～10 | 23.95% | 0.00% | -7.87% | 0.00% | -36.94% |
| 11～20 | 28.46% | 0.00% | -24.26% | -31.62% | 10.08% |
| 21～30 | 35.07% | 35.36% | 22.37% | 0.00% | 4.43% |
| 31～40 | 28.46% | 0.00% | -24.26% | 31.62% | 10.08% |
| 41～50 | 35.07% | -35.36% | 22.37% | 0.00% | 4.43% |
| 高相関 | | | | | |
| 1～10 | 52.78% | 50.00% | 26.49% | 0.00% | 3.61% |
| 11～20 | 34.84% | 0.00% | -33.43% | 38.73% | 13.00% |
| 21～30 | 29.78% | 0.00% | -15.69% | 0.00% | -43.21% |
| 31～40 | 34.84% | 0.00% | -33.43% | -38.73% | 13.00% |
| 41～50 | 52.78% | -50.00% | 26.49% | 0.00% | 3.61% |

以下、4節で説明した分散減少法や準モンテカルロ法の効果を計算事例で検証する。

(2) 分散減少法の効果

まず4節(2)で示したさまざまな分散減少法の効果を確認する。モンテカルロ法の誤差分散は(13)式の $V[EL^0(\mathbf{Y})]$ の大きさで決まる。そこで、表5で各手法の期待損失率の標準偏差

$$\sqrt{\frac{\sum_{l=1}^M [EL^0(\mathbf{Y}_l)]^2}{M} - \left[\frac{\sum_{l=1}^M [EL^0(\mathbf{Y}_l)]}{M} \right]^2}, \quad (23)$$

($M=5,000$)を比較した。表5の「Li」はLi[2000]の手法であり、それ以外は本稿で説明したツリー法を用いた準解析的手法(以下、ツリー法)による結果である。後者については(10)式の多次元積分に用いるモンテカルロ法として4手法をとりあげた。「MT」は一様乱数に擬似乱数(メルセイヌ・ツイスター)を用いた結果であり、「負の相関法」は「MT」に対する負の相関法、「LHC」はラテン・ハイパー・キューブ法(LHC法)、「Halton」は一様乱数に準乱数のHalton列を用いた結果である。

まず、無条件モンテカルロ法の結果である「Li」と条件付モンテカルロ法である「MT」を比べると、「MT」の方が標準偏差が小さく、条件付期待値を計算することによる分散減少効果が確認された。ただし、この分散減少効果は相関の大きさ(共通ファクター係数の絶対値の大きさ)に逆比例しており、相関が低いほど分散減少効果が大きくなるのがわかる。これは、ツリー法の場合、(4)式より共通ファクター係数の絶対値の大きさに比例して $V[p_i]$ (条件付デフォルト確率の分散)が大きくなり、結果として(10)式の期待損失率の分散を押し上げるためである。

次に、「負の相関法」、「LHC」を「MT」と比較してみると、負の相関法は分散減少効果が非常に大きいものに対し、LHC法はいずれのケースにおいても効果が小さかった。

表5 期待損失率の標本標準偏差

| トランシェ | 相関 | Li | MT | 負の相関法 | LHC | Halton |
|-------|----|--------|--------|--------|--------|--------|
| 0~3% | 低 | 33.67% | 10.41% | 0.95% | 10.18% | 10.32% |
| | 中 | 38.55% | 26.11% | 8.71% | 25.49% | 25.83% |
| | 高 | 41.06% | 31.95% | 12.92% | 31.46% | 31.74% |
| 6~9% | 低 | 8.46% | 1.62% | 0.85% | 1.53% | 1.61% |
| | 中 | 18.98% | 12.73% | 7.86% | 11.69% | 12.04% |
| | 高 | 24.90% | 20.40% | 12.80% | 18.48% | 19.54% |

図1～図6は、収束の過程を確かめるために、推定誤差の絶対値の推移を示したものであり、表2のスプレッド X を真の値と仮定し、各手法で M 回試行することにより得られたスプレッドを \hat{X}_M としたときの絶対誤差 $|\hat{X}_M - X|$ を表している³⁶。まず分散減少法の効果が小さかったLHC法でみると、図1、図3、図5のエクイティ（トランシェ：0～3%）の場合は8,000回以上になると誤差が十分小さくなっており、この点はメザニン（トランシェ：6～9%）でも同様である。ただし、図1～図6のいずれのケースでも「MT」に比べて優位性が高いわけではなく、本稿の枠組みでは有効な手法とはいえない。一方、「負の相関法」をみると、表5から予想されるように、相関水準、トランシェ、試行回数に関係なく「MT」と比べ誤差が小さい傾向がある。ただ、高相関のときのメザニン（図6）のように10,000回付近で誤差が拡大する現象も確認された。

（3）準モンテカルロ法の効果

表5では「MT」と「Halton」の標準偏差の値にあまり差がなかったのに対し、図1～図6をみると準モンテカルロ法の「Halton」は、どの相関水準、トランシェでも他の手法と比べ非常に速くかつ安定して収束していることがわかる。これは、4節(3)で指摘したように、準乱数を用いる準モンテカルロ法の収束のオーダーが、基本的なモンテカルロ法より小さいことに起因していると考えられる。

以上の分析により、本稿のモデル設定では、モンテカルロ法に対して分散減少法を使うと確かに分散減少効果が現れるが、準モンテカルロ法の収束スピードはそれをはるかに凌ぐことが確認された。

次に、準モンテカルロ法で用いる準乱数による収束の違いをみるために、Halton列、Faure列、Sobol列の収束性を図7～図12に示した。図1～図6と同様に、各準乱数を使った場合のトランシェ・スプレッドの絶対誤差の推移を表している。これらの図から、どの準乱数も3,000回程度で収束しており、かつ準乱数間の相違もほとんどないことがわかった³⁷。

（4）単位時間当たりの計算効率性

以上の分析より、ツリー法に準モンテカルロ法を用いると少ない試行回数で高い精度のCDOトランシェのスプレッドが得られることがわかった。ただ、4節(1)で指摘したようにツリー法はLi [2000] の手法と比べ、1回の試行当たりの計算時間が長くなるため、計算効率性を考えるうえでは計算時間単位で比較しなくてはならない。

36 この誤差にはモンテカルロ法の近似誤差に加え、5節の手法による、相関行列のファクター化による誤差も含まれている。

37 他のさまざまな相関、トランシェ、銘柄数で計算してみたところ、本稿で用いた（QunatLibの）改良型Halton列を用いると、多くの場合、他の列より速く収束する傾向があることがわかった。

図1 モンテカルロ法の収束 (低相関、トランシェ: 0~3%)

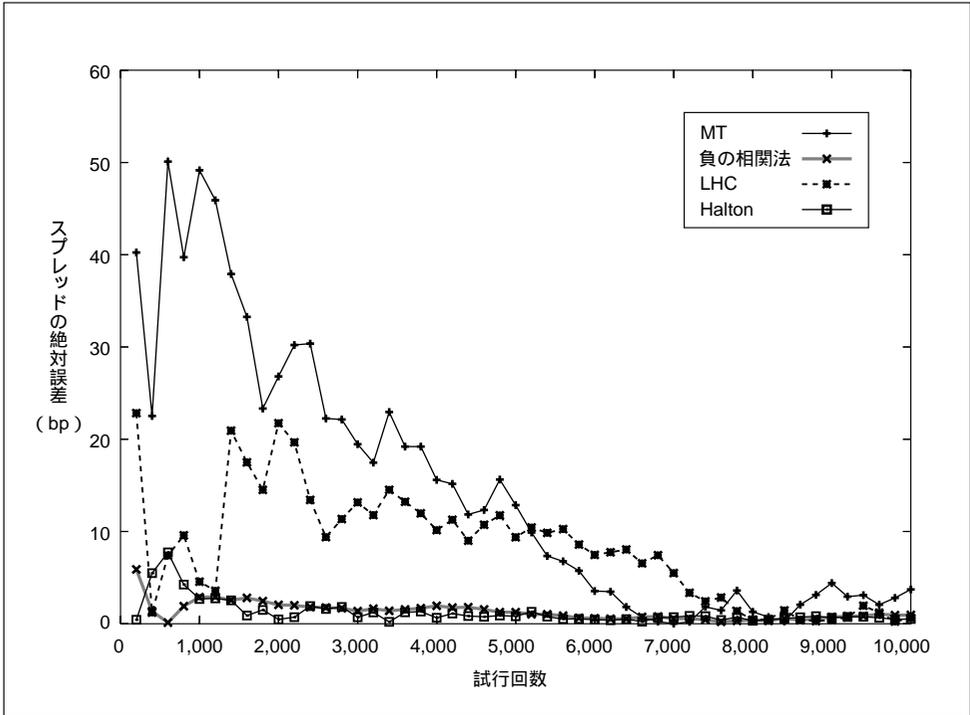


図2 モンテカルロ法の収束 (低相関、トランシェ: 6~9%)

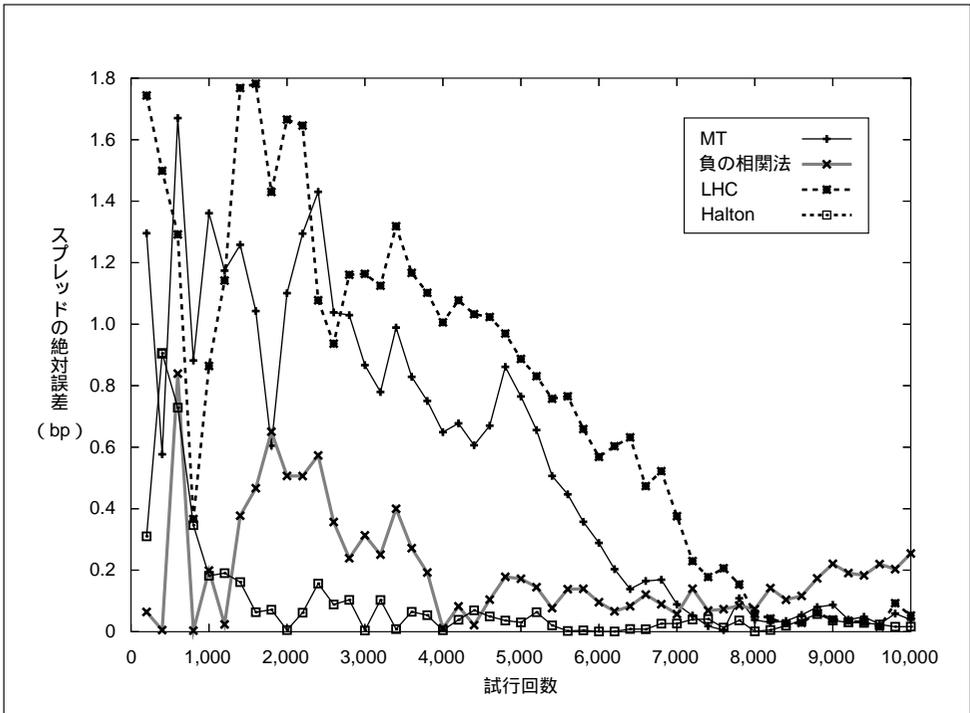


図3 モンテカルロ法の収束（中相関、トランシェ：0～3%）

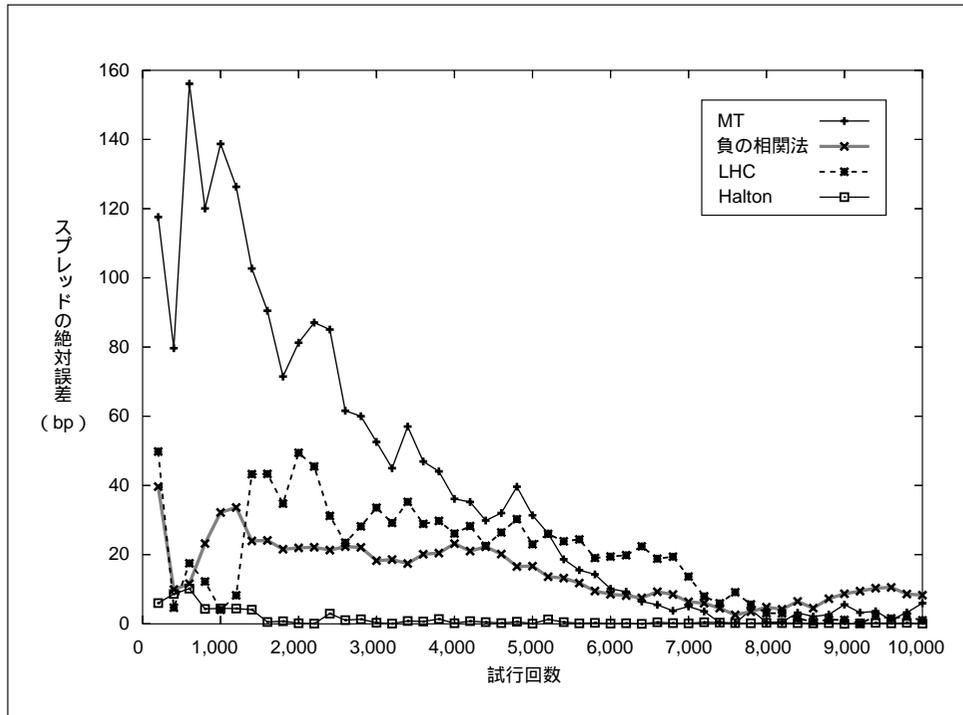


図4 モンテカルロ法の収束（中相関、トランシェ：6～9%）

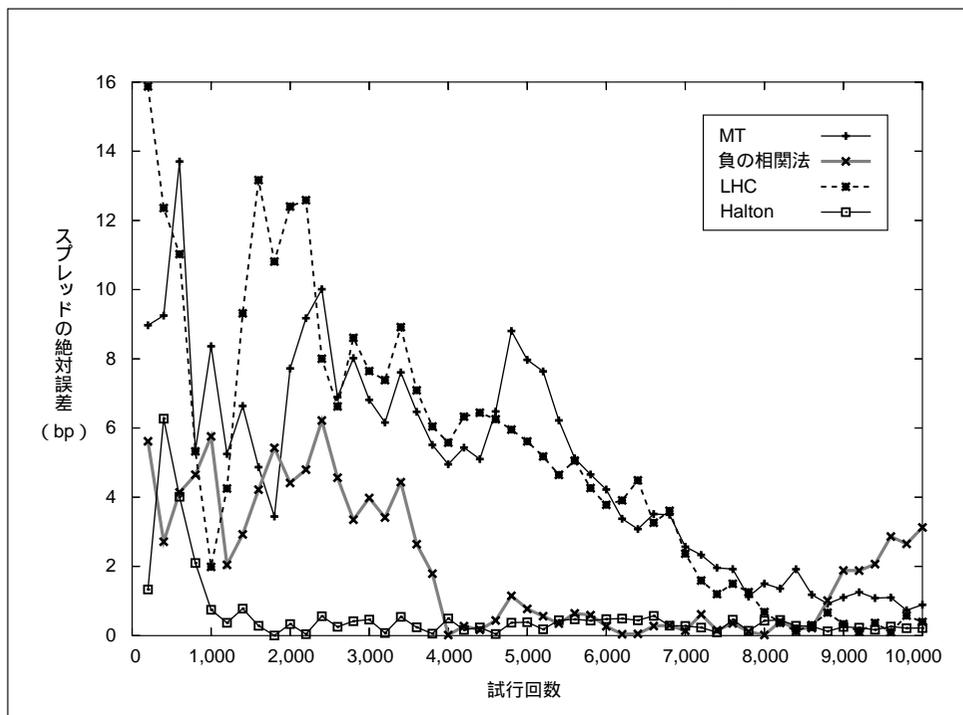


図5 モンテカルロ法の収束 (高相関、トランシェ: 0~3%)

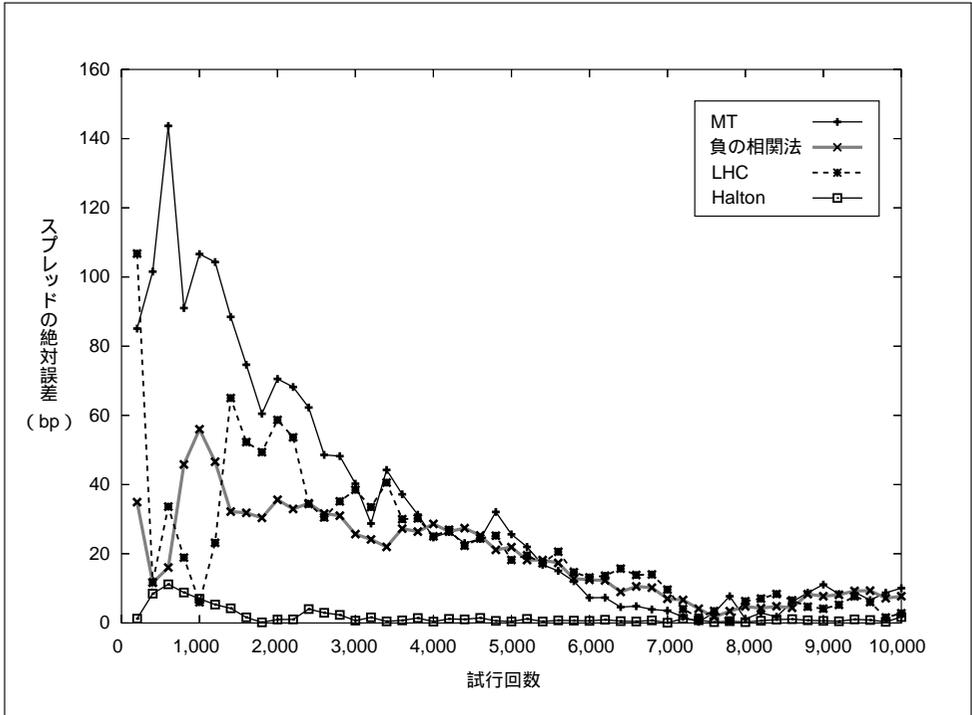


図6 モンテカルロ法の収束 (高相関、トランシェ: 6~9%)

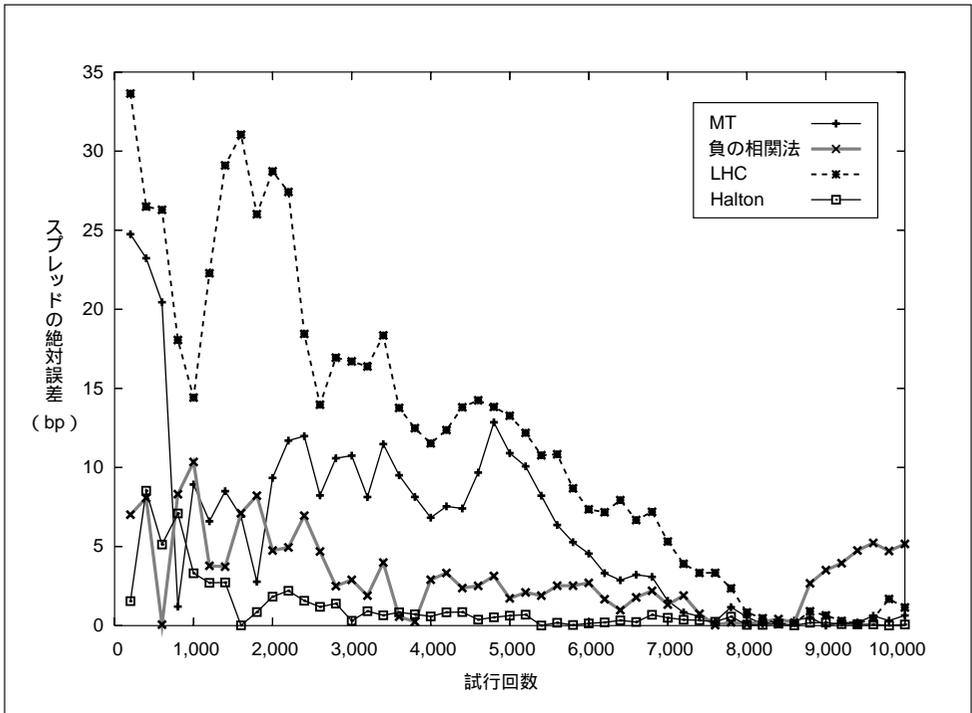


図7 準モンテカルロ法の収束（低相関、トランシェ：0～3%）

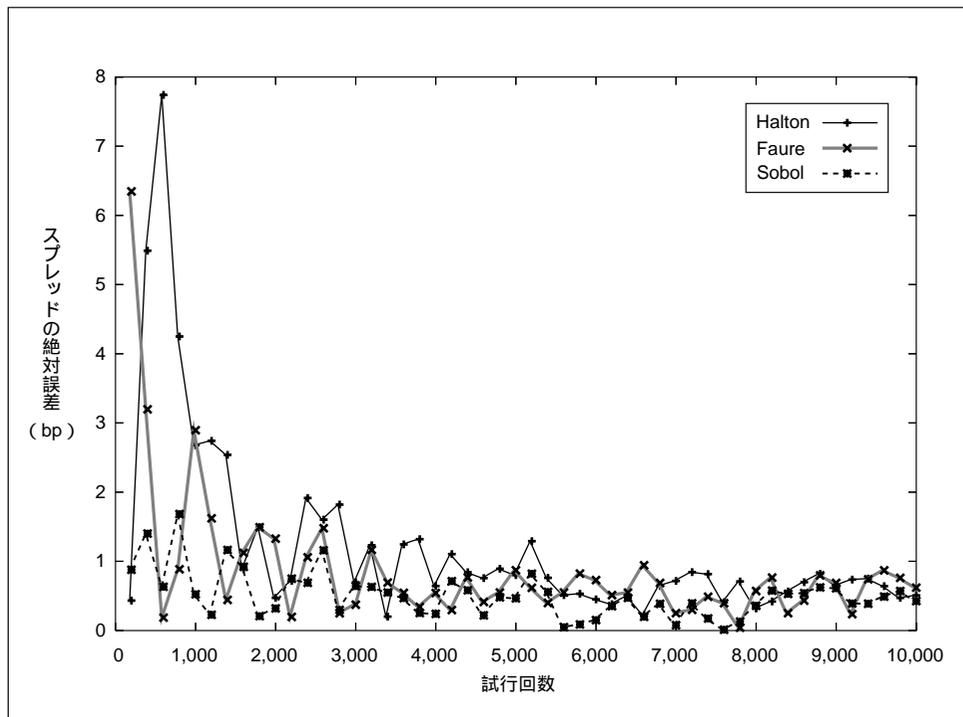


図8 準モンテカルロ法の収束（低相関、トランシェ：6～9%）

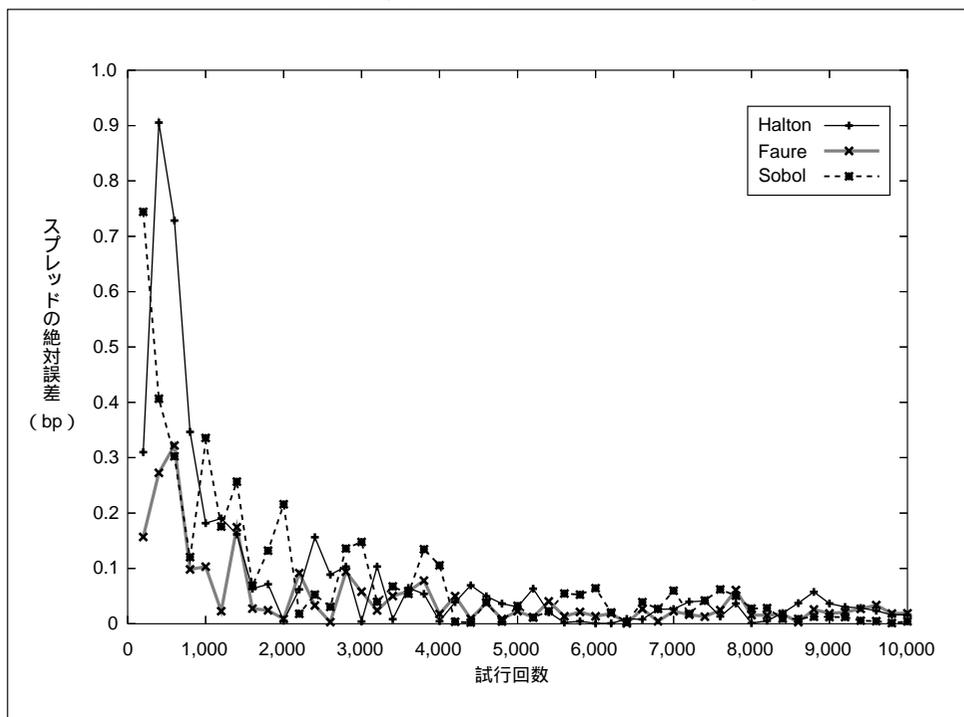


図9 準モンテカルロ法の収束 (中相関、トランシェ: 0~3%)

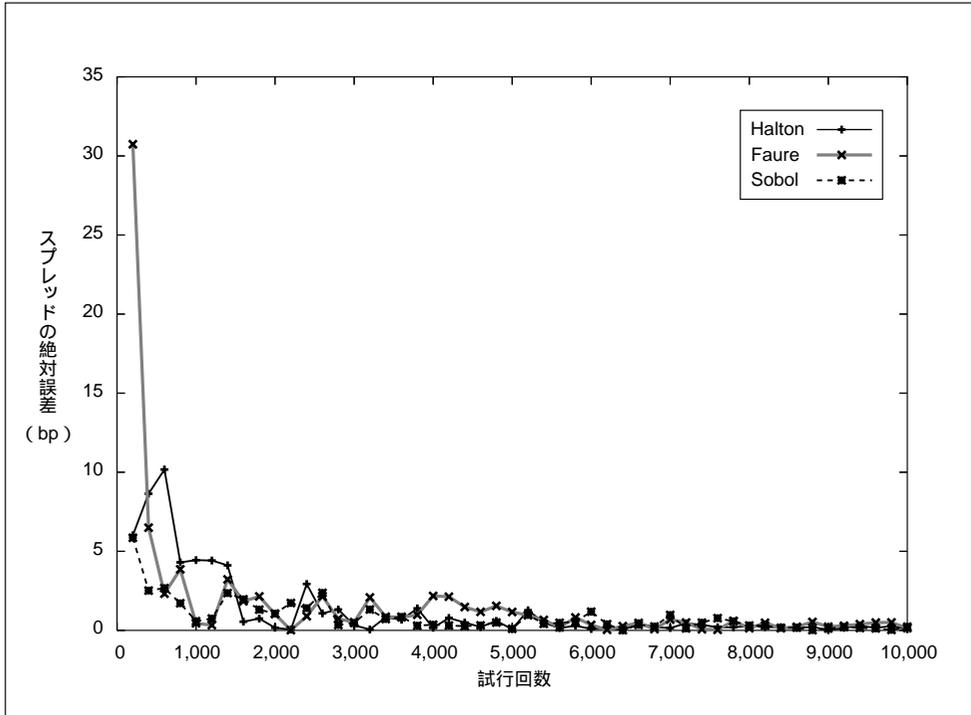


図10 準モンテカルロ法の収束 (中相関、トランシェ: 6~9%)

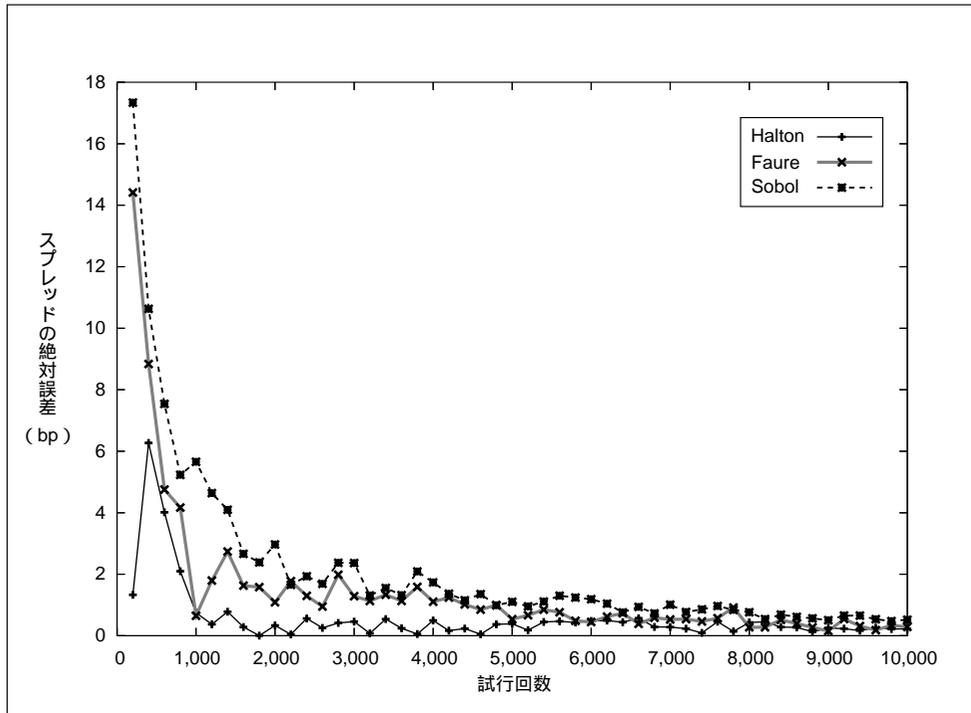


図11 準モンテカルロ法の収束（高相関、トランシェ：0～3%）

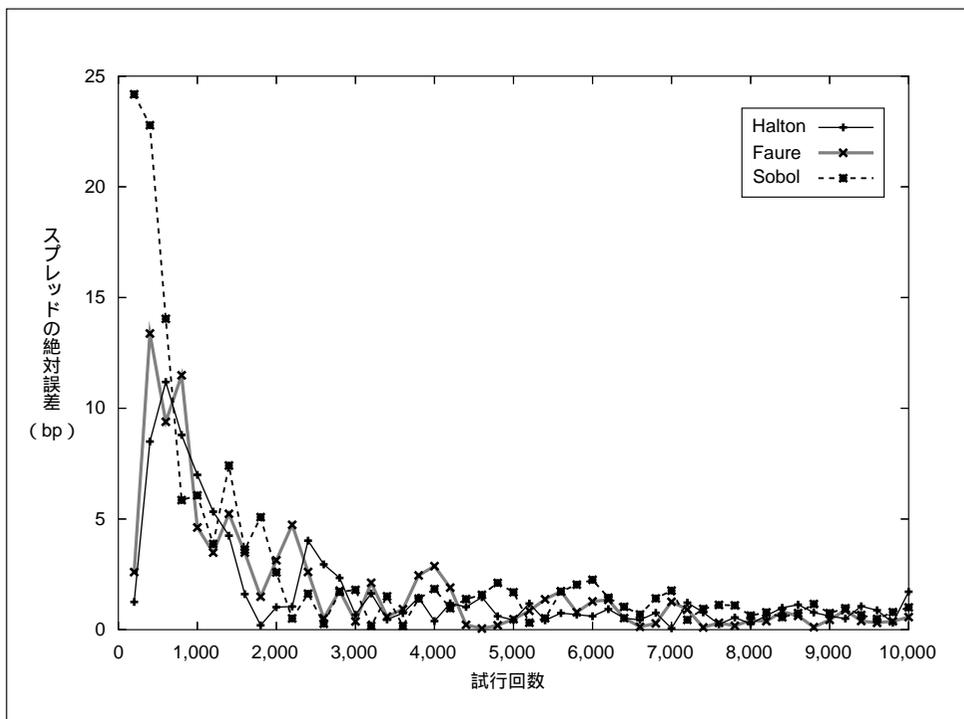
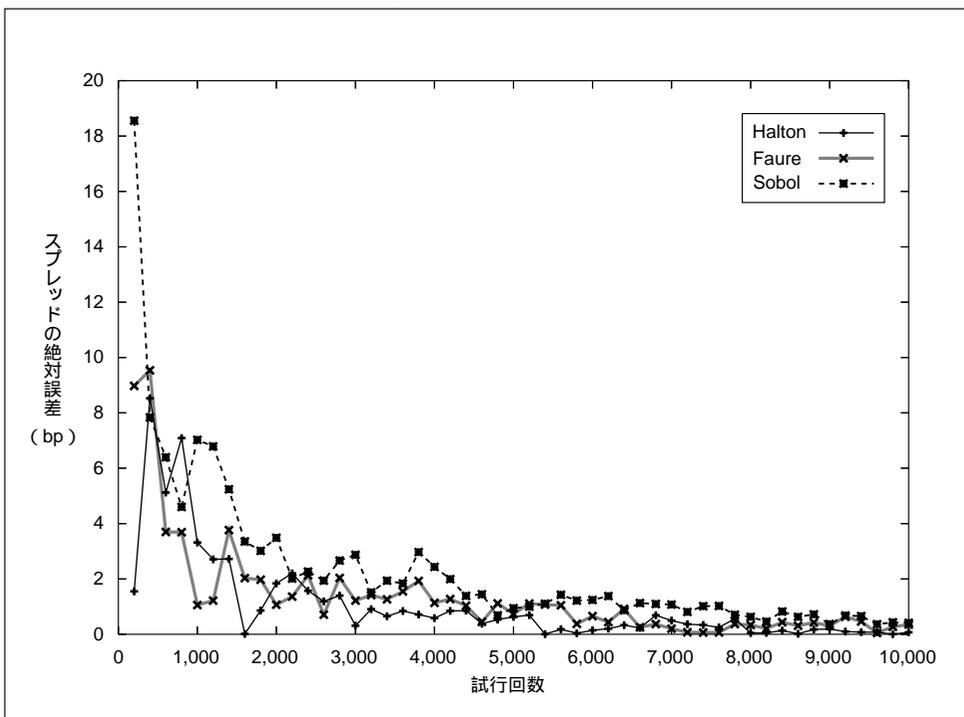


図12 準モンテカルロ法の収束（高相関、トランシェ：6～9%）



そこで、準モンテカルロ法（Halton列を使用）を用いたツリー法とLi [2000] の手法について、計算時間当たりの評価に変換した誤差推移を図13～図18に示した。時間の単位はツリー法で試行回数500回の計算に費やす時間を1単位としている³⁸。Li [2000] の手法で1単位時間中に実行することができる試行回数はエクイティ（0～3%）の場合は22,000回であり、メザニン（6～9%）の場合は25,000回であった。

図13～図18からは、Li [2000] の手法では、メザニン（6～9%）の場合は比較的少ない回数で誤差が小さくなっているものの安定性に欠け、エクイティ（0～3%）の場合は10単位（試行回数220,000回）でも収束していないことが確認される。それに対して、ツリー法ではどの場合も短い単位時間中に安定して収束していることがわかる。表2のスプレッド水準を参考に推定誤差の許容範囲をエクイティ（0～3%）の場合2bp、メザニン（6～9%）の場合1bpとすると、ツリー法は全ての場合で3単位以内（試行回数：1,500回、計算時間：3.93～4.41秒）で収束しており、計算時間で評価してもツリー法は優位性が高いことがわかった。

.....
 38 使用したPCでは、1単位の計算時間はエクイティ（0～3%）で1.31秒、メザニン（6～9%）で1.47秒であった。計算時間は両手法とも試行回数に比例する。エクイティとメザニンの計算時間の差は、3節のツリー・アルゴリズムに起因しており、メザニンよりもエクイティの方が計算負荷が少ないことから生じている。

図13 単位時間当たりの収束（低相関、トランシェ：0～3%）

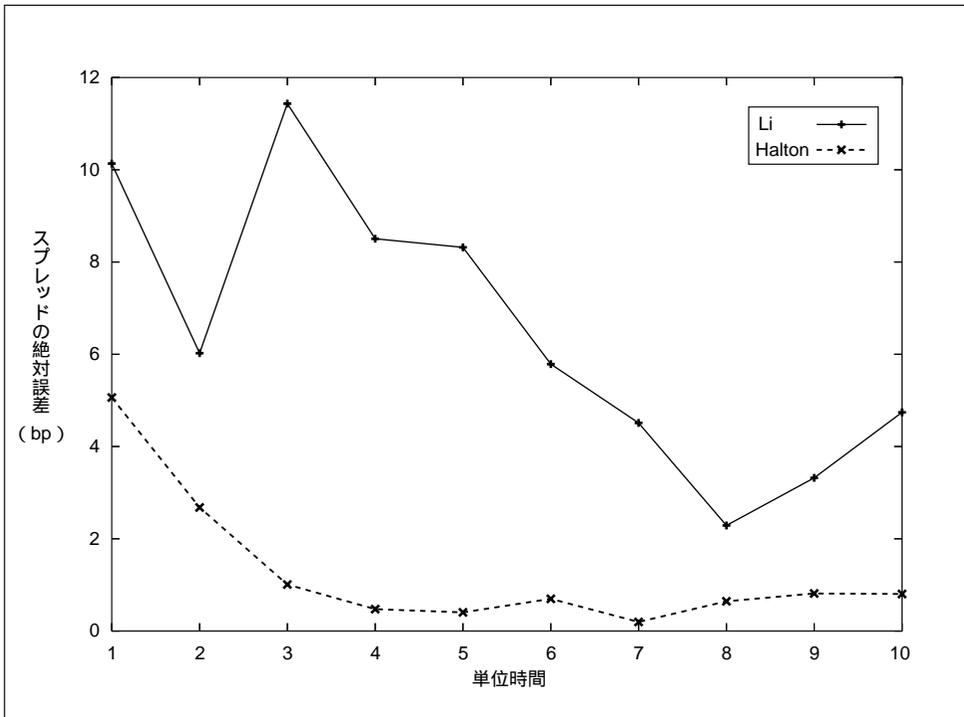


図14 単位時間当たりの収束（低相関、トランシェ：6～9%）

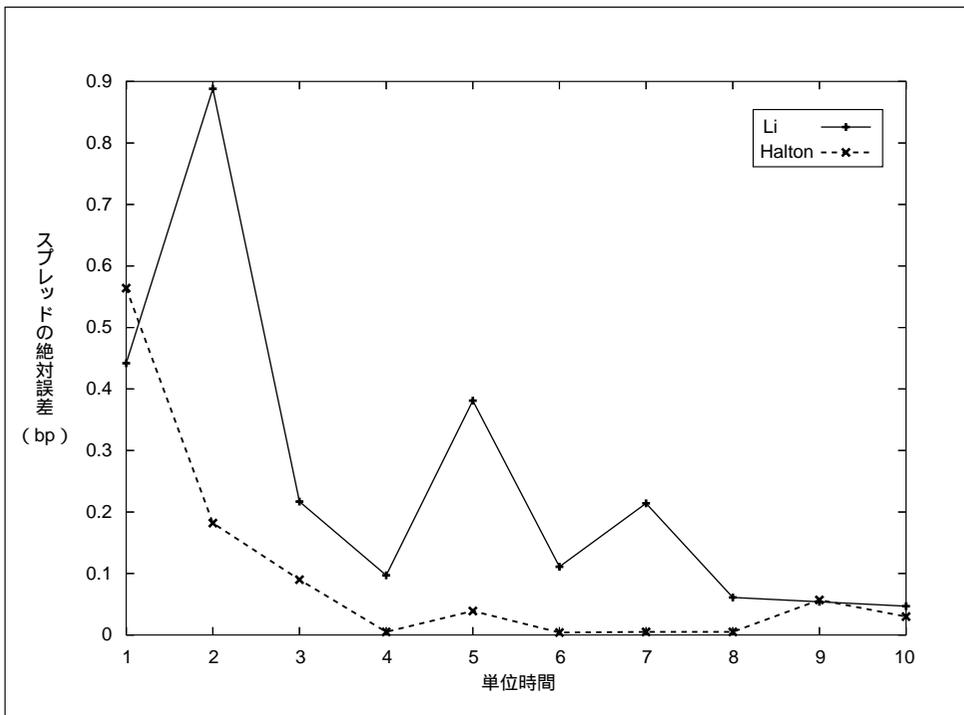


図15 単位時間当たりの収束 (中相関、トランシェ: 0~3%)

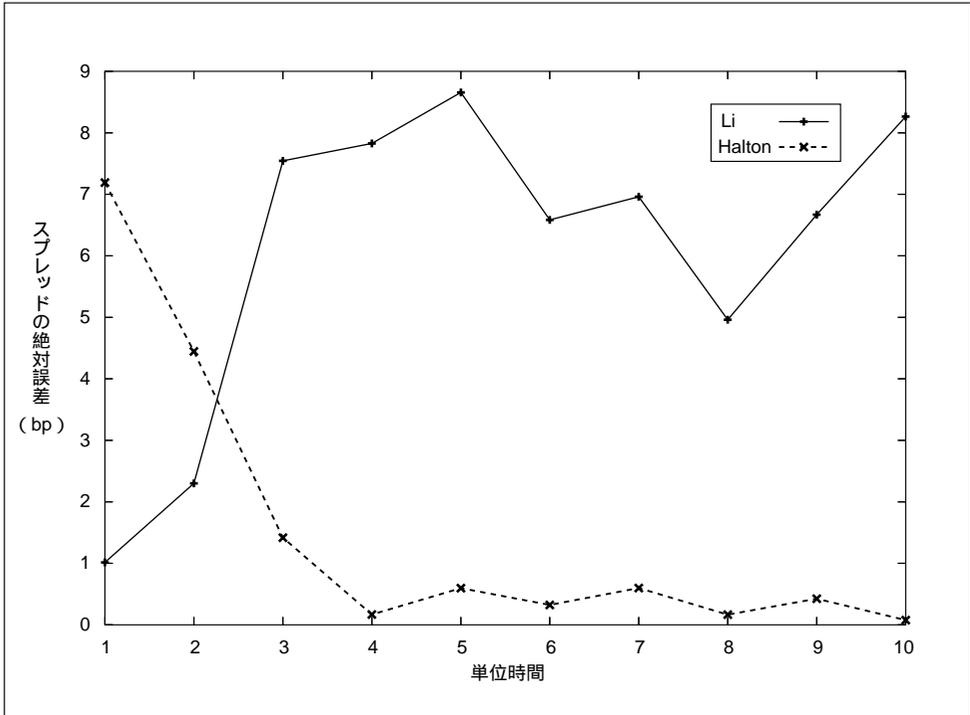


図16 単位時間当たりの収束 (中相関、トランシェ: 6~9%)

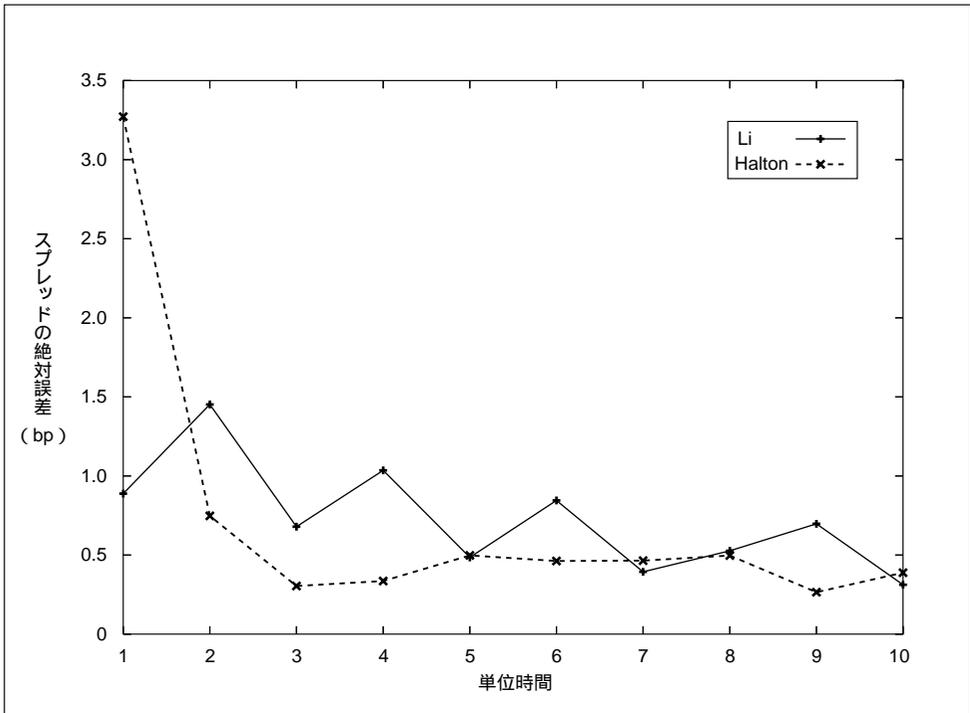


図17 単位時間当たりの収束（高相関、トランシェ：0～3%）

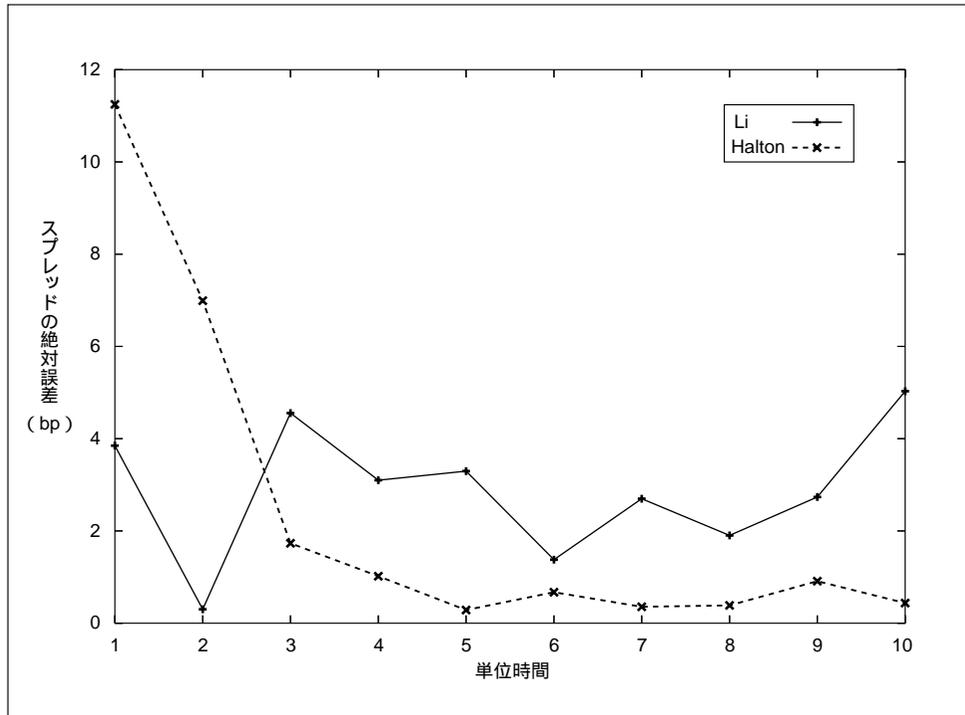
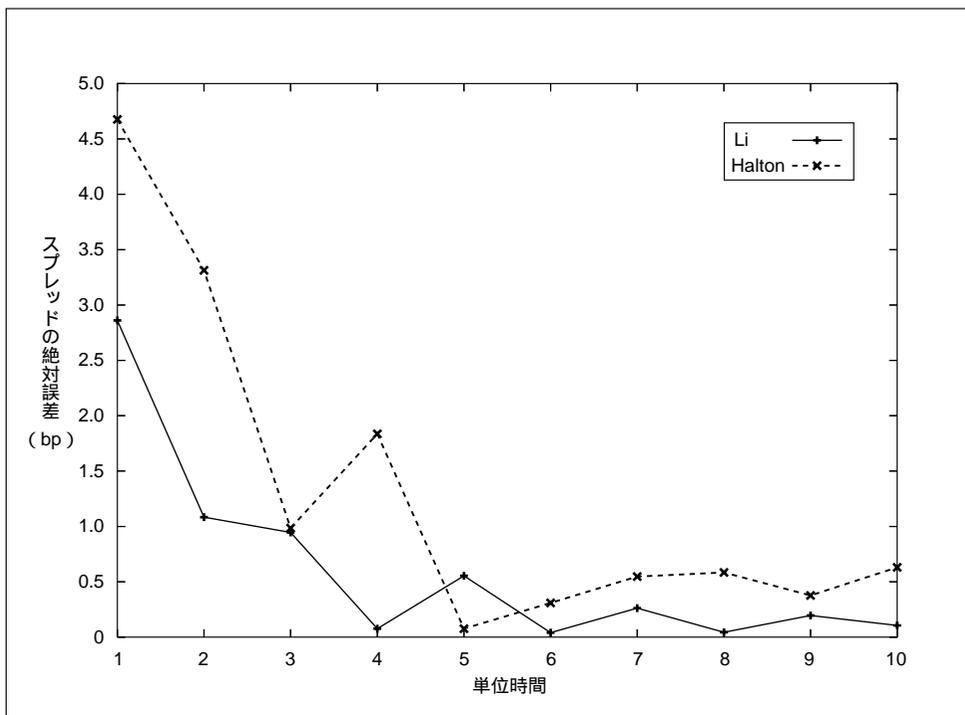


図18 単位時間当たりの収束（高相関、トランシェ：6～9%）



7. まとめ

本稿では、横谷 [2007] で示したCDO高速プライシング手法のマルチ・ファクター・モデルへの応用法を説明した。マルチ・ファクター・モデルの場合、ファクター数が3以内であれば、1ファクター・モデル(横谷 [2007])と同様に条件付期待値の積分で数値積分(ガウス=エルミート法)を用いた方がよい。一方、ファクター数が4以上となれば、モンテカルロ法を用いた方が効率的である。本稿では、モンテカルロ法の分散減少法と一様乱数に準乱数を用いる準モンテカルロ法について説明し、プライシング高速化の効果を計測したほか、CDOプライシングの標準的手法となっているLi [2000]の手法との比較を行った。その結果、準モンテカルロ法を用いることにより、試行回数単位、計算時間単位のいずれで評価してもCDOプライシングを既存の手法より効率的に行えることがわかった。

また、本稿ではAndersen, Sidenius, and Basu [2003]で提示された、相関行列からファクター係数を導出する効率的なアルゴリズム(ISM)を説明したうえで、最適ファクター数の導出方法($ISM+\alpha$)とその理論的枠組みを示した。同手法はファクター・モデルを用いる他の分野においても有用性が高いと考えられる。

補論1．無条件分散と条件付分散

4節(2)で条件付モンテカルロ法の分散減少効果について説明したが、この効果は以下のように理論的に示すことができる。証明は木島・長山・近江 [2000] に沿って説明する。

任意の確率変数を x と y とすると、

$$E[x] = E(E[x|y]),$$

が成り立つ。このとき、

$$V(x|y) = E(x^2|y) - (E[x|y])^2,$$

となるので、両辺の期待値をとると、

$$\begin{aligned} E[V(x|y)] &= E[E(x^2|y)] - E[(E[x|y])^2] \\ &= E[x^2] - E[(x|y)^2], \end{aligned}$$

となる。また、

$$\begin{aligned} V[E[x|y]] &= E[(E[x|y])^2] - (E[E[x|y]])^2 \\ &= E[(E[x|y])^2] - (E[x])^2, \end{aligned}$$

であるので、

$$V[x] = E[V(x|y)] + V[E[x|y]] \geq V[E[x|y]],$$

となる。ここで、 $V(x|y) > 0$ であれば、 $E[V(x|y)] > 0$ であり、 $V[x] > V[E[x|y]]$ が成立する。したがって、単純に確率変数をサンプリングするより、条件付期待値をサンプリングした方が誤差分散が減少する。

補論2．スペクトル分解とISMの理論的枠組み

本補論では、5節で説明したISM (Iterative Spectral decomposition Method) の理論的枠組みを説明する。

\mathbf{Z} を半正定値 (positive semi-definite) である $m \times m$ 対称行列とする。 \mathbf{Z} の固有値を $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m \geq 0$ とし、それらの対角行列を $\Delta = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_m)$ 、この固有値の順に並べた固有ベクトルの束 (行列) を Γ とすると次の等式が成り立つ。

$$\begin{aligned}\mathbf{Z}\Gamma &= \Gamma\Delta, \\ \mathbf{Z} &= \Gamma\Delta\Gamma^{-1} = \Gamma\Delta\Gamma^T.\end{aligned}\tag{A-1}$$

ここで、 $\Delta = \Lambda\Lambda^T$ となる対角行列 $\Lambda(\Lambda_{ii} = \sqrt{\lambda_i}, i = 1, \dots, m)$ を定義すると、(A-1)式より、

$$\mathbf{Z} = \Gamma\Lambda\Lambda^T\Gamma^T = (\Gamma\Lambda)(\Gamma\Lambda)^T,\tag{A-2}$$

が成り立つ。(A-2)式をスペクトル分解と呼ぶ。

また、任意の正方行列 \mathbf{R} のノルムを次のように定義する。

$$\|\mathbf{R}\|^2 = \sum_{ij} (\mathbf{R}_{ij})^2 = \text{tr}(\mathbf{R}\mathbf{R}^T).\tag{A-3}$$

上記の設定のもとで、対称行列に関して次の定理が成り立つ。

定理 (Harville [1999]) [最適なランク減少 (Optimal rank reduction)]

ランクが $k (\geq r)$ である $m \times m$ 対称行列を \mathbf{A} とし、 \mathbf{A} の 0 ではない固有値を $\lambda_1, \lambda_2, \dots, \lambda_k (|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_k|)$ 、固有ベクトルの束 (行列) を Γ とする。

ここで、

$$\mathbf{A} = \Gamma\Delta\Gamma^T, \text{ where } \Delta = \text{diag}(\lambda_1, \dots, \lambda_k, 0, \dots, 0),\tag{A-4}$$

とする。このとき、ランクが r である任意の $m \times m$ 行列を \mathbf{B} とすると、次の関係式が成り立つ。

$$\|\mathbf{B} - \mathbf{A}\|^2 \geq \lambda_{r+1}^2 + \dots + \lambda_k^2.\tag{A-5}$$

上式の等式は次のときに成り立つ。

$$\mathbf{B} = \Gamma\hat{\Delta}\Gamma^T, \text{ where } \hat{\Delta} = \text{diag}(\lambda_1, \dots, \lambda_r, 0, \dots, 0).\tag{A-6}$$

以上の準備のもとで、ISMの収束性について考える。

ISMアルゴリズムの s 回目の $\hat{\mathbf{A}}_s$ について以下のことが成り立つ。

$$\begin{aligned}\hat{\mathbf{A}}_s \hat{\mathbf{A}}_s^T &= \left(\tilde{\Gamma}_s \sqrt{\tilde{\Lambda}}_s \right) \left(\tilde{\Gamma}_s \sqrt{\tilde{\Lambda}}_s \right)^T \\ &= \Gamma_s \tilde{\Delta}_s \Gamma_s.\end{aligned}\tag{A-7}$$

ただし、 Γ_s は $\Sigma - \hat{C}_{s-1}$ の固有値の大きさ順に並べた固有ベクトルの $m \times m$ 行列で、 $\bar{\Delta}_s$ は $\Sigma - \hat{C}_{s-1}$ の固有値 ($\lambda_1^s \geq \lambda_2^s \geq \dots \geq \lambda_m^s$) のうち、 $\lambda_{z+1}^s, \dots, \lambda_m^s$ を 0 として (z はファクター数) それらに対角要素とした行列 ($\bar{\Delta}_s = \text{diag}(\lambda_1^s, \lambda_2^s, \dots, \lambda_z^s, 0, \dots, 0)$) とする。

ここで、 $\mathbf{W}_s = \Sigma - \hat{C}_{s-1} - \hat{A}_s \hat{A}_s^T$ 、 $\mathbf{D}_s = \hat{C}_s - \hat{C}_{s-1}$ という2つの行列を定義する。このとき、 $\hat{C}_s = \mathbf{I} - \text{diag}(\hat{A}_s \hat{A}_s^T)$ より、

$$\begin{aligned} \mathbf{D}_s &= \text{diag}(\mathbf{I} - \hat{A}_s \hat{A}_s^T) - \hat{C}_{s-1} \\ &= \text{diag}(\Sigma - \hat{A}_s \hat{A}_s^T) - \hat{C}_{s-1} \\ &= \text{diag}(\mathbf{W}_s), \end{aligned} \quad (\text{A-8})$$

となり (2つ目の等式は Σ の対角要素が全て1であることから成り立つ)

$$\|\mathbf{W}_s - \mathbf{D}_s\| \leq \|\mathbf{W}_s\|, \quad (\text{A-9})$$

が成り立つ。また、 \mathbf{W}_{s+1} と $(\mathbf{W}_s - \mathbf{D}_s)$ について、

$$\begin{aligned} \|\mathbf{W}_{s+1}\| &= \|\Sigma - \hat{C}_s - \hat{A}_{s+1} \hat{A}_{s+1}^T\|, \\ \|\mathbf{W}_s - \mathbf{D}_s\| &= \|\Sigma - \hat{C}_s - \hat{A}_s \hat{A}_s^T\|, \end{aligned}$$

となるので、(A-7)式と定理 (Harville [1999]) より、ある正の定数 a が存在し、

$$0 \leq a \leq \|\mathbf{W}_{s+1}\| \leq \|\mathbf{W}_s - \mathbf{D}_s\|, \quad (\text{A-10})$$

が成立する。(A-9)式、(A-10)式より、

$$0 \leq a \leq \|\mathbf{W}_{s+1}\| \leq \|\mathbf{W}_s - \mathbf{D}_s\| \leq \|\mathbf{W}_s\|, \quad (\text{A-11})$$

が成り立つことにより、

$$\lim_{s \rightarrow \infty} \|\mathbf{D}_s\| = \lim_{s \rightarrow \infty} \|\hat{C}_s - \hat{C}_{s-1}\| = 0, \quad (\text{A-12})$$

と同時に、

$$\lim_{s \rightarrow \infty} \|\mathbf{W}_s - \mathbf{D}_s\|^2 = \lim_{s \rightarrow \infty} \|\Sigma - \hat{C}_s - \hat{A}_s \hat{A}_s^T\|^2 = a^2 \geq 0, \quad (\text{A-13})$$

が成り立つ。したがって、ISMを用いることで、5節の(19)式を満たす $\bar{\mathbf{A}}$ を導出できることが示せた。

また、(A-10)式、(A-11)式、(A-13)式について $a = 0$ が成り立つのは、定理 (Harville [1999]) より、 $\hat{\Sigma} - \hat{C}_\infty$ のランクがファクター数 z 以下であるときのみである。したがって、ISM + α によって、 $z=1, 2 \dots$ の順に(22)式の $Er(z)$ を計算していけば、最適ファクター数を求めることができる。

参考文献

- 木島正明・長山いづみ・近江義行、『ファイナンス工学入門第 部：数値計算法』、日科技連出版社、2000年
- 小宮清孝、「CDOのプライシング・モデルとそれを用いたCDOの特性等の考察：CDOの商品性、国内市場の概説とともに」、『金融研究』第22巻別冊第2号、日本銀行金融研究所、2003年、89～130頁
- 湯前祥二・鈴木輝好、『モンテカルロ法の金融工学への適用』、朝倉書店、2000年
- 横谷進弥、「CDOプライシングの離散高速アプローチ(1)：ツリーを用いた準解析的プライシングの1ファクター・モデルへの適用」、『金融研究』第26巻別冊第2号、日本銀行金融研究所、2007年、23～44頁(本号所収)
- Andersen, L., J. Sidenius, and S. Basu, "All your hedges in one basket," *Risk*, 16 (11), 2003, pp. 67-72.
- Gerstner, T., and M. Griebel, "Numerical integration using sparse grids," *Numerical Algorithms*, 18 (3-4), 1998, pp. 209-232.
- Glasserman, P., *Monte Carlo Methods in Financial Engineering*, Springer-Verlag New York, Inc., 2004.
- , and J. Li, "Importance sampling for portfolio credit risk," *Management Science*, 51 (11), 2005, pp. 1643-1656.
- Harville, D. A., *Matrix Algebra From a Statistician's Perspective*, Springer, 1999.
- Jäckel, P., *Monte Carlo methods in finance*, Wiley, 2002.
- Laurent, J. P., and J. Gregory, "Basket default swaps, CDOs and factor copulas," *Journal of Risk*, 7 (4), 2005, pp.103-122.
- Li, D. X., "On default correlation: a copula function approach," *Journal of Fixed Income*, 9 (4), 2000, pp. 43-54.
- McKay, M.D., W.J. Conover, and R.J. Beckman, "A comparison of three methods for selecting input variables in the analysis of output from a computer code," *Technometrics*, 21 (2), 1979, pp. 239-245.
- Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing Second Edition*, Cambridge University Press, 1994.
- Schönbucher, P. J., *Credit derivatives pricing models: models, pricing and implementation*, Wiley, 2003.
- Stein, M., "Large sample properties of simulation using Latin hypercube sampling," *Technometrics*, 29 (2), 1987, pp. 143-151.