高速ガウス変換を用いた天候デリバティブの価格計算手法

山 本 有 作† 恵 木 正 史††

気象変動リスクを回避・低減するための金融派生商品「天候デリバティブ」の価格計算高速化のため，高速ガウス変換と呼ばれる数値計算手法を用いた新しいアルゴリズムを開発した．本手法では，計算時間を τ とするとき，計算した価格の誤差が $1/\tau$ のオーダーで減少し，従来広く使われてきたモンテカルロ法の $1/\sqrt{\tau}$ に対し，収束性が大きく改善される．本手法を C 言語を用いて実装し，PC 上で評価したところ，CDD デリバティブと呼ばれる天候デリバティブの価格を計算する場合で，モンテカルロ法の 10 倍以上の高速化が得られた．本手法は，天候デリバティブの最適設計，時価評価など，従来多大な計算時間が必要であった用途に適用可能である．

Valuation of Weather Derivatives using the Fast Gauss Transform

YUSAKU YAMAMOTO† and MASASHI EGI††

We developed a fast algorithm for pricing weather derivatives, which are financial products for hedging weather risks, using the fast Gauss transform. With our algorithm, the error in the computed price decreases as $O(1/\tau)$ as a function of computational time τ, which is considerably faster than the rate of $O(1/\sqrt{\tau})$ for the conventional Monte Carlo method. We implemented our method on a PC using the C language and found that it is more than 10 times faster than the MC method when computing the price of CDD derivatives. Our algorithm opens a way to applications which have been impractical due to a large amount of computation, such as optimal design of derivatives or estimation of the current market price of a portfolio of derivatives.

1. はじめに

気温，降水量，積雪量などの気象条件は，企業の活動に大きな影響を及ぼす．たとえば夏の猛暑は，エアコンメーカーや電力会社にとっては売上増加の要因となるが，デパートや鉄道会社にとっては，冷房費増大による利益減少の要因となる．また，例年に比べて積雪が少ない場合，自治体にとっては除雪費用が少なくて済むメリットがあるが，スキー場のホテルやスキー用品メーカーにとっては，売上減少というデメリットがある．このような気象条件による売上・利益の変動を気象リスクと呼ぶ．

気象リスクを軽減し，収益の安定化を図る手段として，近年，天候デリバティブと呼ばれる金融派生商品が注目されている[9)10)15)]．天候デリバティブとは，気象リスクにさらされている事業体が損害保険会社などの発行体に予め契約料金を支払い，将来一定の気象条件が満たされた場合に補償金を受け取る契約である．たとえば，気温に関するデリバティブの例としては，「7 月中に 20 度以下の日があった場合，1 日につき 100 万円の補償金を支払う」という契約が考えられる．このようなデリバティブは，エアコンメーカーが冷夏による売上減少を補填するために活用できる．天候デリバティブは 1997 年に米国エンロン社が開発した比較的新しい商品であるが，市場は急速に拡大しており，2002 年には欧米で 110 億ドル（補償金額ベース），国内でも数百億円の市場規模となっている．

発行体が天候デリバティブを販売するに当たっては，将来の気象条件を予測し，支払うべき補償金の期待値を計算することにより，妥当な契約料金，すなわちデリバティブの価格を決める必要がある．従来，この期待値計算は，気温・降水量などに対する確率的なモデルを用い，モンテカルロ法で多数の気象シナリオを発生させることにより行われてきた．この手法に基づく天候デリバティブ価格計算システムとして，既に米 Climetrix 社の Climetrix Ver. 3.3[5)]，米 Financial Engineering Associates 社の@ENERGY/Weather[7)]，(株) 日立製

† 名古屋大学大学院工学研究科 計算理工学専攻
Department of Computational Science and Engineering, Nagoya University
†† 株式会社日立製作所中央研究所
Central Research Laboratory, Hitachi Ltd.

作所の Hi-Deri[11] などが製品化されている．

モンテカルロ法で価格を精度良く求めるには，10^8 通り程度の気象シナリオ発生が必要であり，PC 上で計算を行った場合，1 個のデリバティブの価格計算に数分の時間を要する．これは，価格を求めたいデリバティブの条件が予め定まっており，価格計算が 1 回で済む場合には許容できる時間である．しかし実務では，購入者の要求に合ったデリバティブを設計するため，何度も条件を変えて価格計算を繰り返す場合も多い．また，損害保険会社をはじめとする発行体が自社の持つ天候デリバティブ契約の時価評価を行う場合には，数百〜数千個のデリバティブの価格計算を一度に行う必要がある．このような目的のためには，従来のモンテカルロ法による価格計算は時間が掛かりすぎ，実用的でない．もちろん，モンテカルロ法は高い並列性を持つため，複数台の PC を用いて並列計算を行えば計算時間を台数にほぼ反比例して短縮できる．しかし，このように大きな計算資源に頼ることなく，アルゴリズムの改良により高速化を達成できれば，そのほうが望ましいことは明らかである．

そこで本論文では，天候デリバティブの価格計算高速化のため，モンテカルロ法に代わる新たなアルゴリズムを提案する．最近，分子動力学の分野で提案された高速ガウス変換と呼ばれる数値計算手法[1)8)14)] が金融デリバティブの価格計算高速化にも有効であることが示されており[2)3)]，アメリカンオプション，ルックバックオプションなど種々のデリバティブの価格計算がモンテカルロ法に比べて数倍〜数十倍に高速化できることが報告されている．そこで本論文では，天候デリバティブの中でも取引量の多い CDD デリバティブを取り上げ，従来の価格計算法を定式化し直すことにより，高速ガウス変換が適用可能であることを示す．これにより，高速な価格計算アルゴリズムが構成できる．

以下では，まず第 2 章で CDD デリバティブの説明を行い，従来のモンテカルロ法による価格計算法とその問題点を説明する．次に第 3 章で，高速ガウス変換を用いた新しい価格計算法を提案する．第 4 章では提案手法に対する数値実験を行い，モンテカルロ法と計算時間・計算精度を比較する．最後に第 5 章でまとめを行う．

2. 従来の価格計算法とその問題点

2.1 気温天候デリバティブの定義

本論文では，気温天候デリバティブのうち，市場で活発に取引されている CDD（Cooling Degree Days）デリバティブを対象として，高速価格計算手法を提案する．本節では，まず一般的な気温デリバティブを定義し，必要な用語についても定義を行った後，CDD デリバティブの説明を行う．

一般に気温デリバティブを定義するには，次の 7 つの条件を定義する必要がある[16)]．

観測期間	：○月○日から△月△日までの N 日間
観測地点	：○○市
観測指標	：W（℃）
行使値	：S（℃）
ティック値	：k（万円／℃）
種別	：プットまたはコール
契約料金	：Q（万円）

ここで，観測期間・観測地点は，補償金額算定の基となる気温が測定される期間・地点である．また，観測指標 W とは，補償金額算定に使われる観測期間中の気温 $\{T_1, T_2, \ldots, T_N\}$ の関数であり，用途に応じて平均気温 $\frac{1}{N}\sum_{i=1}^{N} T_i$，最高気温 $\max\{T_1, T_2, \ldots, T_N\}$，あるいは後に述べる CDD などが使われる．行使値 S とは，補償金支払いが開始されるための閾値であり，コールオプションと呼ばれるデリバティブでは W が S を上回った場合，プットオプションでは下回った場合に支払いが開始される．また，ティック値 k とは，W が S を 1 単位上回った場合に支払われる金額である．

以上の記号を用いて，コールオプションの補償金額 P_{call} およびプットオプションの補償金額 P_{put} は次のように定義される．

$$P_{call} = k \cdot \max(W - S, 0) \tag{1}$$

$$P_{put} = k \cdot \max(S - W, 0) \tag{2}$$

CDD デリバティブとは，上記の気温デリバティブのうちで，観測指標 W として次の式で定義される量 CDD を使ったデリバティブである．

$$CDD = \sum_{i=1}^{N} \max(0, T_i - \bar{T}) \tag{3}$$

ここで $\bar{T}$ は基準温度と呼ばれる定数である．CDD は，基準温度より気温が高い日が多いほど，また基準温度の超え方が大きいほど大きな値を取る．したがって，たとえば猛暑により冷房費が増大するデパートや鉄道会社は，前もって CDD デリバティブを購入しておくことにより，補償金によって猛暑の際のコスト増加分を補填することができる．

なお，CDD に類似した観測指標として，次の式

$$HDD = \sum_{i=1}^{N} \max(0, \bar{T} - T_i) \tag{4}$$

で定義される HDD（Heating Degree Days）もあり，厳寒による気象リスクの低減などに利用できる．HDD デリバティブの価格計算は CDD デリバティブとほぼ同様に行えるため，本論文では CDD デリバティブのみを取り上げる．

2.2 気温モデルと価格計算法

CDD デリバティブの価格を計算するには，観測期間中の気温 $\{T_1, T_2, \ldots, T_N\}$ を何らかの方法により予測し，それに基づいて発行体が支払う補償金額の期待値を求める必要がある．この目的のため，気温の予測モデルがいくつか提案されており[4)6)9)16)]，中でも Dischel モデル[6)] と呼ばれる次の確率モデルが最も単純かつ有効なモデルとして広く使われている．

$$T_n = (1-\beta)\Theta_n + \beta T_{n-1} + \epsilon_n, \quad \epsilon_n \sim N[\mu, \sigma^2] \tag{5}$$

ここで，T_n は観測期間中の第 n 日目の気温，Θ_n は第 n 日目にあたる日付の平年気温（たとえばこの日が 2 月 1 日ならば，過去数年間の 2 月 1 日の平均気温）である．また，パラメータ β，μ，σ は定数であり，各 ϵ_n は平均 μ、標準偏差 σ の正規分布に従う独立な確率変数である．

β，μ，σ は，気温の時系列を最も良く再現するように過去のデータから最小二乗法を用いて決めるのが一般的である．また，気象庁などから発表される長期予報を取り込んで決める方法も提案されており，予測精度の向上に有効であることがわかっている[10)15)]．

デリバティブの価格 Q は，(5) の確率モデルの下での補償金額 P_{call}（または P_{put}）の期待値を求め，それに発行体が定めるプレミアム e を加えることにより算出される．すなわち，

$$Q = E[P_{call}] + e \tag{6}$$

である．

この期待値を計算するには，通常，モンテカルロ法を用いる．すなわち，T_1 から始めて正規分布に従う乱数 $\epsilon_2, \epsilon_3, \ldots, \epsilon_N$ を用いて気温シナリオ $T_1, T_2, \ldots, T_N$ を生成し，この気温シナリオの下での補償金額 P_{call} を (3)，(1) より求める．この処理を多数の気温シナリオに対して行い，平均を取ることにより，期待値 $E[P_{call}]$ を計算する．シナリオ数 L を増やしていったとき，モンテカルロ法で計算した期待値は，真の期待値へと収束する．

2.3 従来法の問題点

モンテカルロ法による価格計算法は，実装が容易であること，観測指標を変えた場合への対応が柔軟にできることなどの理由から，広く使われている．

しかし，モンテカルロ法には，計算解の真の解への収束が遅いという問題点がある．実際，CDD デリバティブの価格を有効数字 3 桁の精度で求めるには，10^8 通り程度のシナリオが必要であることが経験的にわかっている．現在広く使われている Pentium 4 の PC を用いた場合，この計算には数分の時間がかかる．

実務では，この計算速度では遅すぎて実用的でない場合も多い．たとえば，天候デリバティブを販売する損害保険会社では，顧客から引き合いがあると，行使値，基準温度等の条件を変えた数多くの価格計算を行って，顧客要求に合った最適なデリバティブを設計する．この場合，モンテカルロ法を用いて計算すると，顧客への回答までに 1 時間以上を要してしまう．また，リスク管理のために自社の所有する天候デリバティブの時価評価を行う場合には，数百〜数千の価格計算を一度に行う必要があるが，これにも 1 日〜数日の時間を要してしまう．

このような収束の遅さの原因を調べるため，モンテカルロ法における計算解と真の解との誤差 E が，計算時間 τ の関数としてどのように減少するかを考察する．まず，E はシナリオ数 L の関数として $O(1/\sqrt{L})$ のオーダーで減少することが一般的に示されている[13)]．次に，デリバティブの観測期間 N を固定して考えると，モンテカルロ法の計算量はシナリオ数 L に比例するから，計算時間 τ も L に比例する．以上より，E は $O(1/\sqrt{\tau})$ のオーダーで減少し，$E \cong c_1/\sqrt{\tau}$（ただし c_1 は定数）と書けることがわかる．これは，たとえば誤差 E を 1/10 にしたい（すなわち有効数字を一桁増やしたい）場合には，計算時間 τ を 100 倍にしなくてはならないことを意味する．

このような収束の遅さの問題を解決し，デリバティブの最適設計，時価評価などを実用的な時間で行うための方法として，より高い収束のオーダーを持つ価格計算法，たとえば誤差 E が計算時間 τ に対して $O(1/\tau)$ のオーダーで減少するような計算アルゴリズムを開発することが考えられる．

3. 高速ガウス変換を用いた価格計算法の提案

3.1 原　理

本章では，CDD デリバティブの価格計算をモンテカルロ法によらずに数値積分と漸化式を用いて行う方法を示す．この方法に対して，高速ガウス変換と呼ばれる数値計算法を適用することにより，誤差 E が計算時間 τ に対して $O(1/\tau)$ のオーダーで減少するような価格計算アルゴリズムが構成できる．

いま，CDD の第 1 日目から n 日目までの途中累積値を

$$C_n = \sum_{i=1}^{n} \max(0, T_i - \bar{T}) \tag{7}$$

と定義する．(3) の定義より，C_N が CDD である．

いま，第 $n-1$ 日目の気温が T_{n-1} であるという条件の下で第 n 日目の気温が T_n である条件付確率密度を $P_n(T_n|T_{n-1})$ とすると，(5) において ϵ_i が正規分布 $N[\mu, \sigma^2]$ に従うことより，

$$P_n(T_n|T_{n-1}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(T_n-\mu_n)^2}{2\sigma^2}\right\}, \tag{8}$$

ただし

$$\mu_n = (1-\beta)\Theta_n + \beta T_{n-1} + \mu \tag{9}$$

となる．

次に，第 $n-1$ 日目において気温が T_{n-1}，CDD の途中累積値が C_{n-1} であるという条件の下で第 n 日目における気温が T_n，CDD の途中累積値が C_n である条件付き確率密度を $p_n(T_n, C_n|T_{n-1}, C_{n-1})$ とする．いま，もし $T_n < \bar{T}$ ならば，(7) より $C_n = C_{n-1}$ となる．一方，もし $T_n \geq \bar{T}$ ならば，(7) より $C_n = C_{n-1} + (T_n - \bar{T})$ となる．したがって，Dirac のデルタ関数 $\delta(x)$ を用いて書くと，

(i) $T_n < \bar{T}$ のとき

$$\begin{aligned} & p_n(T_n, C_n|T_{n-1}, C_{n-1}) \\ &= P_n(T_n|T_{n-1})\delta(C_n - C_{n-1}) \\ &= \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(T_n-\mu_n)^2}{2\sigma^2}\right\} \delta(C_n - C_{n-1}) \end{aligned} \tag{10}$$

(ii) $T_n \geq \bar{T}$ のとき

$$\begin{aligned} & p_n(T_n, C_n|T_{n-1}, C_{n-1}) \\ &= P_n(T_n|T_{n-1})\delta(C_n - (C_{n-1} + (T_n - \bar{T}))) \\ &= \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(T_n-\mu_n)^2}{2\sigma^2}\right\} \\ &\quad \times \delta(C_n - (C_{n-1} + (T_n - \bar{T}))) \end{aligned} \tag{11}$$

が成り立つ．

(8)，(10)，(11) より，第 n 日目において気温が T_n，CDD の途中累積値が C_n である確率密度 $p_n(T_n, C_n)$ は次のように計算できる．

(i) $T_n < \bar{T}$ のとき

$$\begin{aligned} & p_n(T_n, C_n) \\ &= \int_{-\infty}^{+\infty} dT_{n-1} \int_0^{+\infty} dC_{n-1} \\ &\quad p_n(T_n, C_n|T_{n-1}, C_{n-1}) \\ &\quad \times p_{n-1}(T_{n-1}, C_{n-1}) \\ &= \int_{-\infty}^{+\infty} dT_{n-1} \int_0^{+\infty} dC_{n-1} \\ &\quad \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(T_n-\mu_n)^2}{2\sigma^2}\right\} \\ &\quad \times \delta(C_n - C_{n-1})\, p_{n-1}(T_{n-1}, C_{n-1}) \\ &= \int_{-\infty}^{+\infty} dT_{n-1} \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(T_n-\mu_n)^2}{2\sigma^2}\right\} \\ &\quad \times p_{n-1}(T_{n-1}, C_n). \end{aligned} \tag{12}$$

(ii) $T_n \geq \bar{T}$ のとき

$$\begin{aligned} & p_n(T_n, C_n) \\ &= \int_{-\infty}^{+\infty} dT_{n-1} \int_0^{+\infty} dC_{n-1} \\ &\quad p_n(T_n, C_n|T_{n-1}, C_{n-1}) \\ &\quad \times p_{n-1}(T_{n-1}, C_{n-1}) \\ &= \int_{-\infty}^{+\infty} dT_{n-1} \int_0^{+\infty} dC_{n-1} \\ &\quad \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(T_n-\mu_n)^2}{2\sigma^2}\right\} \\ &\quad \times \delta(C_n - (C_{n-1} + (T_n - \bar{T}))) \\ &\quad \times p_{n-1}(T_{n-1}, C_{n-1}) \\ &= \int_{-\infty}^{+\infty} dT_{n-1} \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(T_n-\mu_n)^2}{2\sigma^2}\right\} \\ &\quad \times p_{n-1}(T_{n-1}, C_n - (T_n - \bar{T})) \end{aligned} \tag{13}$$

(12)，(13) は，確率密度関数 $p_{n-1}(T_{n-1}, C_{n-1})$ が与えられたときに $p_n(T_n, C_n)$ を求める漸化式とみなすことができる．漸化式の初期条件は第 1 日目の確率密度関数であり，これもデルタ関数を用いることによって次のように確率密度関数の形式で表すことができる．

(i) $T_1 < \bar{T}$ ならば

$$p_1(T, C) = \delta(T - T_1)\,\delta(C), \tag{14}$$

(ii) $T_1 \geq \bar{T}$ ならば

$$p_1(T, C) = \delta(T - T_1)\,\delta(C - (T_1 - \bar{T})). \tag{15}$$

初期条件 (14)，(15) から始めて，漸化式 (12)，(13) に従い，数値積分を用いて順次 $p_n(T_n, C_n)$ を計算していくことにより，最終的に $p_N(T_N, C_N)$ が求められる．補償金額 P_{call} の期待値は，$p_N(T_N, C_N)$ を用いて

$$\begin{aligned} & E[P_{call}] \\ &= \int_{-\infty}^{+\infty} dT_N \int_0^{+\infty} dC_N\, P_{call}\, p_N(T_N, C_N) \\ &= \int_{-\infty}^{+\infty} dT_N \int_0^{+\infty} dC_N \\ &\quad k \cdot \max(C_N - S, 0)\, p_N(T_N, C_N) \end{aligned} \tag{16}$$

と計算できる．

以上より，モンテカルロ法によらずに数値積分と漸

化式のみを用いて補償金額の期待値を求める方法が確立できた．

3.2 高速ガウス変換の適用による高速化

前節の方法で実際に計算を行うには，各時点 n において T_n 方向と C_n 方向にそれぞれ M 個の格子点を取り，この格子点での値を用いて積分 (12)，(13) を計算する．

いま，μ_n が T_{n-1} の関数として (9) で与えられることに注意すると，(12)，(13) の積分は，C_n を固定したとき，関数とガウス分布との畳み込み積分

$$g(y) = \int_{-\infty}^{+\infty} dx \, \exp\left\{-\frac{(y-x)^2}{2\sigma^2}\right\} f(x) \quad (17)$$

と見ることができる．ここで，上式の x，y に相当するのがそれぞれ (12) の T_{n-1}，T_n である．(13) についても同様である．いま，各時点での T_n は M 個の格子点で定義されているため，各 C_n に対するこの畳み込み積分を定義に従って計算すると，積分値を計算すべき点 T_n が M 個，1 個の積分のための標本点 T_{n-1} が M 個で，全部で $O(M^2)$ 回の演算が必要である．さらに，この畳み込み積分を M 個の C_n に対して計算する必要があるため，各時点での計算量は $O(M^3)$ となる．

この計算量を削減するため，高速ガウス変換[1)8)14)]と呼ばれる数値計算手法を適用することができる．高速ガウス変換は，もともと分子動力学における力の計算を高速化するために開発された手法であり，(17) におけるガウス分布 $\exp(-(y-x)^2/2\sigma^2)$ を Hermite 多項式を用いて展開した近似式を用いることにより，関数とガウス分布との畳み込み積分を $O(M)$ の計算量で行うことができる．高速ガウス変換の原理とアルゴリズムについては 1)8)14)，種々の金融デリバティブ価格計算への応用については，2)3) を参照されたい．

高速ガウス変換の適用により，各時点での計算量は $O(M^3)$ から $O(M^2)$ に削減でき，デリバティブの観測期間 N を固定して考えた場合，全計算量が $O(M^2)$ のアルゴリズムが構成できる．

3.3 数値積分上の工夫

上記のアルゴリズムでは，畳み込み積分 (12)，(13) を繰り返すことにより，各時点 n における確率密度関数 $p_n(T_n, C_n)$ を求めていく．しかし，(14)，(15) より明らかなように，確率密度関数 $p_1(T, C)$ は，$T = T_1$，$C = 0$ の位置（$T_1 < \bar{T}$ の場合）あるいは $T = T_1$，$C = T_1 - \bar{T}$ の位置（$T_1 \geq \bar{T}$ の場合）にデルタ関数型のピークを持つ．このピークは，漸化式を通じてその後の $p_n(T_n, C_n)$ にも伝播する．このようなピークを持つ被積分関数は，通常の数値積分公式では想定されていないため，(12)，(13) の積分を通常の数値積分公式を用いて行うと，計算精度が悪化する．

この問題を解決するため，本研究では，確率密度関数 $p_n(T_n, C_n)$ をデルタ関数型のピークの部分とそれ以外の部分とに分け，それぞれに対して別個に数値積分を行う．そのため，まず $p_n(T_n, C_n)$ が，

$$p_n(T_n, C_n) = \begin{cases} q_n(T_n, C_n) + r_n(T_n)\delta(C_n) \\ \quad (T_n < \bar{T}\text{のとき}) \\ q_n(T_n, C_n) + r_n(T_n) \\ \quad \times \delta(C_n - (T_n - \bar{T})) \\ \quad (T_n \geq \bar{T}\text{のとき}), \end{cases} \quad (18)$$

という形を持つと仮定する．$n = 1$ のときには，既に述べたように，この仮定は満たされている．

(18) を (12)，(13) に代入して整理すると，次のようになる．

(i) $T_n < \bar{T}$ のとき

$$\begin{aligned} & p_n(T_n, C_n) \\ &= \int_{-\infty}^{+\infty} dT_{n-1} \; P_n(T_n|T_{n-1}) \, p_{n-1}(T_{n-1}, C_n) \\ &= \int_{-\infty}^{\bar{T}} dT_{n-1} \; P_n(T_n|T_{n-1}) \\ &\qquad \times \{q_{n-1}(T_{n-1}, C_n) + r_{n-1}(T_{n-1}) \, \delta(C_n)\} \\ &\quad + \int_{\bar{T}}^{+\infty} dT_{n-1} \; P_n(T_n|T_{n-1}) \{q_{n-1}(T_{n-1}, C_n) \\ &\qquad + r_{n-1}(T_{n-1}) \, \delta(C_n - (T_{n-1} - \bar{T}))\} \\ &= \int_{-\infty}^{+\infty} dT_{n-1} \; P_n(T_n|T_{n-1}) \, q_{n-1}(T_{n-1}, C_n) \\ &\quad + \int_{-\infty}^{\bar{T}} dT_{n-1} \; P_n(T_n|T_{n-1}) \, r_{n-1}(T_{n-1}) \times \delta(C_n) \\ &\quad + P_n(T_n|C_n + \bar{T}) \, r_{n-1}(C_n + \bar{T}) \end{aligned} \quad (19)$$

(ii) $T_n \geq \bar{T}$ のとき

$$\begin{aligned} & p_n(T_n, C_n) \\ &= \int_{-\infty}^{+\infty} dT_{n-1} \; P_n(T_n|T_{n-1}) \\ &\qquad \times p_{n-1}(T_{n-1}, C_n - (T_n - \bar{T})) \\ &= \int_{-\infty}^{\bar{T}} dT_{n-1} \; P_n(T_n|T_{n-1}) \\ &\qquad \times \{q_{n-1}(T_{n-1}, C_n - (T_n - \bar{T})) \\ &\qquad + r_{n-1}(T_{n-1}) \, \delta(C_n - (T_n - \bar{T}))\} \\ &\quad + \int_{\bar{T}}^{+\infty} dT_{n-1} \; P_n(T_n|T_{n-1}) \\ &\qquad \times \{q_{n-1}(T_{n-1}, C_n - (T_n - \bar{T})) \\ &\qquad + r_{n-1}(T_{n-1}) \\ &\qquad \times \delta(C_n - (T_n - \bar{T}) - (T_{n-1} - \bar{T}))\} \end{aligned}$$

$$
\begin{aligned}
&= \int_{-\infty}^{+\infty} dT_{n-1}\ P_n(T_n|T_{n-1}) \\
&\qquad \times q_{n-1}(T_{n-1}, C_n - (T_n - \bar{T})) \\
&\quad + \int_{-\infty}^{\bar{T}} dT_{n-1}\ P_n(T_n|T_{n-1}) r_{n-1}(T_{n-1}) \\
&\qquad \times \delta(C_n - (T_n - \bar{T})) \\
&\quad + P_n(T_n|C_n + \bar{T} - (T_n - \bar{T})) \\
&\qquad \times r_{n-1}(C_n + \bar{T} - (T_n - \bar{T})) \qquad (20)
\end{aligned}
$$

(19)，(20) と (18) とを比較すると，$n-1$ のときに $p_{n-1}(T_{n-1}, C_{n-1})$ が (18) の形を持つならば，n のときにも $p_n(T_n, C_n)$ が (18) の形を持つことがわかる．(19) と (18)，(20) と (18) において，デルタ関数を含む項同士，含まない項同士をそれぞれ等しいと置くと，$q_n(T_n, C_n)$，$r_n(T_n)$ に対する漸化式が次のように求められる．

(i) $T_n < \bar{T}$ のとき

$$
\begin{aligned}
&q_n(T_n, C_n) \\
&= \int_{-\infty}^{+\infty} dT_{n-1}\ P_n(T_n|T_{n-1})\, q_{n-1}(T_{n-1}, C_n) \\
&\qquad + P_n(T_n|C_n + \bar{T})\, r_{n-1}(C_n + \bar{T}) \quad (21) \\
&r_n(T_n) \\
&= \int_{-\infty}^{\bar{T}} dT_{n-1}\ P_n(T_n|T_{n-1})\, r_{n-1}(T_{n-1}) \quad (22)
\end{aligned}
$$

(ii) $T_n \geq \bar{T}$ のとき

$$
\begin{aligned}
&q_n(T_n, C_n) \\
&= \int_{-\infty}^{+\infty} dT_{n-1}\ P_n(T_n|T_{n-1}) \\
&\qquad \times q_{n-1}(T_{n-1}, C_n - (T_n - \bar{T})) \\
&\qquad + P_n(T_n|C_n + \bar{T} - (T_n - \bar{T})) \\
&\qquad \times r_{n-1}(C_n + \bar{T} - (T_n - \bar{T})) \qquad (23) \\
&r_n(T_n) \\
&= \int_{-\infty}^{\bar{T}} dT_{n-1}\ P_n(T_n|T_{n-1})\, r_{n-1}(T_{n-1}) \quad (24)
\end{aligned}
$$

(21) 〜 (24) は，(18) のようにデルタ関数の成分を分離した後の関数に関する式であるので，被積分関数にはデルタ関数の成分を含まず，通常の数値積分公式によって積分が行える．そのため，漸化式としては (19)，(20) に代えて (21) 〜 (24) を用いる．

3.4 提案手法の計算量と精度

3.2 節で示したように，各時点 n における T_n，C_n 方向の格子点数がそれぞれ M であるとき，高速ガウス変換を用いたアルゴリズムの全計算量は $O(M^2)$ である．(19)，(20) に代えて (21) 〜 (24) を用いた場合は，$p_n(T_n, C_n)$ に関する漸化式が $q_n(T_n, C_n)$ および $r_n(T_n)$ に関する漸化式へと変わるが，後者もガウス分布と関数の畳み込み積分であるため，前者と同様に高速ガウス変換を適用することができ，計算量のオーダーは $O(M^2)$ とすることができる．したがって，計算時間 τ は $O(M^2)$ となる．一方，数値誤差については，式 (21) 〜 (24) の積分において標本点を等間隔に取って Simpson 則による数値積分を行った場合，標本点の数 M に対して誤差は $O(1/M^2)$ で減少することが知られている[12]．価格計算では漸化式に基づいて観測期間の日数（N）分だけ数値積分を繰り返す必要があるが，この場合でも N を固定して考えれば，全体での誤差は $E = O(1/M^2)$ となる．以上より，計算時間 $\tau = O(M^2)$ の関数として見ると，誤差 E は $1/\tau$ のオーダーで減少し，$E \cong c_2/\tau$（ただし c_2 は定数）となる．

これに対してモンテカルロ法の場合は，前章で述べたように，$E \cong c_1/\sqrt{\tau}$ である．したがって，ここで提案した手法は，モンテカルロ法と比べて，真の解への収束速度が向上していると言える．

ただし，以上はオーダーに関する議論であり，ある要求精度に対して実際に提案手法がモンテカルロ法より高速に解を求められるかどうかは，c_1，c_2 に依存する．そこで次章では，具体的な CDD デリバティブを例に取り，数値実験により両手法の計算時間と精度を比較する．

4. 提案手法の評価

4.1 計算時間と精度の評価

前章で提案したアルゴリズムを C 言語を用いて実装し，Pentium 4（2.2GHz）の PC 上で性能評価を行った．使用した OS は Windows XP Professional，コンパイラは Microsoft Visual C++ Ver. 6.0 である．

まず，提案手法の計算量と精度の関係を調べるため，各方向の格子点数 M を変えた場合の精度を評価する．例題としては，次の CDD デリバティブ（例題１）を用いた．

観測期間	：６月１日より３日間
観測地点	：東京
観測指標	：CDD（ただし $\bar{T}$=20 ℃）
行使値	：0（℃）
ティック値	：1（万円／℃）
種別	：コール

なお，開始日の気温は 20 ℃とし，Dischel モデルのパラメータとしては 10) に載っている値である $\beta = -0.56$，$\mu = -0.01$，$\sigma = 1.83$ を用いた．また，$\Theta_i = 20$ ℃とした．このとき，真の価格は 2.4233 万円（10^8 回のモンテカルロ法により計算）である．

本例題に対し，M を変えたときの計算時間と精度を表 1 に示す．表より，誤差は M の関数として明らかに $O(1/M^2)$ のオーダーで減少していることがわかる．一方，計算時間は，M が小さい領域では種々のオーバーヘッドのために $O(M^2)$ からずれるが，$M = 240$ 以降はほぼ $O(M^2)$ で増加していることがわかる．これより，3.4 節で述べた M，計算量，計算精度の関係が確認できた．

表 1 格子点数 M を変えたときの提案手法の計算時間と精度（例題 1）

M	計算価格（万円）	誤差（万円）	計算時間（秒）
60	2.4089	0.0144	0.10
120	2.4197	0.0036	0.28
240	2.4225	0.0008	0.38
480	2.4231	0.0002	1.02
960	2.4233	0.0000	3.13

4.2 モンテカルロ法との比較

次に，モンテカルロ法と計算時間・精度の比較を行うため，次の CDD デリバティブ（例題 2）を用いて計算を行った．

観測期間 ：7 月 7 日より N 日間
観測地点 ：東京
観測指標 ：CDD（ただし $\bar{T}$=24 ℃）
行使値 ：S（℃）
ティック値 ：1（万円／℃）
種別 ：コール

ここで，観測期間の長さ N は 5, 10, 20 の 3 通りに変化させる．また行使値 S は，このそれぞれに対して 10, 20, 40 の 3 通りに変化させる．日数 N が増えるにつれて CDD の累積値 C_N の期待値も大きくなるため，デリバティブが有効となるためのしきい値である S は N に応じて増加させるのが普通であるが，ここでは計算時間・精度に対する N と S の影響を独立に見るため，これら 2 つのパラメータを独立に変化させることとした．なお，開始日の気温は 24 ℃とし，Dischel モデルのパラメータは過去の気温データを用いて最小二乗法により $\beta = 0.7763$，$\mu = 0.0896$，$\sigma = 2.3734$ と定めた．また，事前の数値実験の結果から，T_n 方向の格子点数は C_n 方向の格子点数の半分程度でよいことがわかったため，ここでは，C_n 方向の格子点数を M，T_n 方向の格子点数を $M/2$ として計算量を節約した．

本例題に対し，M を変えたときの提案手法の計算時間と計算価格を，$N = 5$,10,20 の場合についてそれぞれ表 2，4，6 に示す．ここでは，計算価格に加え，第 N 日目における確率密度関数 $p_N(T_N, C_N)$ を T_N および C_N の全領域に渡って積分した量を全確率として表示した．これは，計算誤差がなければ当然 1 になるべき量である．また，シナリオ数 L を変えたときのモンテカルロ法の計算時間と計算価格を $N = 5$,10,20 の場合についてそれぞれ表 3，5，7 に示す．この計算価格に対しては，95%の信頼区間 $\pm 1.96\sigma^{se}$ を付加した．ここで，σ^{se} は次の式により計算される標準誤差である[13]．

$$\sigma^{se} = \sqrt{\frac{1}{L(L-1)}\left\{E[P_{call}^2] - (E[P_{call}]^2)\right\}} \quad (25)$$

また，(a) $N = 5$, $S = 10$ の場合，(b) $N = 10$, $S = 20$ の場合，(c) $N = 20$, $S = 40$ の場合の 3 つについて，両手法の収束の様子をそれぞれ図 1，2，3 に示す．

これら 3 つの場合について表 3，5，7 よりモンテカルロ法の精度と計算時間の関係を調べると，95%の信頼度で小数点以下 2 桁までの精度の価格を求めるには，(a)，(b)，(c) の場合にそれぞれ 10^7 回，10^8 回，10^8 回程度のシナリオ数が必要であり，数分から数十分の計算時間が必要であることが分かる．一方，表 2，4，6 より，提案手法では 1 分以下の計算時間で同じ精度の価格が求められている．この収束の速さは，図 1，2，3 からも明らかである．また，提案手法の精度と計算時間がパラメータ S に依存しないことも，表 2，4，6 より明らかである．以上より，$N = 5 \sim 20$ 程度の CDD デリバティブの価格を計算する問題において，提案手法はモンテカルロ法に比べ，大幅な速度向上を達成できると考えられる．

4.3 N を大きくした場合の収束性に関する考察

最後に，日数 N を更に大きくした場合のモンテカルロ法と提案手法の比較について，上記の実験結果を元に考察を行う．ただし，上記 (a)，(b)，(c) の場合と同様，行使値 S は日数 N に比例して増加させると仮定する．

まず，モンテカルロ法については，(a)，(b)，(c) の場合を比較することにより，シナリオ数 L を一定にした場合，標準誤差は N に比例して増加することが観察される．したがって，標準誤差を一定値に保つには，サンプル数を N^2 に比例して増加させることが必要となる．さらに，1 サンプルあたりの計算時間は明らかに N に比例するため，標準誤差を一定値に保って N を増加させる場合の計算時間は N^3 に比例すると考えられる．

一方，提案手法については，N を一定にした場合の計算量と精度の関係は 3.4 節で導いたが，N を変化さ

せた場合の理論的考察はより複雑である．そこで，ここでは表 2，4，6 を元に経験的な考察を行う．いま，表より計算価格の収束と全確率の収束とに高い相関があることに着目し，収束性の評価のために，全確率と 1 との差が 10^{-6} 程度になるという基準を用いることにする．すると，上記 (a)，(b)，(c) において，この基準はそれぞれ $M = 480, 1920, 3840$ の場合に達成され，計算時間はそれぞれ 2.45 秒，36.51 秒，450.73 秒と，N が 2 倍になるにつれて 12 〜 15 倍になることが観察される．

モンテカルロ法では N が 2 倍になったときの計算時間は 8 倍となるため，N をより大きくした場合には，モンテカルロ法と提案手法の差は徐々に縮まってゆくと予想される．この点を改良するには，現在均一に取っている (T_n, C_n) 空間での格子点に粗密をつけるなどの方法が考えられ，今後の課題である．

表 2 格子点数 M を変えたときの提案手法の計算時間と計算価格（例題 2，$N = 5$）

S	M	計算価格（万円）	計算時間（秒）	全確率
10	120	2.0320	0.55	0.997094
	240	2.0322	1.09	0.999916
	480	2.0323	2.45	0.999999
	960	2.0323	5.91	1.000000
	1920	2.0323	15.97	1.000000
20	120	0.4153	0.60	0.997094
	240	0.4152	1.09	0.999916
	480	0.4153	2.48	0.999999
	960	0.4153	5.94	1.000000
	1920	0.4153	15.98	1.000000
40	120	0.0041	0.55	0.997094
	240	0.0041	1.15	0.999916
	480	0.0041	2.50	0.999999
	960	0.0041	5.94	1.000000
	1920	0.0041	15.95	1.000000

表 3 シナリオ数 L を変えたときのモンテカルロ法の計算時間と計算価格（例題 2，$N = 5$）

S	L	計算価格（万円）	計算時間（秒）
10	10^5	2.0123 ±0.0288	0.32
	10^6	2.0276 ±0.0092	3.14
	10^7	2.0305 ±0.0028	31.05
	10^8	2.0322 ±0.0010	310.52
20	10^5	0.4098 ±0.0061	0.35
	10^6	0.4136 ±0.0020	3.18
	10^7	0.4150 ±0.0006	31.24
	10^8	0.4154 ±0.0002	313.65
40	10^5	0.0035 ±0.0005	0.37
	10^6	0.0041 ±0.0002	3.17
	10^7	0.0042 ±0.0001	31.20
	10^8	0.0042 ±0.0000	319.03

表 4 格子点数 M を変えたときの提案手法の計算時間と計算価格（例題 2，$N = 10$）

S	M	計算価格（万円）	計算時間（秒）	全確率
10	120	9.8257	1.25	0.996184
	240	9.8413	2.51	0.999851
	480	9.8423	5.60	0.999987
	960	9.8425	13.35	0.999996
	1920	9.8426	35.75	0.999998
20	120	5.0507	1.25	0.996184
	240	5.0578	2.47	0.999851
	480	5.0585	5.58	0.999987
	960	5.0586	13.62	0.999996
	1920	5.0587	36.51	0.999998
40	120	0.9850	1.30	0.996184
	240	0.9858	2.53	0.999851
	480	0.9859	5.61	0.999987
	960	0.9859	13.31	0.999996
	1920	0.9859	35.71	0.999998

表 5 シナリオ数 L を変えたときのモンテカルロ法の計算時間と計算価格（例題 2，$N = 10$）

S	L	計算価格（万円）	計算時間（秒）
10	10^5	9.7857 ±0.0411	0.59
	10^6	9.8236 ±0.0130	5.79
	10^7	9.8362 ±0.0041	58.07
	10^8	9.8430 ±0.0013	578.29
20	10^5	5.0709 ±0.0620	0.60
	10^6	5.0548 ±0.0196	5.80
	10^7	5.0582 ±0.0062	57.40
	10^8	5.0587 ±0.0020	576.21
40	10^5	0.9765 ±0.0134	0.64
	10^6	0.9848 ±0.0042	5.83
	10^7	0.9850 ±0.0013	57.84
	10^8	0.9862 ±0.0004	577.66

5. おわりに

本研究では，天候デリバティブの価格計算の高速化を目的として，従来のモンテカルロ法に代わる，高速ガウス変換を用いた新しいアルゴリズムを開発し，その評価を行った．本研究で明らかにした点は次の通りである．

(1) 提案手法では，モンテカルロ法と比べて計算した価格の真の価格への収束速度が向上する．すなわち，計算時間を τ とするとき，モンテカルロ法では $1/\sqrt{\tau}$ のオーダーでしか誤差が減少しないのに対し，提案手法では $1/\tau$ のオーダーで誤差が減少する．

(2) 提案手法を C 言語を用いて実装し，Pentium 4 PC 上で評価したところ，期間が 5 日から 20 日の CDD デリバティブの価格を計算する場合に，モ

表 6 格子点数 M を変えたときの提案手法の計算時間と計算価格（例題 2，$N = 20$）

S	M	計算価格（万円）	計算時間（秒）	全確率
10	120	35.0562	5.46	0.994479
	240	35.1993	10.82	0.999553
	480	35.2134	23.86	0.999918
	960	35.2164	56.72	0.999980
	1920	35.2172	152.12	0.999995
	3840	35.2174	461.58	0.999999
20	120	26.4290	5.45	0.994479
	240	26.5314	10.76	0.999553
	480	26.5424	27.08	0.999918
	960	26.5448	60.09	0.999980
	1920	26.5454	153.37	0.999995
	3840	26.5455	457.01	0.999999
40	120	13.3526	5.41	0.994479
	240	13.3922	10.80	0.999553
	480	13.4046	23.84	0.999918
	960	13.4058	57.16	0.999980
	1920	13.4061	151.78	0.999995
	3840	13.4062	450.73	0.999999

表 7 シナリオ数 L を変えたときのモンテカルロ法の計算時間と計算価格（例題 2，$N = 20$）

S	L	計算価格（万円）	計算時間（秒）
10	10^5	35.1280 ±0.0860	1.13
	10^6	35.2102 ±0.0272	11.11
	10^7	35.2062 ±0.0086	111.16
	10^8	35.2175 ±0.0027	1151.43
20	10^5	26.4690 ±0.0812	1.17
	10^6	26.5392 ±0.0257	11.10
	10^7	26.5345 ±0.0081	111.15
	10^8	26.5459 ±0.0026	1109.69
40	10^5	13.3585 ±0.1272	1.09
	10^6	13.4043 ±0.0402	11.02
	10^7	13.3983 ±0.0128	110.41
	10^8	13.4075 ±0.0040	1102.81

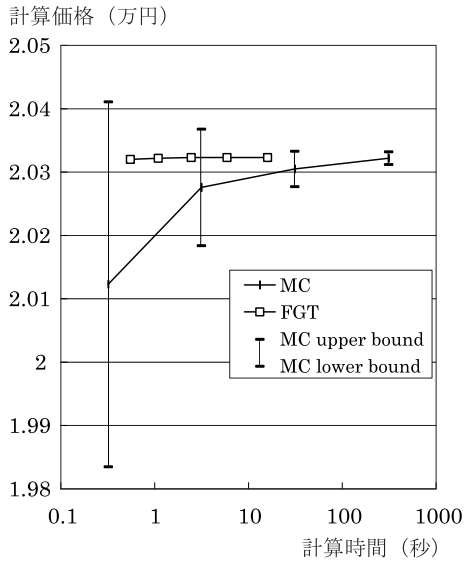

図 1 モンテカルロ法と提案手法の収束性（例題 2，$N = 5$，$S = 10$）

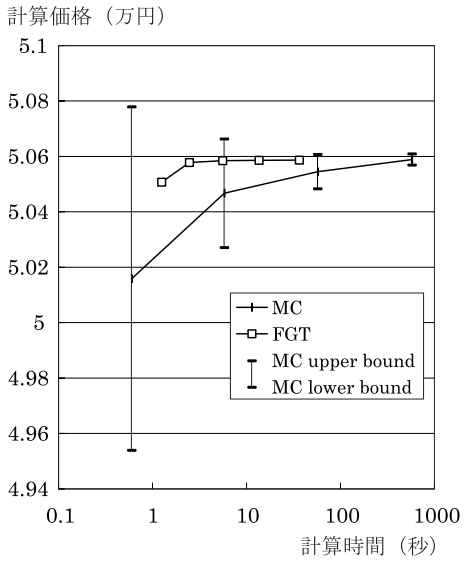

図 2 モンテカルロ法と提案手法の収束性（例題 2，$N = 10$，$S = 20$）

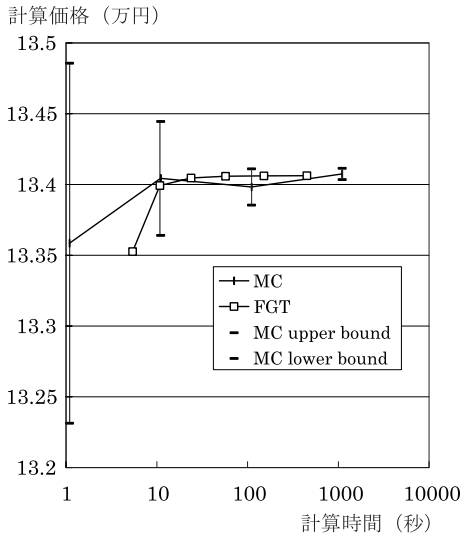

図 3 モンテカルロ法と提案手法の収束性（例題 2，$N = 20$，$S = 40$）

ンテカルロ法の 10 倍以上の高速化が達成できた．

今後は CDD デリバティブ以外の天候デリバティブに対しても本手法を拡張し，その有効性を検証するとともに，デリバティブの最適設計や時価評価に本手法を適用していきたい．

謝辞 本研究に対して有益な助言を下さった査読者の方々，及び（株）日立製作所本社の土方薫氏，ビジネスソリューション事業部の二木誠司主任技師，高橋俊博士，エンタープライズサーバ事業部の合田徳夫博士に感謝いたします．また日頃からご指導頂いている

名古屋大学大学院工学研究科の杉原正顯教授,（株）日立製作所中央研究所の濱中直樹部長，藤井啓明主任研究員，直野健研究員に感謝いたします.

参 考 文 献

1) Baxter, B. and Roussos, G.: A New Error Estimate of the Fast Gauss Transform, *SIAM Journal on Scientific Computing*, Vol. 24, No. 1, pp. 257-259 (2002).
2) Broadie, M. and Yamamoto, Y.: Application of the Fast Gauss Transform to Option Pricing, *Management Science*, Vol. 49, No. 8, pp. 1071-1088 (2003).
3) Broadie, M. and Yamamoto, Y.: A Double-Exponential Fast Gauss Transform Algorithm for Pricing Discrete Path-Dependent Options, conditionally accepted by *Operations Research*.
4) Cao, M. and Wei, J.: Pricing Weather Derivative: An Equilibrium Approach, *Department of Economics, Queen's University, Kingston, Ontario, Working Paper*, 1999.
5) http://www.climetrix.com
6) Dischel, B.: The D1 Stochastic Temperature Model for Valuing Weather Futures and Options, *Applied Derivatives Trading*, 1999.
7) http://www.fea.com/web/home
8) Greengard, L. and Strain, J.: The Fast Gauss Transform, *SIAM Journal on Scientific and Statistical Computing*, Vol. 12, No. 1, pp. 79-94 (1991).
9) 土方薫 編著: 天候デリバティブ, シグマベイズキャピタル, 2000.
10) 土方薫 著: 総論 天候デリバティブ - 天候リスクマネジメントのすべて, シグマベイズキャピタル, 2003.
11) http://www.hitachi.co.jp/New/cnews/030424a.html
12) Press, W., Flannery, B., Teukolsky, S. and Vetterling, W.: Numerical Recipes in FORTRAN, Cambridge University Press, 1992.
13) 津田孝夫: モンテカルロ法とシミュレーション, 三訂版，培風館, 1995.
14) Strain, J.: The Fast Gauss Transform with Variable Scales, *SIAM Journal on Scientific and Statistical Computing*, Vol. 12, No. 5, pp. 1131-1139, (1991).
15) 高橋俊, 土方薫, 恵木正史, 家島健司: 天候デリバティブの動向, 情報処理, Vol. 45, No. 1, pp. 34-41 (2004).
16) Zeng, L.: Pricing Weather Derivatives, *The Journal of Risk Finance*, Vol. 1, No. 3, pp. 72-78 (2000).

(平成？年？月？日受付)
(平成？年？月？日採録)

山本 有作（正会員）

1966 年生．1990 年東京大学工学部計数工学科（数理工学コース）卒業．1992 年同大学院工学系研究科物理工学専攻修士課程修了．同年 (株) 日立製作所中央研究所入所．2003 年名古屋大学大学院工学研究科 計算理工学専攻助手．現在，同講師．並列計算機向け行列計算アルゴリズムおよび金融工学向け高速計算アルゴリズムの研究開発に従事．高性能計算とその応用に興味を持つ．博士 (工学)．SIAM，INFORMS 各会員．

恵木 正史（正会員）

1971 年生．1994 年名古屋大学工学部応用物理学科卒業．1996 年同大学院理学研究科素粒子宇宙物理学博士前期課程修了．2000 年同大学院理学研究科素粒子宇宙物理学博士後期課程満了．2000 年 (株) 日立製作所中央研究所入所．以来、情報システムの性能評価理論、遺伝統計学、経済物理学の研究に従事．数理統計学・統計物理学の応用に興味を持つ．