# 3. モンテカルロ法

## (a) 一様乱数

(a-1) レーマー（Lehmer）合同乗積法

一様乱数の発生に関して最も一般的なのがレーマーの合同乗積法である。この方法はつぎの漸化式を用いて乱数を発生する

$$x_{i+1} = a x_i \bmod m$$

$a$,$m$ として適切な大きな正の整数を選ぶと数列 $x_0,\ x_1, \ldots$ は区間 $[1 : m-1]$ の整数値をとる一様乱数もどきとなる。コンピューターではどんな数列もなにかの規則をもって発生させるわけだから得られる数列は厳密には乱数ではなく擬似乱数（pseud-random numbers）である。

例えば、$a = 5^{11}$, $m = 2^{31}$ として初期値 $x_0$ を奇数にとると、この数列の周期は $2^{29} \sim 10^8$ となる。任意の区間の一様乱数を得るには上の乱数を適当な一次式で変換すれば良い。例えば区間 $[0 : 1]$ の一様乱数を欲しいのであれば、上の数列の $1/m$ の値を用いればよい。ただし、この方法は連続する数の間に強い相関があるという欠点があるので、その点では注意が必要である。

(a-2) トーズワース（Tausworthe）法

この方法は次の漸化式によって一様乱数を発生する

$$x_i = x_{i-p} \oplus x_{i-q}$$

ここで、$x$ は正の整数であり、演算子 $\oplus$ は $x_{i-p}$、$x_{i-q}$ のそれぞれを 2 進数で表したときの各桁（ビット）ごとの排他的論理和（exclusive-OR）をとることを意味する。例えば、$49 \oplus 18 = (11001)_2 \oplus (01010)_2 = (10011)_2 = 35$ である。$p$ と $q$ は乱数の周期が長くなるように選ばなければならない。例えば、$p, q$ として $(p, q) = (250, 130)$ とするとよい。この場合、周期は $2^{250} - 1 \sim 10^{75}$ となる。この方法の特徴は非常に長い周期の乱数を作れるところにあり、大規模なシミュレーションに適している。

## (b) 一様でない確率分布に従う乱数

一様乱数が発生できれば、これをもとに他の確率分布に従う乱数を作ることができる。

(b-1) 簡単な場合

ある一変数の確率分布の確率分布関数が $p(x)$ であるとする。次の関数 $P(x)$

$$P(x) = \int_{-\infty}^{x} p(t)\, dt$$

の逆関数 $P^{-1}(x)$ が得られれば、この分布に従う乱数は $x_1,\ x_2, \ldots$ を区間 $[0 : 1]$ の一様乱数だとすれば、$P^{-1}(x_1),\ P^{-1}(x_2), \ldots$ で得られる。この方法の難点は与えられた分布に対して $P^{-1}(x)$ を容易に求める方法がなければ現実的でないという点にある。

(b-2) フォンノイマン（von Neumann）の棄却法

確率分布関数 $p(x_1, x_2, \ldots)$ に対しこの分布の最大値を $P_{\max}$ とすれば、

1) 変数 $x_1, x_2, \ldots$ 各々を適当な区間の一様乱数で決める。

2) 1) で決めた座標 $(\xi_1, \xi_2, \ldots)$ に対し $p(\xi_1, \xi_2, \ldots)$ を求め、これを区間 $[0 : P_{\max}]$ の一様乱数 $\zeta$ と比較し、$p(\xi_1, \xi_2, \ldots) < \zeta$ であれば,$(\xi_1, \xi_2, \ldots)$ をこの分布に従う乱数として採択し、そうでなければ棄却する。

1),2) を繰り返すことによりこの分布に従う乱数が得られる。この方法の欠点は、1) で変数のとり得る範囲が無限大であっても、適当な有限区間に制限しなければならないこと、ごく限られた領域のみ確率密度が高いという様な分布に対しては 2) で棄却される割合が多くなって効率的でない、などである。

(b-3) メトロポリス（Metropolis）法

ある状態から他の状態へと次々とランダムに遷移を行っていく一種のランダムウォークを考える。これらの状態は変数 $\omega$ で一意的に指定できるものとし、各状態 $\omega$ に対し確率分布関数 $p(\omega)$ が与えられているとする。任意の二つの状態 $\omega_n$,$\omega_m$ に対し、状態 $\omega_n$ から状態 $\omega_m$ への遷移確率 $W(n \to m)$ 及びその逆の遷移確率 $W(m \to n)$ を次の詳細つりあいの条件が満たされるように決める。

$$p(\omega_n)W(n \to m) = p(\omega_m)W(m \to n) \tag{1}$$

さらに、この状態から状態へのランダムウォークではある任意の状態から他の任意の状態へ必ず到達できるというエルゴート性が満たされている必要がある。

これらの条件が満たされていれば、このランダムウォークの軌跡からなる状態の集合 $\{\omega_1, \omega_2, \ldots, \omega_n\}$ は分布 $p(\omega)$ に漸近的に近づいていく。(1) 式で $W(i \to j)$ は一意的には決まらないが、

$$W(i \to j) = \begin{cases} 1 & (p(\omega_i) < p(\omega_j)) \\ \frac{p(\omega_j)}{p(\omega_i)} & (p(\omega_i) > p(\omega_j)) \end{cases}$$

とすると (1) 式はみたされる。（メトロポリス法）

従って、この方法によるアルゴリズムは次のようになる。

1) i 番目の状態 $\omega_i$ から次の状態 $\omega_{i+1}$ の候補を乱数等を用いて決める。
2) $w = \frac{p(\omega_{i+1})}{p(\omega_i)}$ を求める。
3) $w >= 1$ であれば、次の状態として $\omega_{i+1}$ を無条件に採択する。
4) $w < 1$ であれば, 区間 $[0 : 1]$ の一様乱数を一つ作り,
   $w$ がこれより小さければ $\omega_{i+1}$ を次の状態として採択する。
   $w$ がこれより大きい場合は 1) で決めた $\omega_{i+1}$ を採択せず、$\omega_{i+1} = \omega_i$ とする。
5) 必要な回数に達するまで 1) から 4) をくり返し行う。

1) の状態遷移の方法であまりかけ離れた状態を次の候補すると多くの場合、採択率が低下して効率が悪い。一方、次の候補として近い状態を選ぶと、長いステップをへても特定の状態の近傍に踏み止まり、効率が悪いことがある。後者の様な場合は、他の離れた状態へジャンプできる様な遷移を時々挿入すれば問題を解消できるであろう。また、エルゴート性が満たされていれば、状態遷移の方法は必ずしもランダムである必要はない。