計算機実験II (L2) — モンテカルロと統計力学

藤堂眞治
wistaria@phys.s.u-tokyo.ac.jp

2019/10/18

1 多体系の統計力学
2 乱択アルゴリズム
3 物理過程のシミュレーション
4 擬似乱数
5 ヒストグラム
6 モンテカルロ積分
7 マルコフ連鎖モンテカルロ

典型的な統計力学モデル

- 古典粒子系
  - 調和振動子　$H = \frac{p^2}{2m} + \frac{k}{2}x^2$
  - 多粒子系

$$H = \sum \frac{p_i^2}{2m} + \sum_{ij} V(x_i, x_j)$$

  - バネビーズ模型

$$H = \sum \frac{p_i^2}{2m} + \frac{k}{2}\sum_{ij}(x_i - x_j)^2$$

- 磁性体
  - イジング模型　$H = -J\sum_{ij}\sigma_i\sigma_j \quad \sigma_i = \pm 1$

多体系の統計力学

- カノニカル分布　$P(c) = \exp[-\beta H(c)]/Z$　　$(\beta = k_{\mathrm{B}} T)$
- 分配関数・自由エネルギー

$$Z(T) = \int \exp[-\beta H(p,x)]\, dp\, dx \qquad \text{(粒子系)}$$

$$= \sum_{c} \exp[-\beta H(c)] \qquad \text{(イジング模型)}$$

$$f(T) = -\beta^{-1} \log Z(T)$$

- 物理量の期待値

$$\langle A \rangle = Z^{-1} \int A(p,x) \exp[-\beta H(p,x)]\, dp\, dx \qquad \text{(粒子系)}$$

$$= Z^{-1} \sum_{c} A(c) \exp[-\beta H(c)] \qquad \text{(イジング模型)}$$

多体系の統計力学

- 内部エネルギー

$$E = -\frac{\partial}{\partial \beta} \log Z = Z^{-1} \sum_c H(c) \exp[-\beta H(c)]$$

- 比熱

$$C = \frac{1}{N} \frac{\partial E}{\partial T} = \frac{\beta^2}{N} (\langle H^2 \rangle - \langle H \rangle^2)$$

代表的な数値計算手法

- 数え上げ
  - 計算コスト × (指数関数的)
  - メモリコスト ○ ($\mathcal{O}(1)$)
- 転送行列法
  - 計算コスト △ (指数関数的)
  - メモリコスト △ (指数関数的)
- 分子動力学法
  - 計算コスト ○ ($\mathcal{O}(N)$)
  - メモリコスト ○ ($\mathcal{O}(N)$)
  - 統計誤差あり
- マルコフ連鎖モンテカルロ法
  - 計算コスト ○ ($\mathcal{O}(N)$)
  - メモリコスト ○ ($\mathcal{O}(N)$)
  - 統計誤差あり

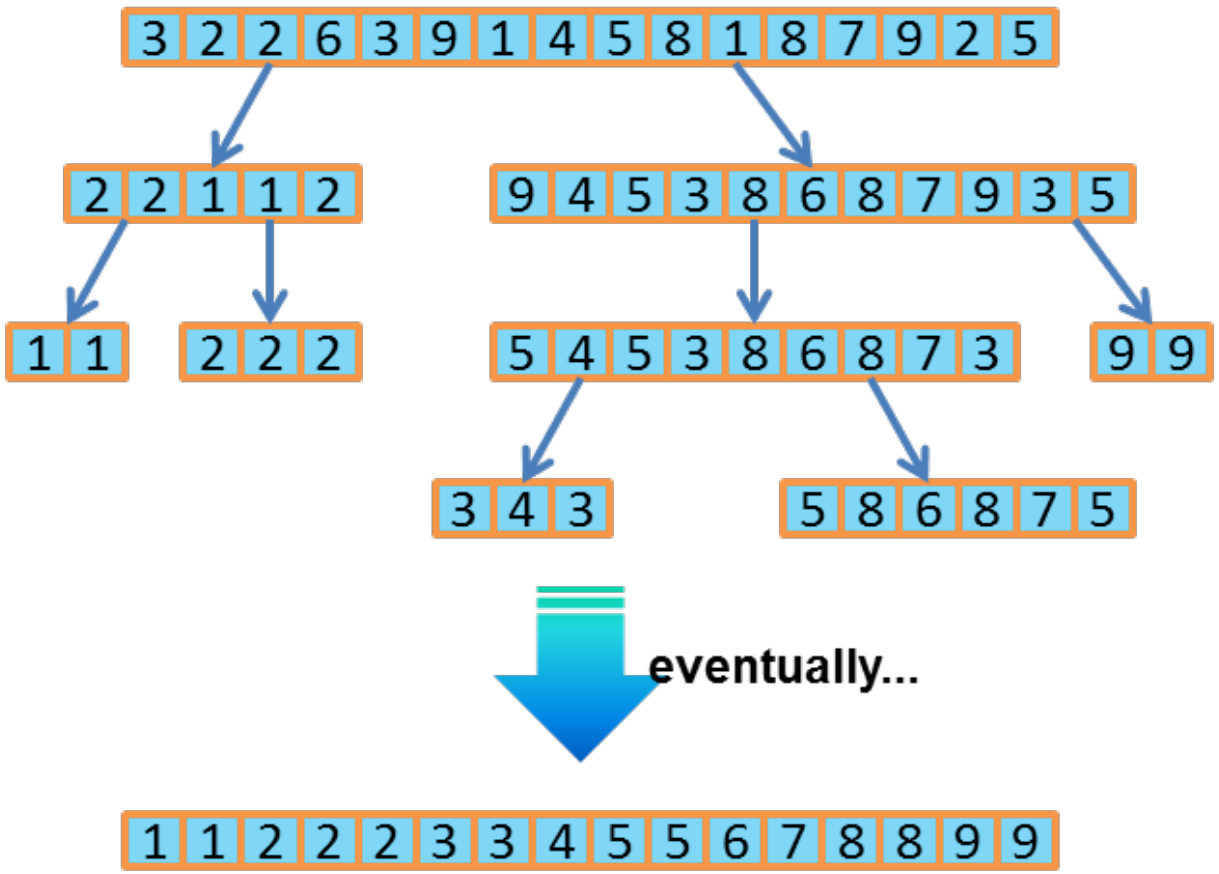
乱択アルゴリズム (randomized algorithm)

- 実行中に乱数を参照してその値によって振る舞いをかえるアルゴリズム
- ラスベガスアルゴリズム
  - 乱数の出方によらず常に正しい結果を与えるアルゴリズム
  - 平均化効果を利用するアルゴリズム：クイックソート
- モンテカルロアルゴリズム
  - 乱数の出方によっては誤った結果を与えるアルゴリズム
  - 標本を利用するアルゴリズム：最大カット問題
  - くじ引き型のアルゴリズム：素数性判定、関数の同一性、行列積の検算
  - サンプリングアルゴリズム：モンテカルロ積分、マルコフ連鎖モンテカルロ

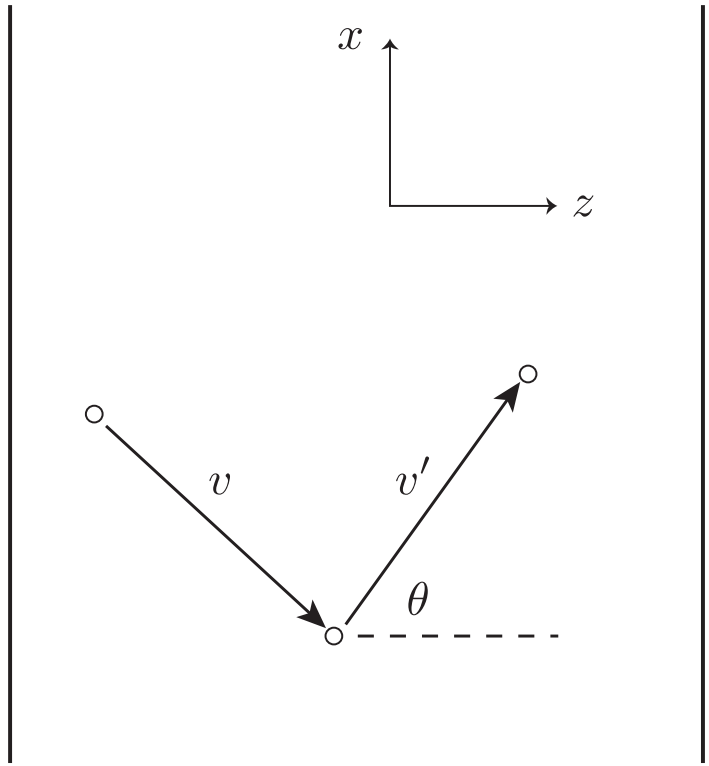
乱択クイックソート

- 分割統治法に基づく再帰的なソートアルゴリズム
  - 配列の中から要素を一つ選び、それより小さい要素からのみなる集合と大きい要素のみからなる集合の 2 つに分ける
  - それぞれの集合をソートし、結合する
- ほぼ同じ大きさの集合に分けることができる場合の実行ステップ数 $\sim \mathcal{O}(n \log n)$
- 最悪 (選んだ要素が常に最大 or 最小値) の場合のステップ数 $\sim \mathcal{O}(n^2)$
- 分割に用いる要素を「ランダムに」選ぶ $\Rightarrow$ 平均ステップ数 $\sim \mathcal{O}(n \log n)$

クイックソート


3 2 2 6 3 9 1 4 5 8 1 8 7 9 2 5
2 2 1 1 2
9 4 5 3 8 6 8 7 9 3 5
1 1
2 2 2
5 4 5 3 8 6 8 7 3
9 9
3 4 3
5 8 6 8 7 5
eventually...
1 1 2 2 2 3 3 4 5 5 6 7 8 8 9 9

物質中の中性子輸送



物質中の中性子輸送

- ある板状の物質 (厚さ D) に垂直に中性子が入射したときの吸収率・透過率・反射率
- 中性子はある確率で物質の原子核に衝突し、確率 p_{c} で吸収、$p_{\mathrm{s}} = 1 - p_{\mathrm{c}}$ で散乱される
  - 衝突は確率的に起きるので、次の衝突までの距離 ℓ は指数分布にしたがう (ℓ: 平均自由行程)

$$p(\ell) = \lambda e^{-\lambda \ell}$$

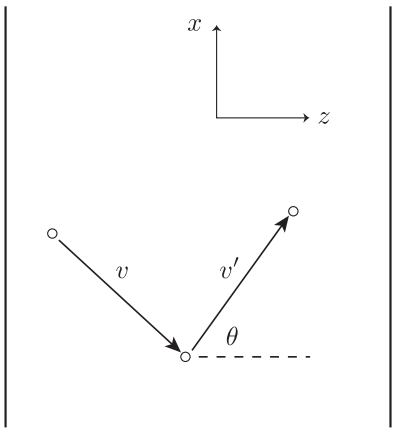
  - 衝突時、ランダムな方向に散乱されるとすると

$$p(\theta, \phi)\, d\theta\, d\phi = \frac{d\Omega}{4\pi} = \frac{\sin\theta}{4\pi}\, d\theta\, d\phi$$

$$p(\theta) = \frac{\sin\theta}{2} \qquad p(\phi) = \frac{1}{2\pi}$$

モンテカルロシミュレーション

1. 初期条件 $z = 0,\ \theta = 0$
2. 指数分布から ℓ を選ぶ
3. $z \leftarrow z + \ell \cos\theta$
4. $z < 0 \rightarrow$ 反射 (終了)
   $z > D \rightarrow$ 透過 (終了)
   $0 < z < D \rightarrow$ 確率 p_{c} で吸収 (終了)、p_{s} で散乱
5. 散乱後の θ を選び、2 に戻る



乱数

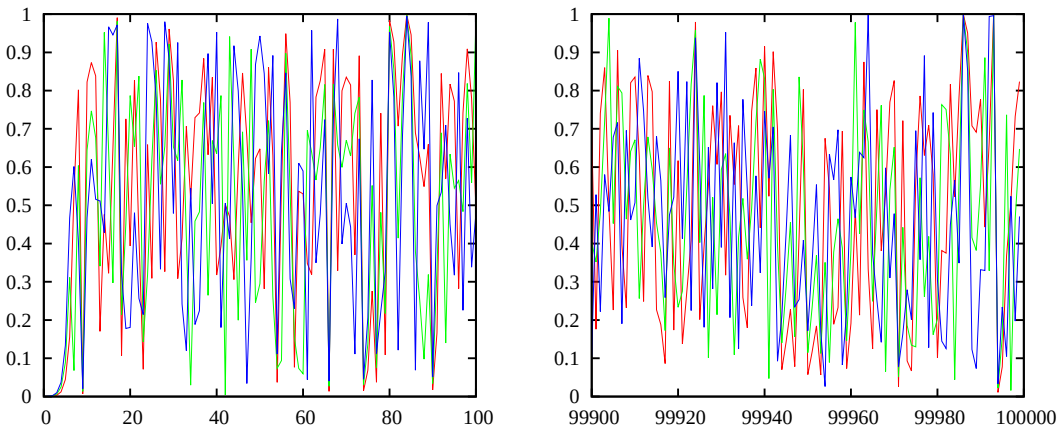
- 自然乱数 (ハードウェア乱数)
  - さいころ, コイン, ルーレット, 核分裂反応, 熱雑音, ショット雑音 ...
- モンテカルロシミュレーションにおける必要条件
  - 多数の乱数が必要
  - ポータビリティ
  - 生成速度
  - 再現性
- 擬似乱数 (pseudo random number)
  - 計算機でプログラムに従って生成
  - 分布の一様性, 相関, 周期に注意する必要あり

擬似乱数発生器

- 最も簡単な乱数発生器：線形合同法 (linear congruential method)

$$x_{n+1} = (ax_n + c) \bmod m$$

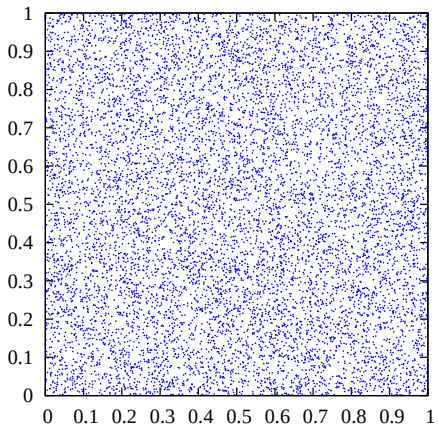
- 例) $a = 65539$, $c = 0$, $m = 2147483648$ (周期 $m - 1$)
- 少しだけ異なる初期値 ($x0 = 1, 2, 3$) から始めた場合

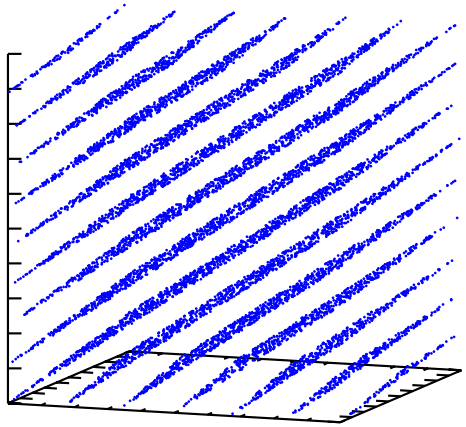


- 数十ステップ進むとバラバラな振舞い ⇒ カオス的

擬似乱数生成器における相関

- 合同乗算法で多次元超立方体中に「ランダムに」点を打つと、それらの点は全て比較的小数の等間隔に並んだ超平面の上にのってしまう (多次元疎結晶構造)





- 計算式に従って生成するため、必ず何らかの相関は残る
- できる限り相関が少なく周期の長い、理想的な乱数の開発が続けられている
  - 現時点で、標準的な乱数発生器：メルセンヌ・ツイスター
  - 周期 $2^{19937} - 1$、高速、日本製! (例: random.c)

乱数発生器の選び方

- 万能乱数発生器は存在しない
- 生成された乱数のもつ性質について, 様々な数学的に厳密な証明, 多くのテスト結果がすでに存在するが, 特定のシミュレーションに使った場合の結果については何も保証してくれない
- 自分で乱数発生器を「発明」してはいけない
- 自分で乱数発生器をプログラムしてはいけない (既存のライブラリを使う)
- 初期化 (種の設定) を正しく行う
- 実際にそれらしい乱数が生成されているか, 目でみて確認する
- 二種類以上の乱数発生器を使ってみて, 互いに一致する結果が出るかどうか確認する

様々な分布

- 乱数発生器は通常、一様な整数乱数あるいは実数乱数を生成
- 一様分布以外の分布にしたがう乱数の発生方法の代表例
- 逆関数法
  - 確率分布関数 $F(x)$ の逆関数 $F^{-1}(y)$ と (0,1) の一様乱数 u から $v = F^{-1}(u)$
  - 例: 指数分布 $p(x) = \frac{1}{\mu} e^{-x/\mu}$
    $F(x) = 1 - e^{-x/\mu}$　　$F^{-1}(y) = -\mu \log(1-y)$
  - 一般の確率分布関数について逆関数を求めるのは困難
- 棄却法
  - 確率密度関数を完全に囲むような箱を用意し、その箱の中で一様乱数を生成
  - 確率密度関数の下側の点が生成されたら、その x 座標を乱数として採用。上側の点の場合には再度生成
  - もとの確率密度関数よりも箱が大きくなりすぎると非効率

ヒストグラムの作り方

- 連続変数 (実数) のデータの場合 ([] 内はサンプルプログラムでの変数名)
  - N: サンプル数 [samples]
  - $x_{\min}$: データの最小値 (カットオフ) [xmin]
  - $x_{\max}$: データの最大値 (カットオフ) [xmax]
  - n: ビンの個数 [bins]
  - Δ: ビンの幅 ($\Delta = (x_{\max} - x_{\min})/n$) [dx]
- サイズ n の配列を準備
  - データ毎にどのビンに入るか計算: $j = (x - x_{\min})/\Delta$
  - (必要に応じて) $0 \leq j < n$ であることを確認 (範囲外のデータは無視する)
  - 配列の j 番目の値を 1 増やす
- サンプルプログラム: histogram.c

ビンの個数の設定

- 最適の幅というものはない
- 個数を増やすと表現の自由度は増えるが、各ビンのエラーバーが大きくなる
  - データが統計的に独立である場合、それぞれのビンのカウント数 m はポワソン分布に従う
  - 統計誤差 $\sim \sqrt{m}$
- いくつかの方法・公式が提案されているが、分布の形によっては不適切な場合も
  - スタージェスの公式 $n = \log_2 N + 1$
  - スコットの公式 $\Delta = 3.5\sigma/N^{1/3}$
- 実際には、ビンの個数を何通りか試してみるのが良い
- データを取り直すことが出来ない and/or コストがかかる場合も多いので、生データはいったんファイルに保存しておく

モンテカルロ積分

- 円周率を与える公式

$$\pi = \lim_{c\to\infty} \int_0^c f(x)\,dx \quad f(x) = \frac{2}{\cosh x}$$

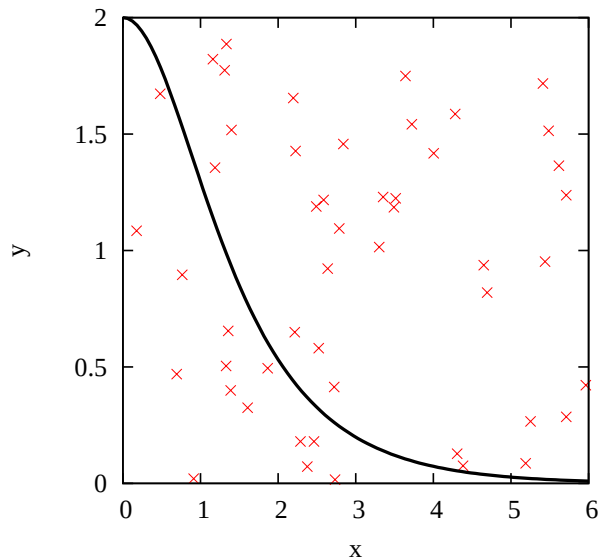
- スタンダードな数値積分法: 台形公式 (一次式補間), シンプソン公式 (二次式補間), etc
- カットオフ c の値
  - 誤差は c が大きくなると指数関数的に小さくなる
  - 例えば $c = 20$ で誤差は 8.3×10^{-9} 以下

単純サンプリング

- $[0, c]$ と $[0, 2]$ の一様分布から二次元上の点 (x, y) を M 組生成
- $f(x)$ の下に入った数 N をカウント

$$\pi \simeq 2c \times \frac{N}{M}$$

M	平均値	誤差
100	4.8	1.3
10000	3.12	0.11
1000000	3.154	0.011



統計誤差の評価

- このモンテカルロ積分が実際に評価している積分

$$\frac{1}{2c}\int_0^c\int_0^2 \theta(x,y)\,dx\,dy \quad \theta(x,y)=\begin{cases}2c & \text{if } y<f(x)\\ 0 & \text{otherwise}\end{cases}$$

- 統計誤差の評価
  - 試行の成功確率 (success probability): $q=\frac{\pi}{2c}$
  - 一回の試行の平均値 (mean): $\mu = 2c\times q=\pi$
  - 分散 (variance):

$$s^2=(2c)^2q+0^2(1-q)-\mu^2=2c\pi-\pi^2=4c^2q(1-q)$$

  - $c=20$ の時:

$$q\simeq 0.0785 \quad s^2\simeq 116$$

中心極限定理 (central limiting theorem)

- M 回の試行のうち N 回成功する確率 (π の見積もり値が $m = 2cN/M$ となる確率)

$$p(m = 2c\frac{N}{M}) = \frac{M!}{N!(M-N)!}q^N(1-q)^{M-N}$$

- 両辺の対数をとってスターリングの公式を使う

$$\log p(m) \simeq \frac{M}{2c}(m \log \frac{\pi}{m} + (2c-m)\log \frac{2c-\pi}{2c-m})$$

- m に関して平均値 π の周りで二次まで展開

$$\log p(m) \simeq -\frac{M}{2s^2}(m-\pi)^2$$

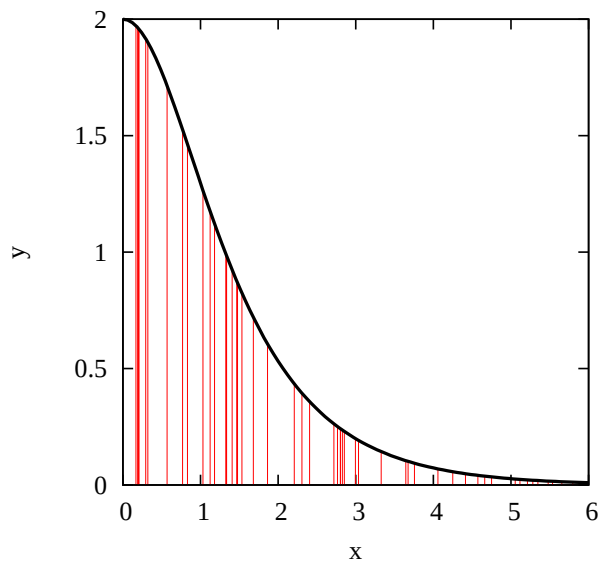
- 分散 s^2/M の正規分布 (中心極限定理)
- 統計誤差は $\sqrt{M}$ に反比例して減少 $\Rightarrow$ 1 桁小さくするには 100 倍の計算が必要

単純サンプリング (2)

- y に関してあらかじめ積分
- $[0, c]$ の一様乱数 x を用いて

$$\int_0^c \frac{f(x)}{p(x)} p(x)\, dx \simeq \frac{1}{M} \sum_i c f(x_i) \quad p(x) = \frac{1}{c}$$

M	平均値	誤差
100	3.1	0.8
10000	3.00	0.08
1000000	3.147	0.008



誤差の評価

- 関数 $f(x)/p(x)$ の分散

$$s^2 = \int_0^c \left(\frac{f(x)}{p(x)}\right)^2 p(x)\,dx - \pi^2 = c\int_0^\infty f^2(x)\,dx - \pi^2 = 4c - \pi^2$$

- $c = 20$ のとき $s^2 \simeq 70.1$
- 同じ試行回数 M の時, 誤差は $\sqrt{70.1/116} = 0.77$ 倍
- もしくは M を $116/70.1 = 1.65$ 倍したのと同じ効果
- 積分次元は低ければ低いほど良い

次元の呪い (curse of dimensionality)

- n 次元超立方体 (1 辺の長さ 2, 体積 2^n) に対する n 次元単位球の体積の割合

$$q = \frac{\pi^{n/2}/\Gamma(\frac{n}{2}+1)}{2^n} \sim (\pi/n)^{n/2}$$

  $n = 10$ で 0.2%, $n = 20$ で 10^{-8}, $n = 100$ で 10^{-70}
- モンテカルロ積分で球の体積を計算しようとすると, 標準偏差に対する平均値の割合は指数関数的に小さい

$$\frac{q}{\sqrt{q(1-q)}} \sim \sqrt{q}$$

- 次元が高くなるにつれて指数関数的に大きな M が必要となる
- c.f. 通常の数値積分 (台形公式等) でも同様

重点的サンプリング

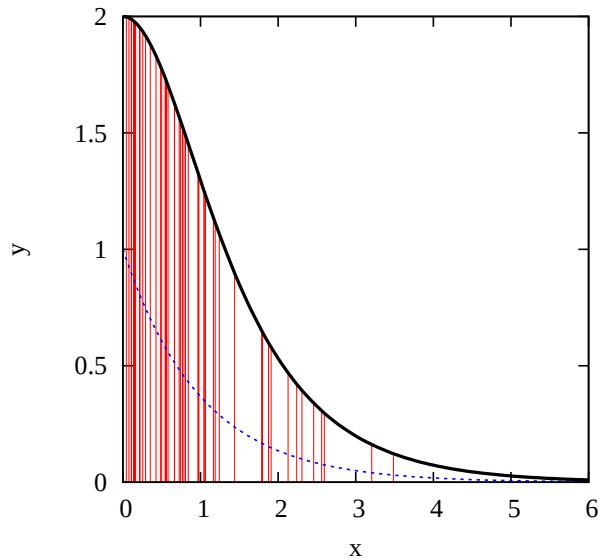
- (平均値が同じなら) 被積分関数の分散が小さければ小さいほど良い (= 統計誤差が小さい)
- サンプリングの分布 $p(x)$ の形が $f(x)$ に近い程良い
- $f(x)$ の値が大きい所はより頻繁にサンプリング
- 重点的サンプリング (importance sampling)

重点的サンプリング

- 積分への寄与が大きな箇所をより重点的にサンプリング

$$p(x) = e^{-x}$$

M	平均値	誤差
100	3.06	0.06
10000	3.142	0.006
1000000	3.1412	0.0006

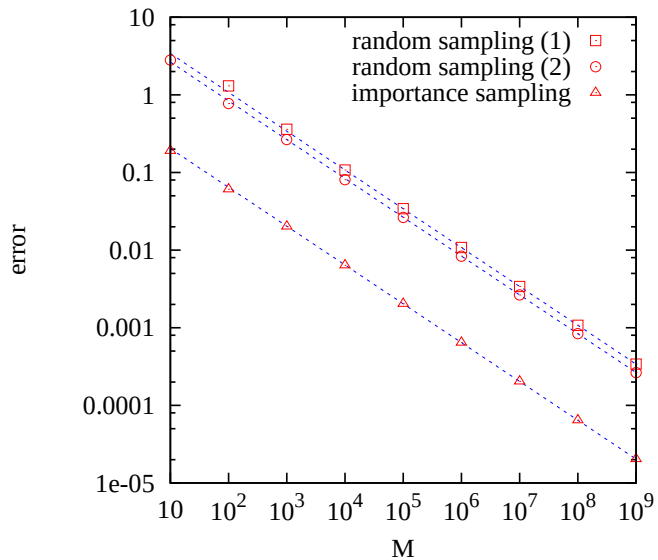


誤差のサンプル数依存性

- 関数 $f(x)/p(x)$ の分散

$$s^2 = \int_0^c \left(\frac{f(x)}{p(x)}\right)^2 p(x)\,dx - \pi^2 \simeq 2(2+\pi) - \pi^2 = 0.414$$

- 同じ試行回数 M の時, 誤差は $\sqrt{0.414/116} = 0.06$ 倍
- もしくは M を 280 倍したのと同じ



理想的な重点的サンプリング?

- 理想的には $p(x)$ を $f(x)$ に比例するように取れば良い
- このとき $f(x)/p(x)$ は定数 (分散 0) → 1 回のサンプリングで厳密な結果が得られる???
- 実際には $p(x)$ が確率密度となるように規格化条件から定数 c を決めておく必要あり

$$\int p(x)\,dx = c\int f(x)\,dx = 1$$

- c は今欲しい答そのもの!

統計物理における平衡状態

- Boltzmann 分布 ($\beta \equiv 1/k_B T$)

$$\pi(s) = \exp[-\beta \mathcal{H}(s)] \Big/ \sum_s \exp[-\beta \mathcal{H}(s)]$$

- 物理量の期待値

$$\langle A \rangle = \sum_s A(s) \exp[-\beta \mathcal{H}(s)] \Big/ \sum_s \exp[-\beta \mathcal{H}(s)]$$

- $\sum_s$ は全ての状態に関する和 (系の体積に対して指数関数的に増加)

- 全ての状態について和をとるかわりに、Boltzmann 重みが大きい ($=\mathcal{H}$ が小さい) ところだけをモンテカルロ・サンプリング

マルコフ連鎖モンテカルロ

- 直接 Boltzmann 分布から独立したサンプリングをおこなうことは難しい
- 直前の状態からある確率にしたがって、次の状態を生成 (マルコフ連鎖、Markov chain)

$$\begin{aligned}\Pr(X_{n+1} = s_j \,|\, X_0 = s_{i_0}, X_1 = s_{i_1}, \cdots, X_n = s_i) \\ = \Pr(X_{n+1} = s_j \,|\, X_n = s_i) = P_{i,j}\end{aligned}$$

- 長時間極限で Boltzmann 分布が達成されるように遷移確率 (行列)$P_{i,j}$ を選ぶ

$$\lim_{n\to\infty} \Pr(X_n = s_j) \sim \pi_j = \exp[-\beta \mathcal{H}(s_j)]$$

遷移行列が満たすべき条件

- 確率であるための条件: $0 \leq P_{i,j} \leq 1$
- 確率保存: $\sum_j P_{i,j} = 1$
- エルゴード性 (ergodicity):
  ある整数 M が存在し、$n \geq M$ の全ての n で $(P^n)_{i,j} > 0$
- つりあいの条件 (balance condition):

$$\sum_{i=1}^{k} \pi_i P_{i,j} = \pi_j$$

Boltzmann 分布が固有値 1 の左固有ベクトル

Perron-Frobenius の定理

- 正の正方行列 A(すべての要素が正) について以下が成り立つ
  - 他の全ての固有値よりも絶対値の大きな正の固有値 r が存在する
  - 固有値 r は単純固有値である (縮退していない)
  - 固有値 r に対する右 (左) 固有ベクトル v (w) は正のベクトルである
  - 固有値 r は $\min_i \sum_j a_{ij} \leq r \leq \max_i \sum_j a_{ij}$ を満たす
- A が零の要素を持つ場合でも A が原始的 (primitive = エルゴード的) である限り、上の結果は成り立つ
- 遷移行列は上の条件を満たす
  - Boltzmann 分布は絶対値最大の固有ベクトル
  - 遷移行列を掛けていくと Boltzmann 分布に収束

詳細つりあいの条件

- 実際には「つりあいの条件」よりもさらに厳しい「詳細つりあいの条件 (detailed balance condition)」を考えることが多い

$$\pi_i P_{i,j} = \pi_j P_{j,i}$$

- 両辺を i について和をとると「つりあいの条件」に帰着する
- 「詳細つりあいの条件」は「つりあいの条件」の十分条件

調和ポテンシャル中の古典粒子

- ポテンシャルエネルギー

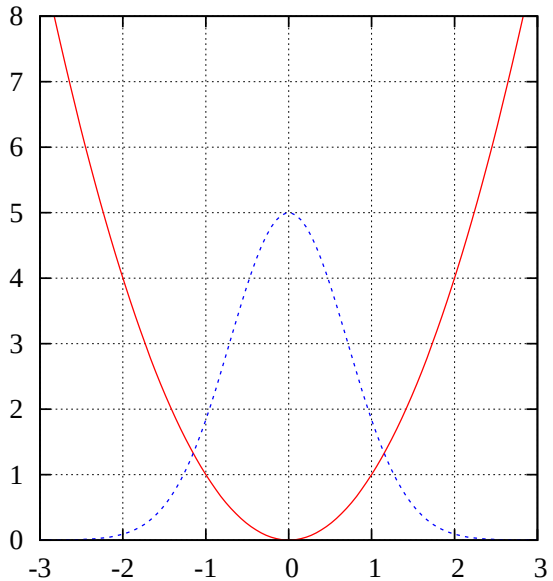
$$V(x) = x^2$$

- Bolotzmann 分布

$$P(x) = \frac{e^{-\beta V(x)}}{\int e^{-\beta V(x)}\, dx}$$

- 物理量の期待値

$$\langle x^2 \rangle = \frac{\int x^2 e^{-\beta V(x)}\, dx}{\int e^{-\beta V(x)}\, dx}$$



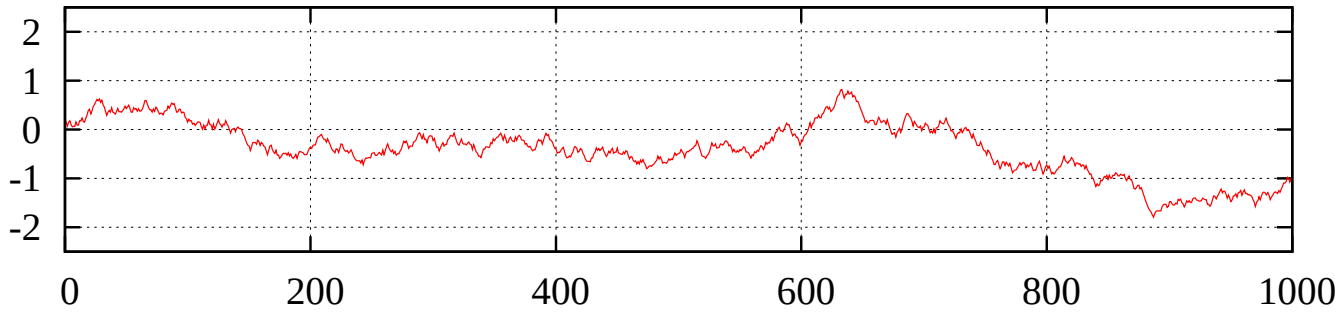
- 逆温度 β が大きいと被積分関数の分散が非常に大きい $\Rightarrow$ 重点的サンプリング

Metropolis 法

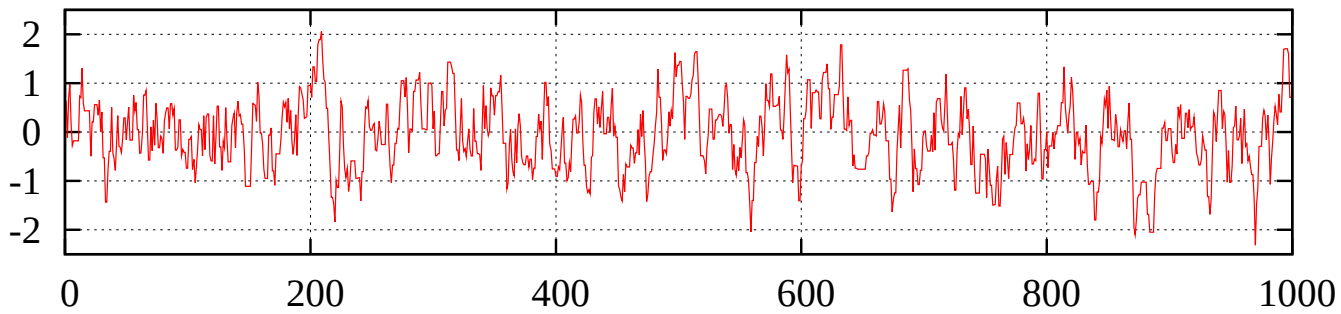
- 現在の配位 x から、試行配位 (trial configuration) x' を $x - \Delta \sim x + \Delta$ の一様分布より選ぶ
- 確率 $\min\left(1, \frac{e^{-\beta V(x')}}{e^{-\beta V(x)}}\right)$ で x' を採択 (accept)。棄却 (reject) された場合にはもとの x のまま
- 物理量の測定 (reject された場合にもカウントする)
- 採択確率 (acceptance probability) は、$\frac{e^{-\beta V(x')}}{e^{-\beta V(x)} + e^{-\beta V(x')}}$ でもよい
- 例: harmonic.c

Metropolis 法によるシミュレーション

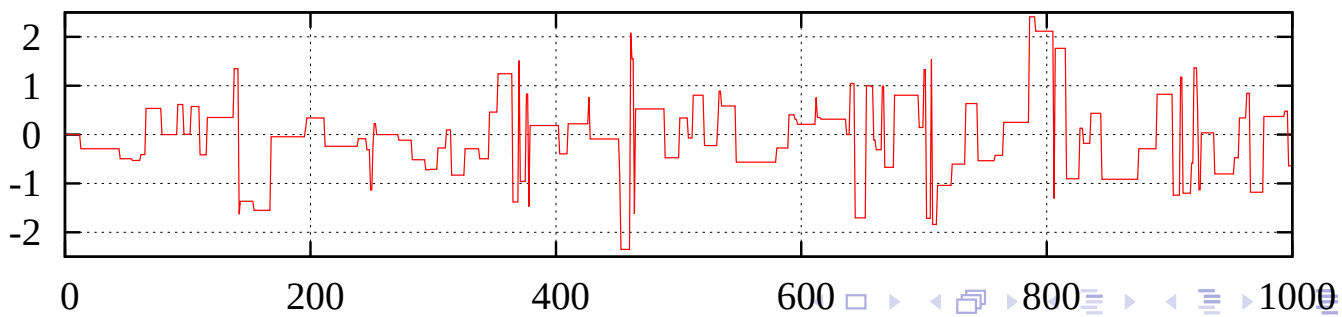
- $\Delta = 0.1$



- $\Delta = 1$



- $\Delta = 10$



自己相関関数 (autocorrelation function)

- エルゴード性 + つりあい条件 ⇒ 原理的に正しいマルコフ連鎖モンテカルロ
- 実際には自己相関を考慮する必要あり
- 自己相関関数

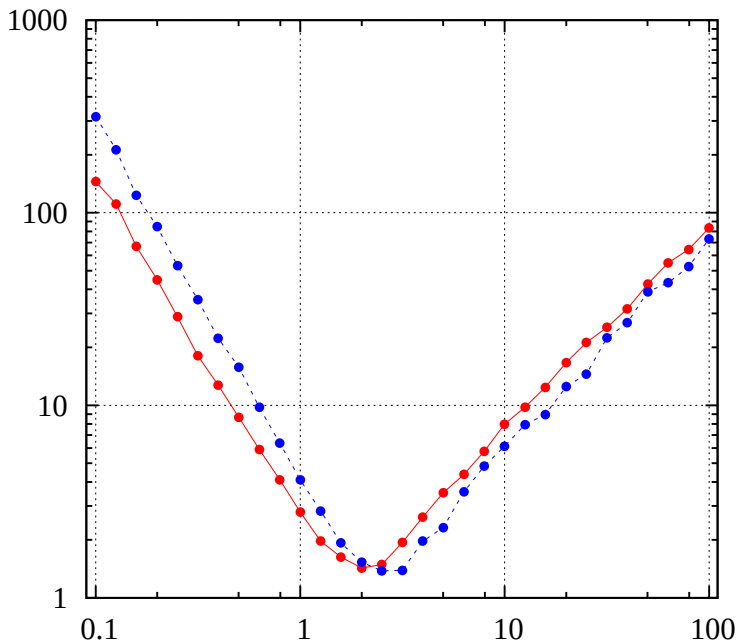
$$C(t) = \frac{\langle A_{i+t} A_i \rangle - \langle A \rangle^2}{\langle A^2 \rangle - \langle A \rangle^2} \sim \exp(-\frac{t}{\tau})$$

- τ: 自己相関時間 (autocorrelation time)
- 自己相関の影響により、統計的な「有効サンプル数」が減少

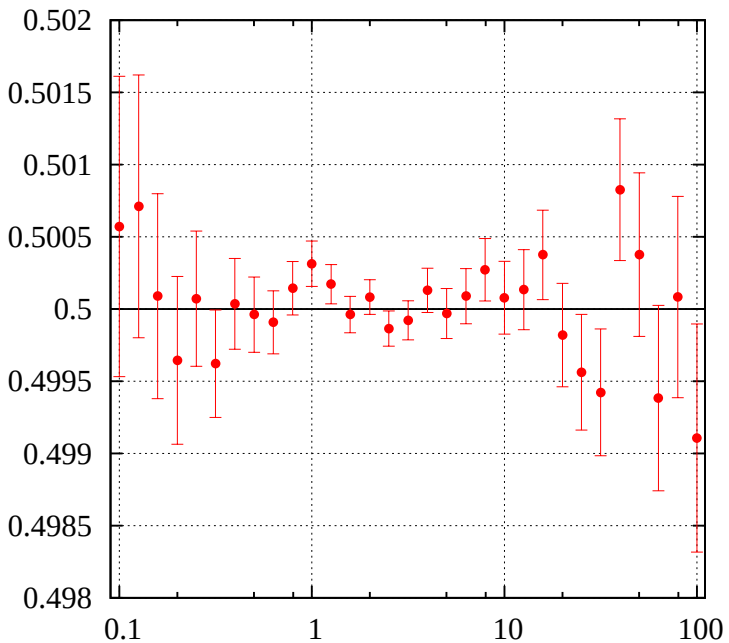
$$M \to \frac{M}{1 + 2\tau}$$

自己相関時間と統計誤差

自己相関時間



統計誤差



マルコフ連鎖モンテカルロ法

- 統計誤差はサンプルの生の分散 s^2 とサンプル数 M、自己相関時間 τ で決まる

$$\sigma^2 \simeq \frac{s^2(1+2\tau)}{M}$$

  - 一度に大きく配位を動かそうとすると棄却率が増加 $\Rightarrow \tau$ が増加
  - 動かす幅を小さくすると棄却率は高いが相関が消えない $\Rightarrow \tau$ が増加
  - 非局所更新法、拡張アンサンブル法など様々な方法が使われている
- 物理以外でも、Bayes 推定や機械学習、社会現象のシミュレーションなど広く使われている

イジング模型に対するモンテカルロ法

- 更新の単位は一つのスピンとするのが一番自然
- メトロポリス法に必要なのは更新前後のエネルギー差だけなので、全エネルギーを計算しなおす必要なし

```
for (s = 0; s < num_sites; ++s) {
  delta = 0.0;
  for (j = 0; j < num_neighbors; ++j) {
    v = neighbor(s, j);
    delta += 2 * J * spin[s] * spin[v];
  }
  if (random() < exp(-beta * delta))
    spin[s] = -1 * spin[s];
}
```

物理量の計算

- 内部エネルギー E
  - 初期状態のスピンを全て上向き (1) に取ると
    $E = -J \times$ ボンド数
  - モンテカルロステップ毎にエネルギーの変化分を計算しているので、採択された場合にはその値を足し込む
- 比熱 C: 内部エネルギーのゆらぎから計算できる

$$C = \frac{1}{N}\frac{\partial E}{\partial T} = \frac{1}{NT^2}(\langle E^2\rangle - \langle E\rangle^2)$$

- 磁化 m: スピンの値の平均値 $m = \frac{1}{N}\sum_i \sigma_i$
  - 外部磁場がない場合、対称性から m の長時間平均は厳密には零になる
  - 熱力学極限では対称性が自発的に破れて、低温で有限の m
  - シミュレーションでは m ではなく m^2 を見るとよい

モンテカルロステップの設定

- 全てのスピンについて一通り更新を試すのを、1 モンテカルロステップと数える
- どれくらいのモンテカルロステップが必要かは、あらかじめは分からない
- 典型的には、10^4—10^6 程度にとることが多い
- 熱平衡化 (thermalization)
  - 初期配位依存性を取り除くため、モンテカルロステップの最初の部分は捨てる (burn-in time)
  - 典型的には、全体の 1 割程度を捨てることが多い