

Population Annealing の紹介

Yuichi Motoyama (ISSP)

2021-10-22

物性アプリオープンフォーラム@東大物性研

目的

- 分布パラメータ β_i のもとでの配位 x の重みを $f_i(x)$ と書く
 - 例1: ボルツマン因子 $f_i(x) = \exp[-\beta_i E(x)]$
 - 例2: 興味のある $f(x)$ に対して、 $f_i(x) = f(x)^{\beta_i}$ とする ($0 \leq \beta_i \leq 1$)
- 適当な観測量 $A(x)$ の期待値は

$$\langle A \rangle_i = \frac{\int dx A(x) f_i(x)}{\int dx f_i(x)}$$

- 規格化因子を求める際の利便性のために少し書き換え
 - 分母（規格化因子・分配関数）を Z_i
 - 配位 x の確率（密度）を $\tilde{f}_i(x) = f_i(x)/Z_i$
 - $\langle A \rangle_i = \int dx A(x) \tilde{f}_i(x)$
- 目的は **複数のパラメータについて期待値および規格化因子（の比）を数値的に求めること**

目的

- 目的は **複数のパラメータについて期待値および規格化因子（の比）を数値的に求めること**
- マルコフ連鎖モンテカルロ法 (MCMC) を用いればそれぞれの f_i に対する期待値を求められる
 - x_j から次の配位 x_{j+1} を遷移確率 $T_n(x_j, x_{j+1})$ で逐次生成
 - 遷移確率が釣り合い条件、エルゴード性、非周期性を満たしていれば、十分長い時間の後に配位列 $\{x_j\}$ 中の頻度分布が \tilde{f}_i に従う
 - 配位列に対する単純平均を計算すれば良い
 - 😊 パラメータ独立に計算ができる = 自明並列化可能
 - 😞 **十分長い時間** が現実的な時間で収まるか???
 - 臨界減速、一次転移、スピングラス、などなど
 - 特に局所安定解にトラップされるとつらい
 - 😞 分配関数を求めるのは難しい

Simulated Annealing (SA)

- 計算の簡単な分布（高温極限 $\beta_0 = 0$ ）から MCMC を始める
- 適当なタイミングで徐々に冷却 $\beta_i \rightarrow \beta_{i+1} > \beta_i$
- 😊 局所安定解へのトラップから抜けられる可能性がある
- 😞 高温からの逐次計算になる
- 😞 強制的に温度を変える操作は分布を歪ませる
 - 期待値を求めたい場合は毎温度で緩和が必要
 - どちらかというともサンプリングよりは最適化向け
- 温度変化も MCMC の枠組みで捉えればいいのでは？
 - → Simulated tempering, exchange MC (parallel tempering)
- 冷却時の影響を後で補填すればよいのでは？
 - → Annealed importance sampling, population annealing

Annealed Importance Sampling (AIS)

Ref: R. M. Neal, *Statistics and Computing* **11**, 125-139 (2001).

- 次の同時確率分布 \tilde{f} に従う配位列(経路) $\{x_0, x_1, \dots, x_n\}$ を考える

$$\tilde{f}(x_0, x_1, \dots, x_n) = \tilde{f}_n(x_n) \tilde{T}_n(x_n, x_{n-1}) \tilde{T}_{n-1}(x_{n-1}, x_{n-2}) \cdots \tilde{T}_1(x_1, x_0)$$

$$\tilde{T}_i(x_i, x_{i-1}) = T_i(x_{i-1}, x_i) \frac{\tilde{f}_i(x_{i-1})}{\tilde{f}_i(x_i)}$$

- $T_i(x, x')$ は β_i のもとでの配位 x から x' への遷移確率
 - $T_i(x, x')$ は釣り合い条件をみたすようにする

$$\int dx \tilde{f}_i(x) T_i(x, x') = \tilde{f}_i(x')$$

- このとき、 $\tilde{f}_n(x_n)$ は $\tilde{f}(x_0, x_1, \dots, x_n)$ の周辺分布となる

$$\begin{aligned} \int \prod_{i=0}^{n-1} dx_i \tilde{f}(x_0, x_1, \dots, x_n) &= \int \prod_{i=0}^{n-1} dx_i \tilde{f}_n(x_n) \tilde{T}_n(x_n, x_{n-1}) \tilde{T}_{n-1}(x_{n-1}, x_{n-2}) \cdots \tilde{T}_1(x_1, x_0) \\ &= \tilde{f}_n(x_n) \end{aligned}$$

$$\therefore \int dx_{i-1} \tilde{T}_i(x_i, x_{i-1}) = \int dx_{i-1} \frac{\tilde{f}_i(x_{i-1}) T_i(x_{i-1}, x_i)}{\tilde{f}_i(x_i)} = 1 \quad (\text{釣り合い条件より})$$

- したがって、 \tilde{f}_n における平均値 $\langle A \rangle_n$ は経路の重み付き平均（あるいはサンプル平均）としてかける

$$\begin{aligned} \langle A \rangle_n &\equiv \int dx_n A(x_n) \tilde{f}_n(x_n) \\ &= \int \prod_{i=0}^n dx_i A(x_n) \tilde{f}(x_0, x_1, \dots, x_n) \end{aligned}$$

- $\tilde{f}_n(x_n)$ から x_n をサンプルして x_{n-1}, x_{n-2}, \dots と生成する？
- 始点となる $\tilde{f}_n(x_n)$ から直接サンプルするのが難しいケースを考えている
 - これができるならそもそもサンプル平均をとれる
 - $\tilde{f}(x_0, x_1, \dots, x_n)$ に従うような点列を直接生成することはむずかしい

$$\tilde{f}(x_0, x_1, \dots, x_n) = \tilde{f}_n(x_n) \tilde{T}_n(x_n, x_{n-1}) \tilde{T}_{n-1}(x_{n-1}, x_{n-2}) \cdots \tilde{T}_1(x_1, x_0)$$

$$\tilde{T}_i(x_i, x_{i-1}) = T_i(x_{i-1}, x_i) \frac{\tilde{f}_i(x_{i-1})}{\tilde{f}_i(x_i)}$$

- 一方で $\tilde{f}_0(x)$ として、サンプリングが簡単な分布を用意できることが多い
 - ボルツマン分布の高温極限 $\beta_0 = 0$ は $f_0(x) = 1$
 - $f_i(x) = f(x)^{\beta_i}$ の時は $\beta_0 = 0$ でやはり $f_0(x) = 1$
- \tilde{f}_0 から始めて逆にたどってみる

1. 確率 $\tilde{f}_0(x)$ に従う x_0 を生成する

2. x_i から $T_{i+1}(x_i, x_{i+1})$ によって x_{i+1} を生成する

◦ 確率行列 T_{i+1} は釣り合い条件を満たすので、普通にMCMCを行えば良い

という流れに従ってパス $\{x_i\}$ を生成すると、これは同時確率分布

$$\tilde{g}(x_0, x_1, \dots, x_n) = \tilde{f}_0(x_0) T_1(x_0, x_1) T_2(x_1, x_2) \dots T_n(x_{n-1}, x_n)$$

に従う。これを利用すると期待値 $\langle A \rangle_n$ は

$$\begin{aligned} \langle A \rangle_n &= \int \prod_i dx_i A(x_n) \tilde{f}(x_0, x_1, \dots, x_n) \\ &= \int \prod_i dx_i A(x_n) \frac{\tilde{f}(x_0, x_1, \dots, x_n)}{\tilde{g}(x_0, x_1, \dots, x_n)} \tilde{g}(x_0, x_1, \dots, x_n) \\ &= \langle A \tilde{f} / \tilde{g} \rangle_{g,n} \end{aligned}$$

と評価できる (reweighting method)。

$\langle A\tilde{f}/\tilde{g} \rangle_{g,n}$ にでてくる \tilde{f} と \tilde{g} との比は

$$\begin{aligned} \frac{\tilde{f}(x_0, \dots, x_n)}{\tilde{g}(x_0, \dots, x_n)} &= \frac{\tilde{f}_n(x_n)}{\tilde{f}_0(x_0)} \prod_{i=1}^n \frac{\tilde{T}_i(x_i, x_{i-1})}{T(x_{i-1}, x_i)} \\ &= \frac{\tilde{f}_n(x_n)}{\tilde{f}_0(x_0)} \prod_{i=1}^n \frac{\tilde{f}_i(x_{i-1})}{\tilde{f}_i(x_i)} \\ &= \frac{Z_0}{Z_n} \frac{f_n(x_n)}{f_0(x_0)} \prod_{i=1}^n \frac{f_i(x_{i-1})}{f_i(x_i)} \\ &= \frac{Z_0}{Z_n} \prod_{i=0}^{n-1} \frac{f_{i+1}(x_i)}{f_i(x_i)} \\ &\equiv \frac{Z_0}{Z_n} w_n(x_0, x_1, \dots, x_n) \end{aligned}$$

となるので、代入すると (→次ページ)

$$\langle A \rangle_n = \langle A \tilde{f} / \tilde{g} \rangle_{g,n} = \frac{Z_0}{Z_n} \langle Aw \rangle_{g,n}$$

となる。ここで $\langle \mathbf{1} \rangle_n = \mathbf{1}$ を利用すると、分配関数の比は

$$\frac{Z_n}{Z_0} = \langle w \rangle_{g,n}$$

となり、 A の期待値は

$$\langle A \rangle_n = \frac{\langle Aw \rangle_{g,n}}{\langle w \rangle_{g,n}}$$

となる。

- 分母・分子ともにパスのサンプリングを用いて評価することになる
- \tilde{g} に従うようなパス $\{x_i^{(k)}\}$ を K 個独立に生成して ($k = 1, 2, \dots, K$)

$$\begin{aligned} \langle A \rangle_n &= \frac{\langle Aw \rangle_{g,n}}{\langle w \rangle_{g,n}} \\ &= \frac{\sum_{k=1}^K A(x_n^{(k)}) w(x_0^{(k)}, x_1^{(k)}, \dots, x_n^{(k)})}{\sum_{k=1}^K w(x_0^{(k)}, x_1^{(k)}, \dots, x_n^{(k)})} \end{aligned}$$

- すなわち、配位列に対して次で定義された Neal-Jarzynski 重みによる加重平均となる

$$w^{(k)} = w(x_0^{(k)}, x_1^{(k)}, \dots, x_n^{(k)}) = \prod_{i=0}^{n-1} \frac{f_{i+1}(x_i)}{f_i(x_i)}$$

- K 個の random walker を用いた annealed importance sampling の擬似コードは次の通り（配列などの初期化は割愛）：

```
for k in range(K):
    w[0, k] = 1.0
    x[0, k] = draw_from( $\beta[0]$ )
for i in range(1, N):
    for k in range(K):
        w[i, k] = w[i-1, k] * ( f(x[i-1, k],  $\beta[i]$ )
                               / f(x[i-1, k],  $\beta[i-1]$ ) )
        x[i, k] = transfer(x[i-1, k], beta[i])
    a[i] = sum(A(x[i, :])*w[i, :]) / sum(w[i, :])
```

- 標準誤差などを計算したい場合は jackknife 法や bootstrap 法などを使うべき

AIS の利点

- x, w を保存して期待値を最後にまとめて計算することで、 k のループを外側に出すことも可能
 - 必要ならば追加計算も可能！
- 温度が強制的に変わる
 - 交換法は調整に失敗すると温度変化が起きないことがある
- サンプルングに無駄がない
 - パス間の相関は無い（始点が独立ならば）
 - **パス内の相関も気にしなくて良い**
 - 各温度内での $T_i(x, x')$ による更新は **1step でもよい**
 - 0step まで減らすとただのランダムサンプリングに帰着する
 - 並列化数を増やすとそのままサンプル数が向上する

Population Annealing (PA) [2]

Ref: [K. Hukushima and Y. Iba, AIP Conf. Proc. 690, 200 \(2003\).](#)

- AIS を使うと各 β に対する期待値を重み付き平均という形で計算できる
 - 局所安定にトラップされると重みが小さくなる
 - 結果として重み w の分散が大きくなりがち
 - 実効的なサンプル数が減り、精度が悪くなる
- AIS にwalker のリサンプリング操作を追加したものがPA
 - 適当な周期で確率 $p^{(k)} = w^{(k)} / \sum_k w^{(k)}$ に従い walker をリサンプリング
 - walker に割り当てられた重みをリセット ($w = 1$) する

擬似コード

```
for k in range(K):
    w[0, k] = 1.0
    x[0, k] = draw_from( $\beta[0]$ )
for i in range(1, N):
    for k in range(K):
        w[i, k] = w[i-1, k] * ( f(x[i-1, k],  $\beta[i]$ )
                               / f(x[i-1, k],  $\beta[i-1]$ ) )

    if i % interval == 0:
        x[i, :] = resample(x[i, :], w[i, :])
        w[i, :] = 1.0
    for k in range(K):
        x[i, k] = transfer(x[i-1, k],  $\beta[i]$ )
    a[i] = sum(A(x[i, :]) * w[i, :]) / sum(w[i, :])
```

resampling のやり方

- walker 数が変化しない方法と変化する方法の2通りのやり方がある
- resampling 頻度もハイパーパラメータ
- 近年の論文は後者の方法で毎回resampling していることがおおい？

walker 数固定 resampling (K. Hukushima and Y. Iba, *op. cit.*)

- 各 walker 毎に $p^{(k)}$ に従って整数 k を生成し、配位 $x^{(k)}$ を採用する
 - walker の数が多いので、Walker's alias method を用いるのがよい

walker 数変動 resampling (J. Machta, PRE 82, 026704 (2010))

- 各walker を独立に消滅・分裂させることでresampling を達成する
 - 各walker に対して期待値 $\lambda^{(k)} = Np^{(k)}$ のポアソン分布に従うように整数 $n^{(k)}$ をランダム生成
 - walker を複製・消滅させて $n^{(k)}$ 個にする

PA の利点

- 基本的には AIS と同じ
 - 温度が強制的に変わる
 - 温度ごとのステップ数は少なくても良い
- 局所安定にトラップされたwalker がもっとエネルギーの低いものに生まれ変わることがある

PA の注意点

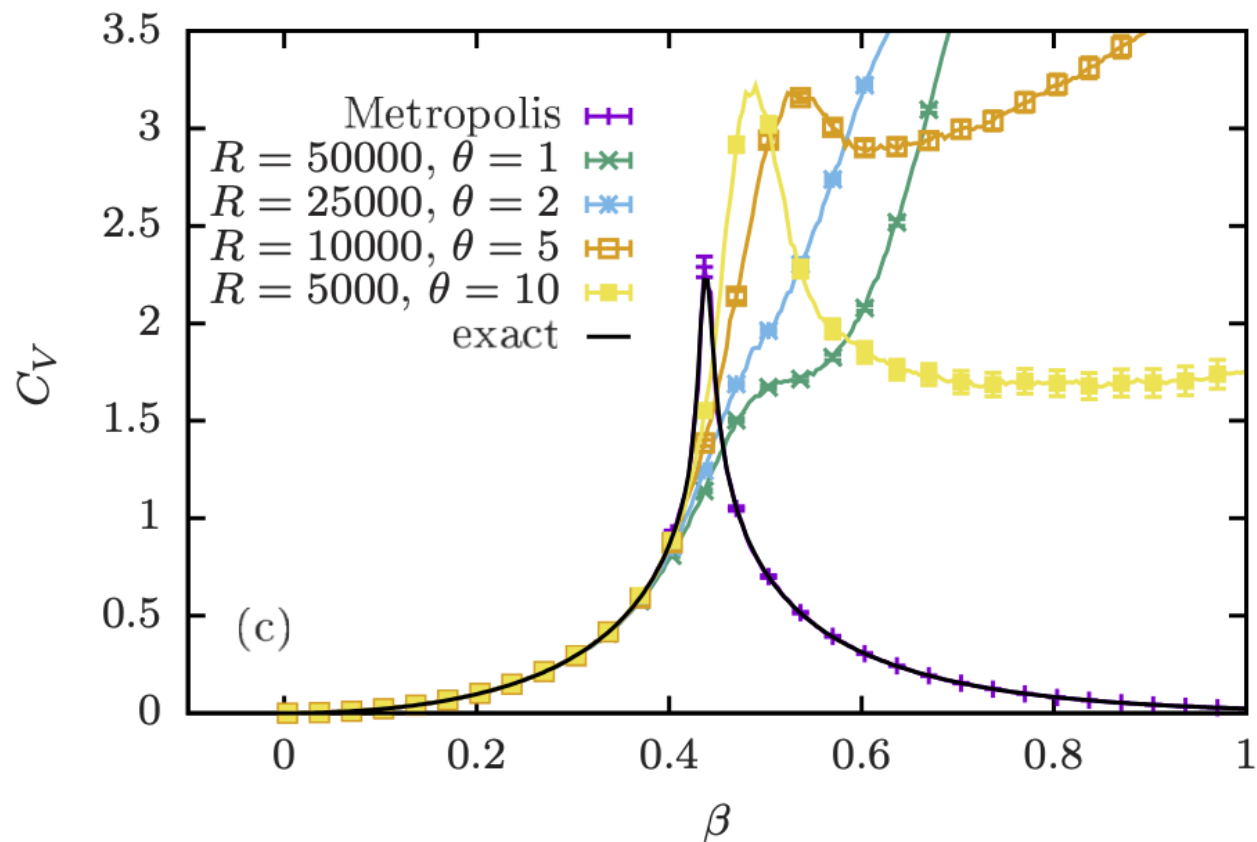
- リサンプリング操作には通信が必要
- リサンプリング操作によってwalker 間に相関が発生する
 - 実効的なwalker 数が減少する
 - 各温度での MCMC のステップ数を増やせば相関は減る
 - エラーバーの付け方が非自明になる
 - 追加計算として、単純にwalker を追加して足すことができない
 - 独立なPA 計算として外側であらためて平均や誤差を見積もることはできる
- walker 間の相関や有効サンプル数などについては次の論文が詳しく議論している
 - [M. Weigel, L. Barash, L. Shcur, and W. Janke, 103, 053301 \(2021\)](#)

AIS によるイジング模型 の比熱計算 (64×64)

From M. Weigel, L. Barash, L. Shcur,
and W. Janke, **103**, 053301 (2021)

(Fig1c)

- R はレプリカの数
- θ は温度ごとのMCMC ステップ数
- AIS だと相転移を越えられない

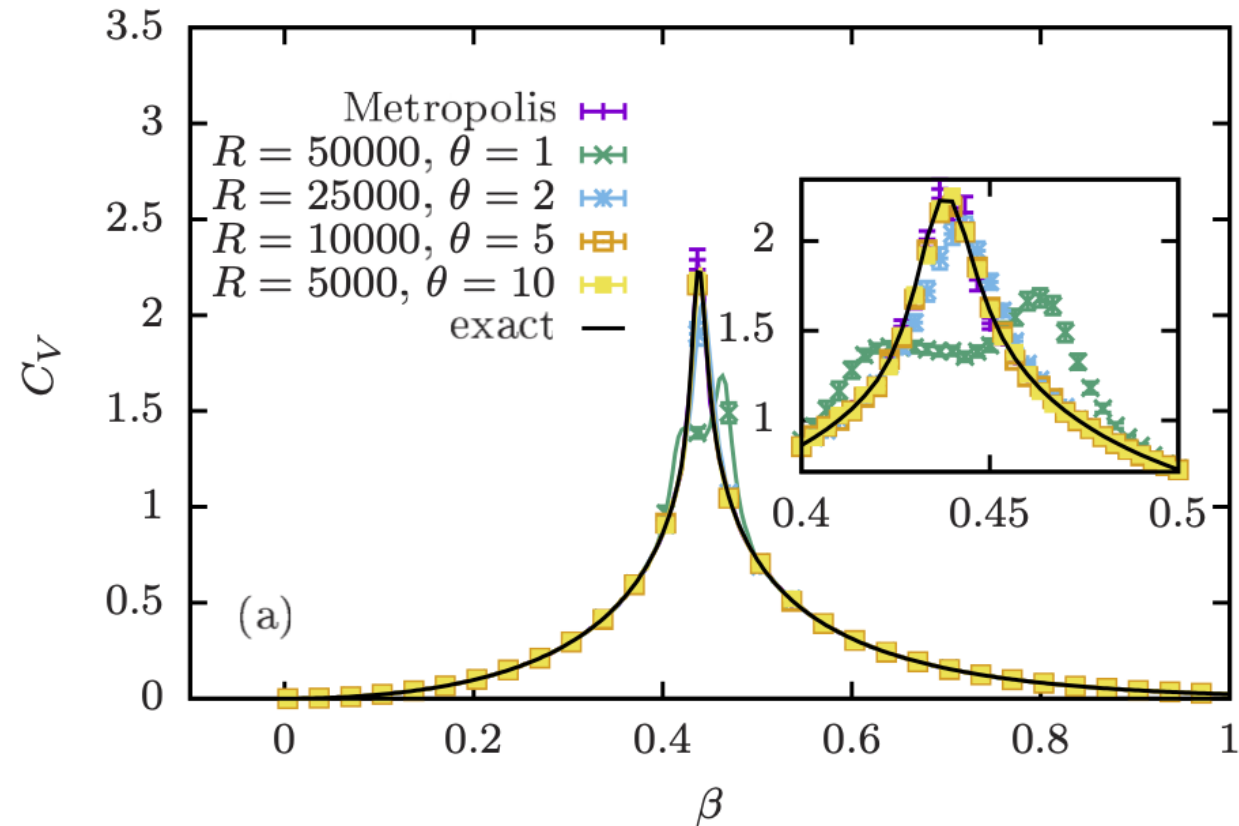


PA によるイジング模型 の比熱計算 (64×64)

From M. Weigel, L. Barash, L. Shcur,
and W. Janke, **103**, 053301 (2021)

(Fig1a)

- R はレプリカの数
- θ は温度ごとのMCMC ステップ数
- **PA なら相転移を越えられる!**
 - 同じ計算数ならばMCMC ステップ数を増やしたほうがよい



スピングラス計算 (Simple cubic Ising)

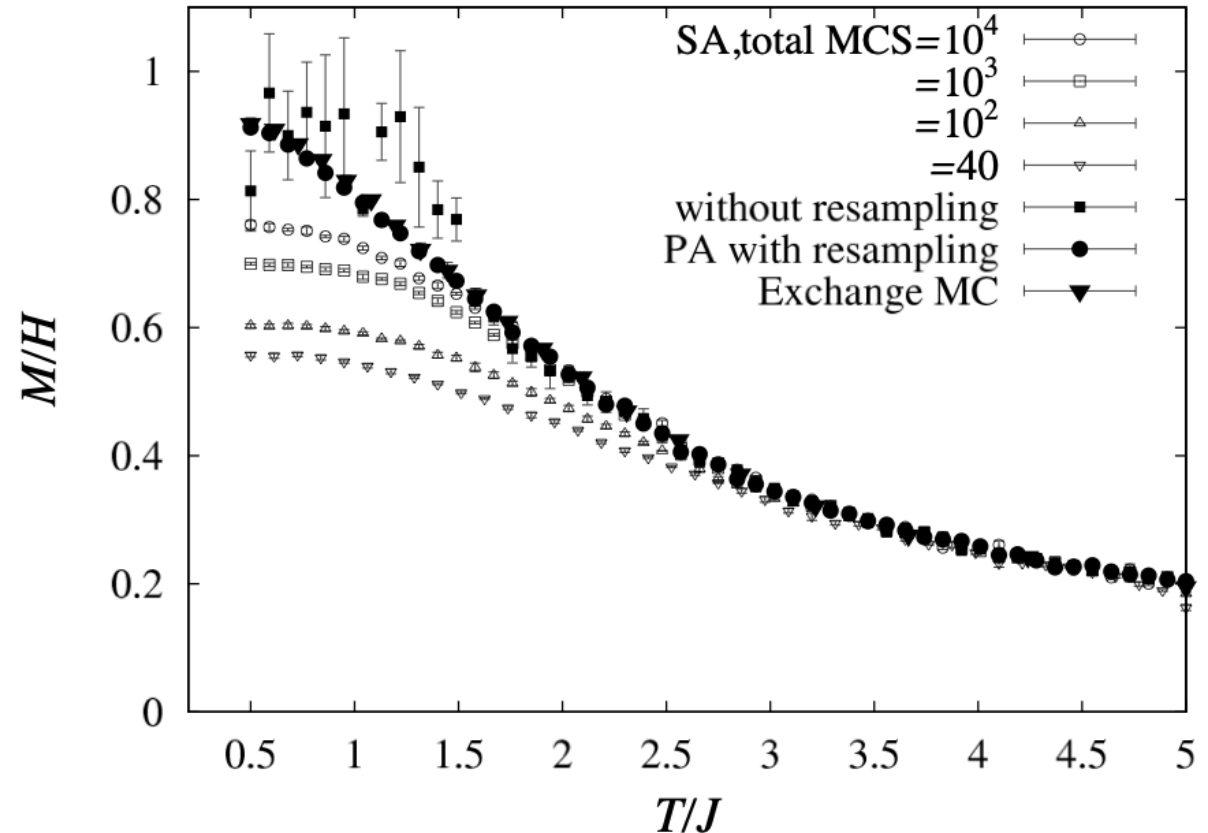
From [K. Hukushima and Y. Iba, *op. cit.*](#)

(Fig2)

$$\mathcal{H} = \sum_{\langle ij \rangle} J_{ij} \sigma_i^z \sigma_j^z - H \sum_i \sigma_i^z$$

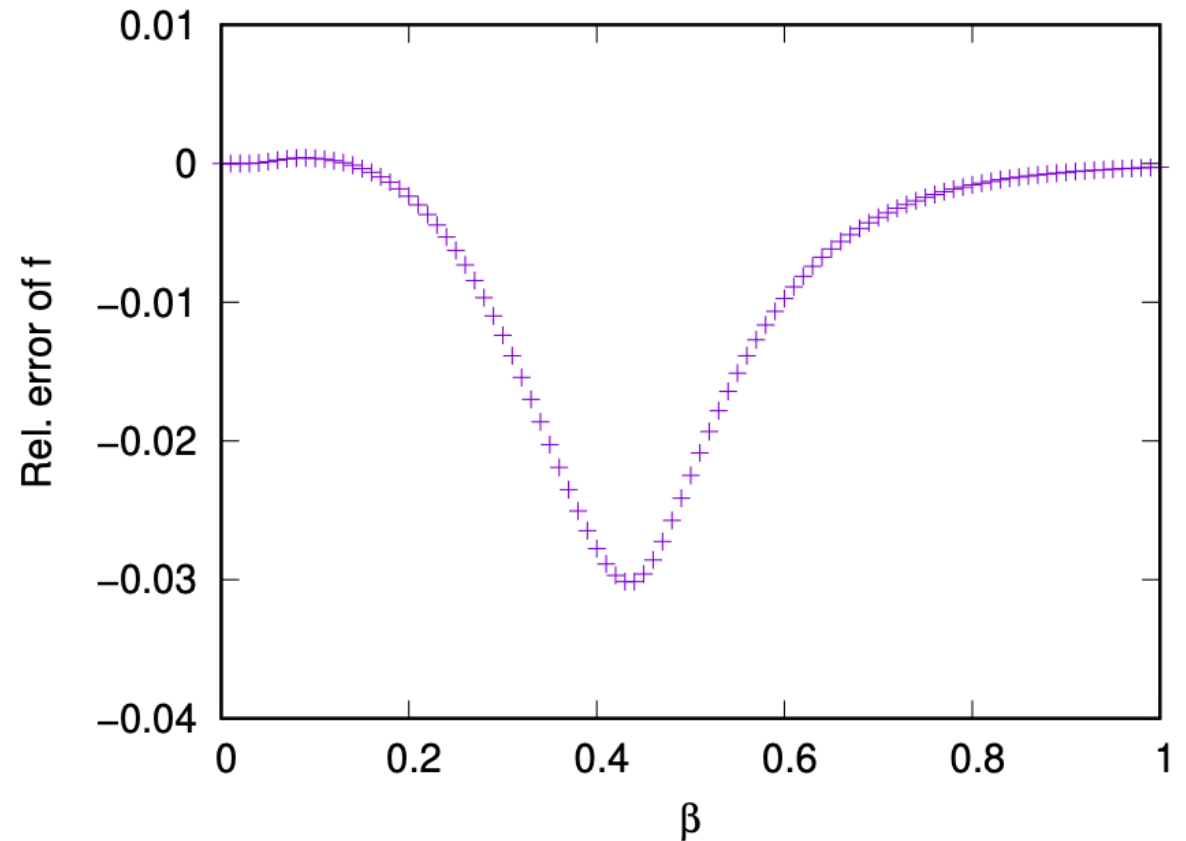
$$J_{ij} \sim N(0, J^2), \quad H/J = 0.1$$

- SA (open symbol) は緩和しきっていない
- AIS (filled square) は低温で精度が悪い
- PA (filled circle) と 交換MC (filled triangle) は互いにコンパラ



PA によるイジング模型 の自由エネルギー計算 (32×32)

- $R = 18432$ $\theta = 1$
- 自由エネルギーの $L = \infty$ での厳密解からの相対誤差
- 2DMAT (次ページ) に修正を加えてイジング模型を計算できるようにしたもの



2DMAT

- 最適化やモンテカルロサンプリングを行うソフトウェア
 - Nelder-Mead, グリッドサーチ、ベイズ最適化
 - 交換モンテカルロ法、**PA** (v2 β 以降)
- web: <https://www.pasums.issp.u-tokyo.ac.jp/2dmat>
 - 過去の発表資料が上部メニューの「ドキュメント」 → 「**発表資料**」で見られます
- GitHub: <https://github.com/issp-center-dev/2DMAT>
- 2020,2021 年度の物性研ソフトウェア開発・高度化プロジェクトの支援を受けて開発されています

まとめ

- ポピュレーション型のサンプリング手法である AIS と PA のアルゴリズムを紹介した
 - 並列計算機の性能を活かしやすいアルゴリズムになっている
 - PA は AIS よりも低温や相転移に強い
- PA を実装しているソフトウェアとして2DMAT の紹介をした
- 謝辞
 - 東大駒場の福島先生には、PA のレクチャーや質問への回答など、たいへんお世話になっております