

成果発表・情報共有

Group A

物性研 金子 隆威

議題の目的（Combo + $H\Phi$ に集中）

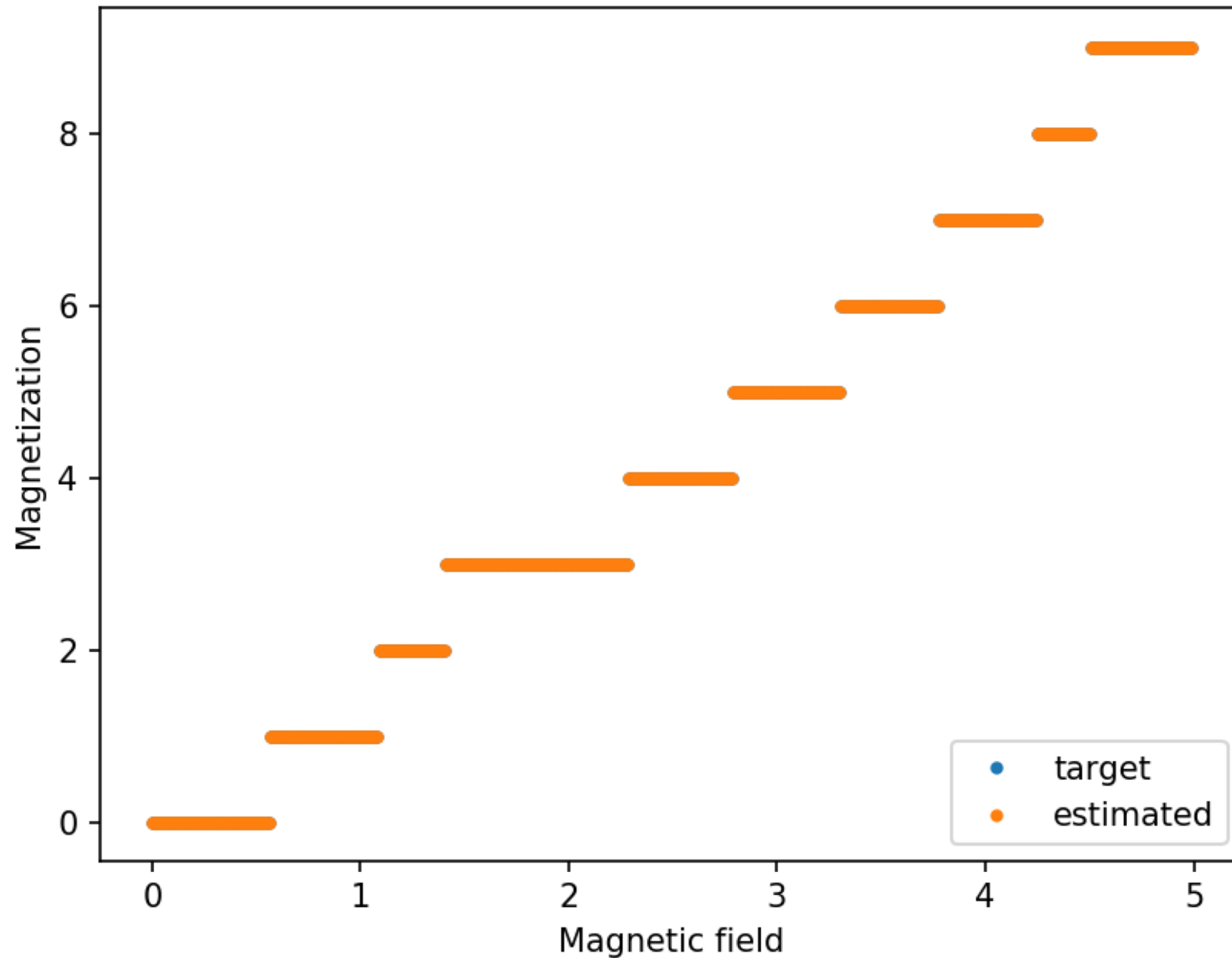
- 他の格子系でも試したい
 - 2次元フラストレート系：とくに、三角格子Heisenberg模型の磁化過程
 - 幾何学的フラストレーション効果？ $1/3$ プラトーは再現される？
- Scoreや乱数の選び方で結果が変わるか調べたい
 - Score：TS (default), EI, PI
 - 乱数：6通り
- 他の物理量でも動作確認したい
 - 与えられた比熱を再現するような有効模型推定

アプローチ方法（他の格子系）

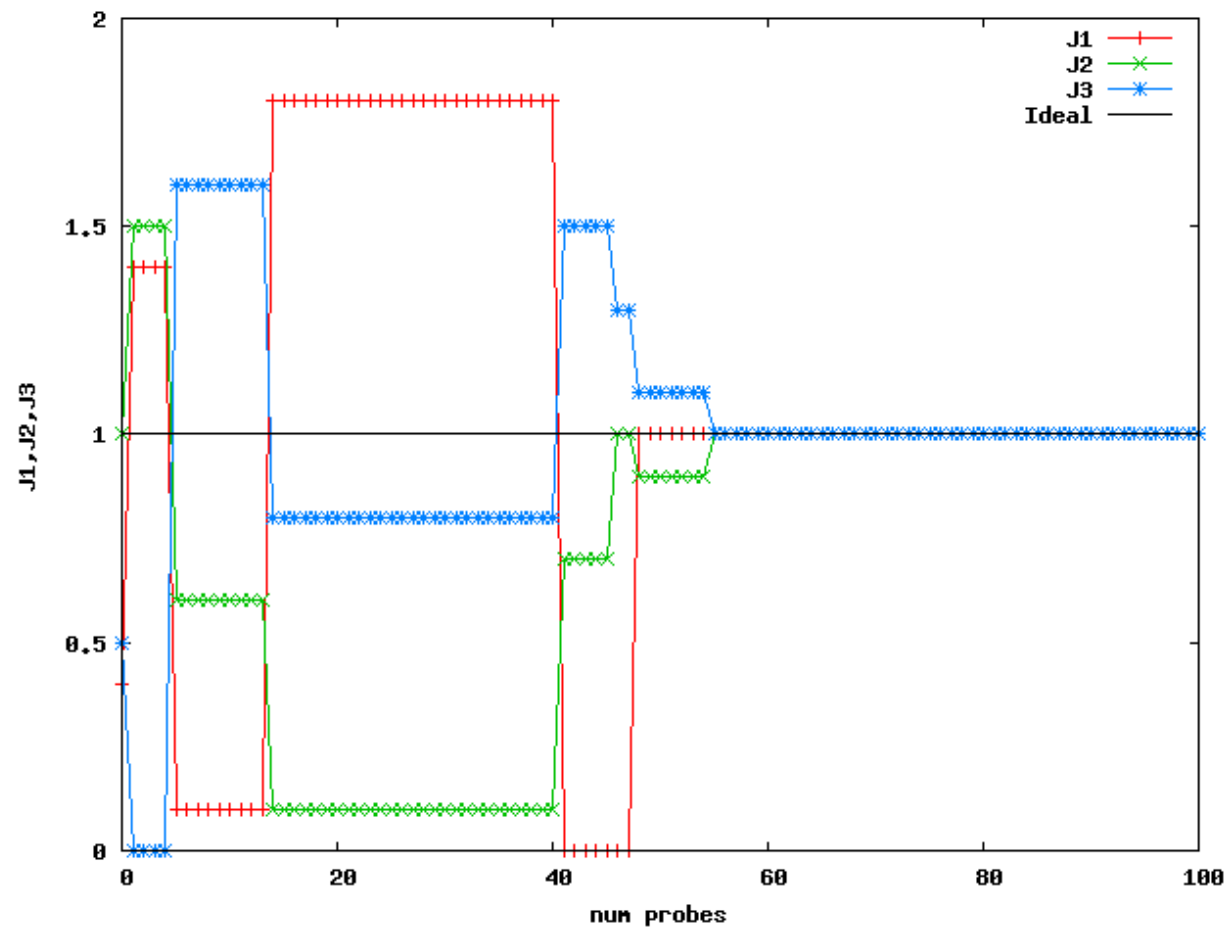
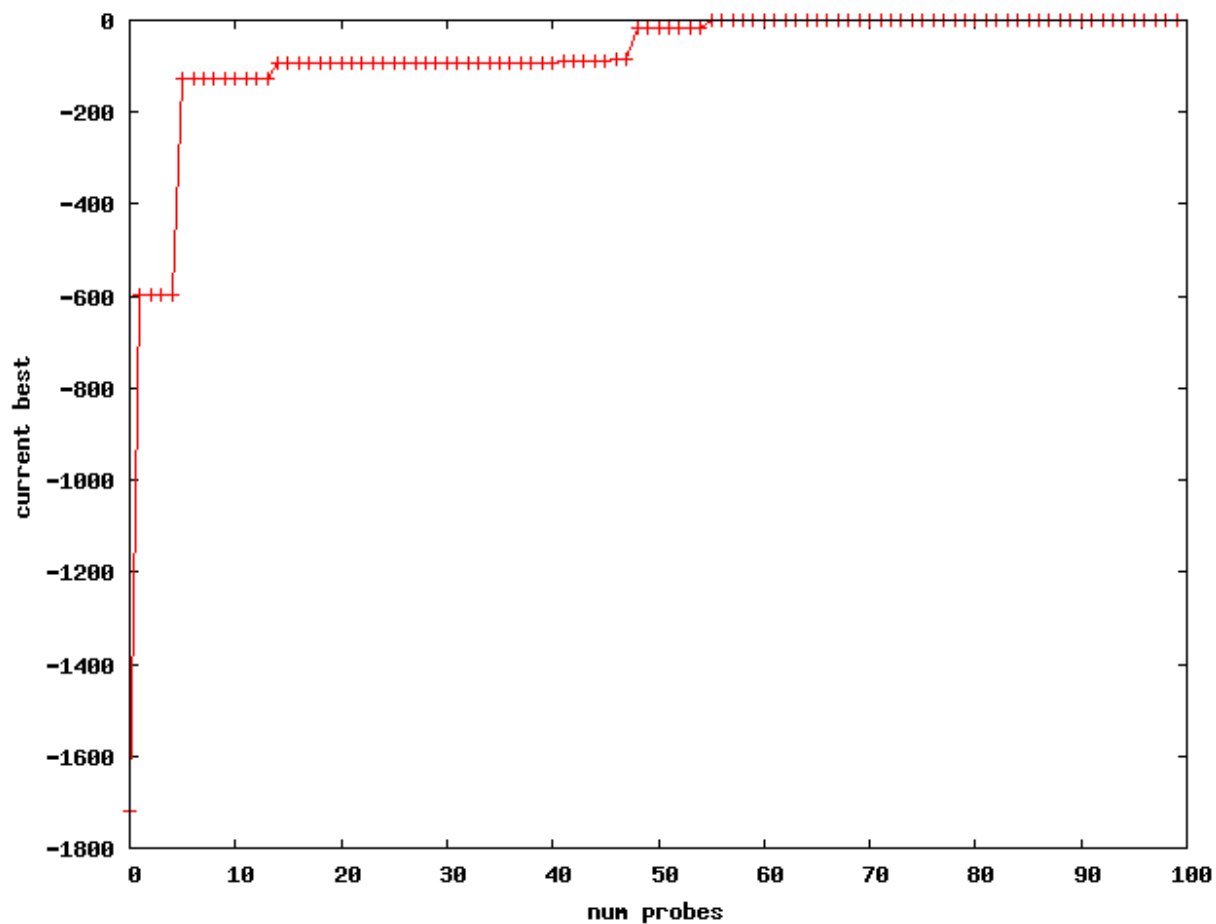
- 元のCombo+H Φ コードを流用
- 1次元→2次元：1変数 $L \rightarrow 2$ 変数 (L, W)
- 飽和磁場が元々の模型と異なるので、
magnetic_field=[0.01*num for num in range(maxnum)]
のmaxnumを適切に設定（ $(L, W)=(6, 3)$ のとき500でOK）
- x方向、y方向、x=y方向の相互作用をJ1, J2, J3に設定
- 6x3 sites、J1=J2=J3=1の結果を再現するか？
- J1, J2, J3を[0,2]区間でそれぞれ20刻み
- 最初の20 stepsは乱数、残りの80 stepsでBayesian最適化

結果

- 磁化過程を再現
- 1/3プラトー
- $J_1=J_2=J_3=1$ を正しく見つけていた



結果 (Bayesian最適化過程)



アプローチ方法（Score・乱数依存性）

- 先ほどと同じ条件
- 乱数6通り
 - 1, 11, 111, 1111, 11111, 111111
- Scoreの選び方3通り
 - TS (default), EI, PI

結果（抜粋）

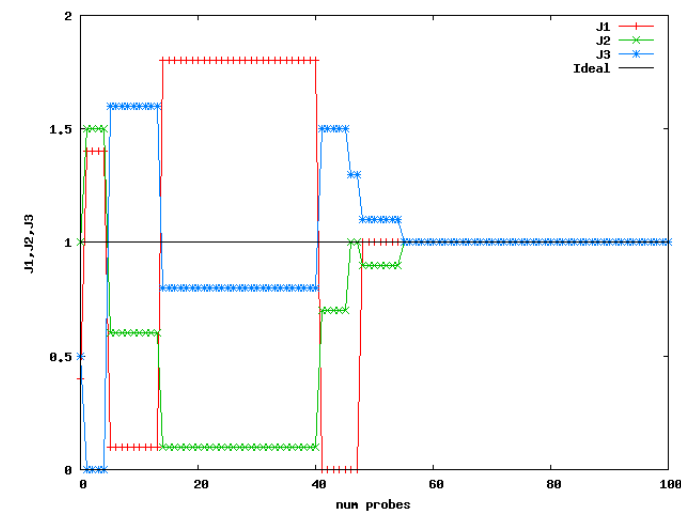
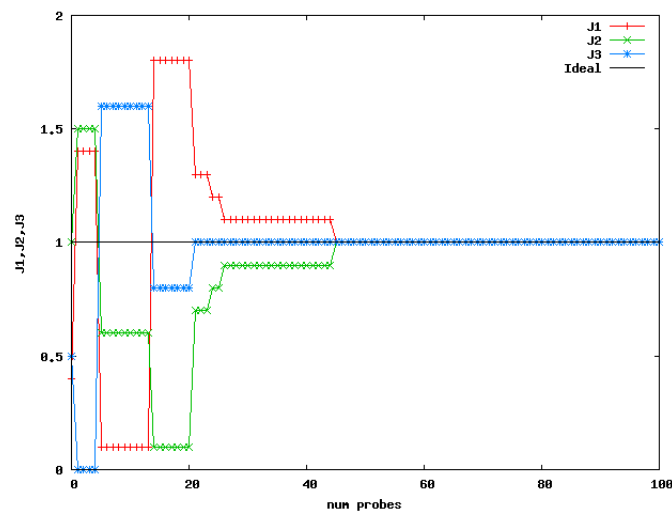
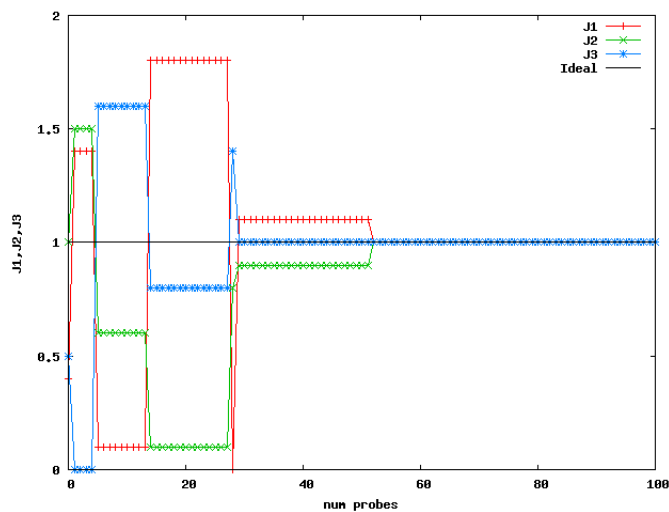
（18通り、すべて正しく収束していた）

EI

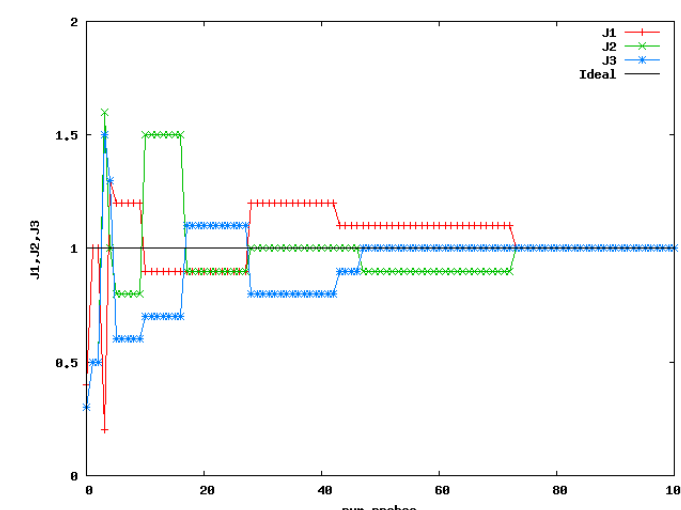
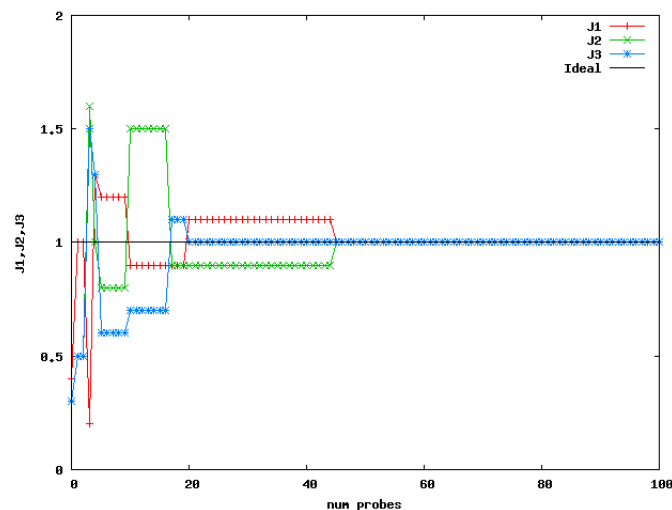
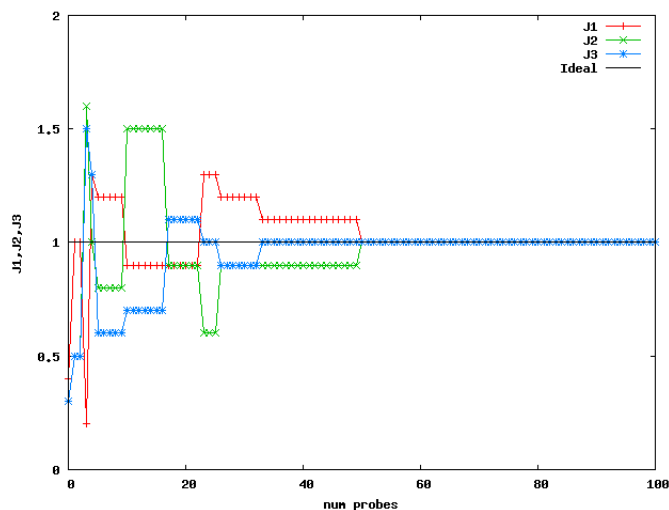
PI

TS

rnd111



rnd1111

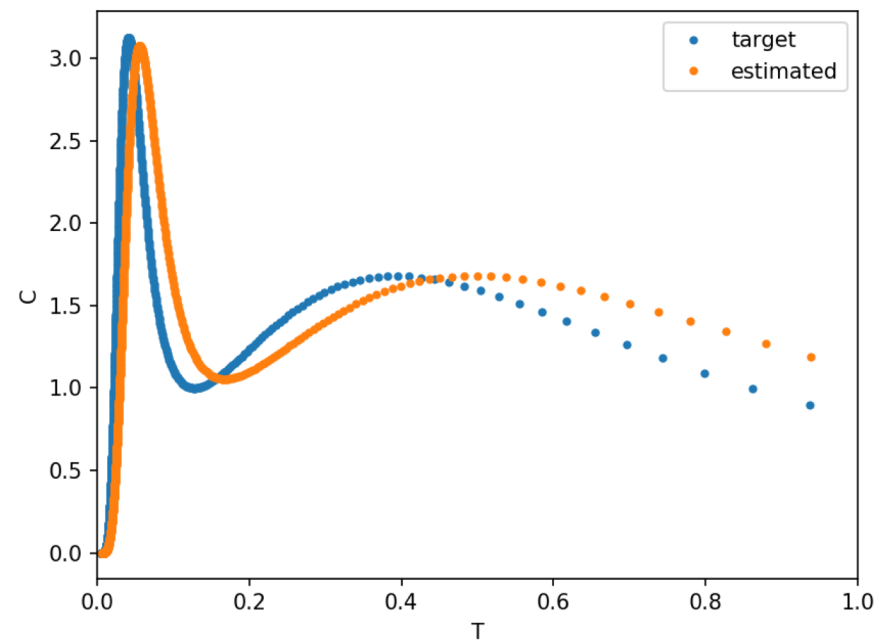
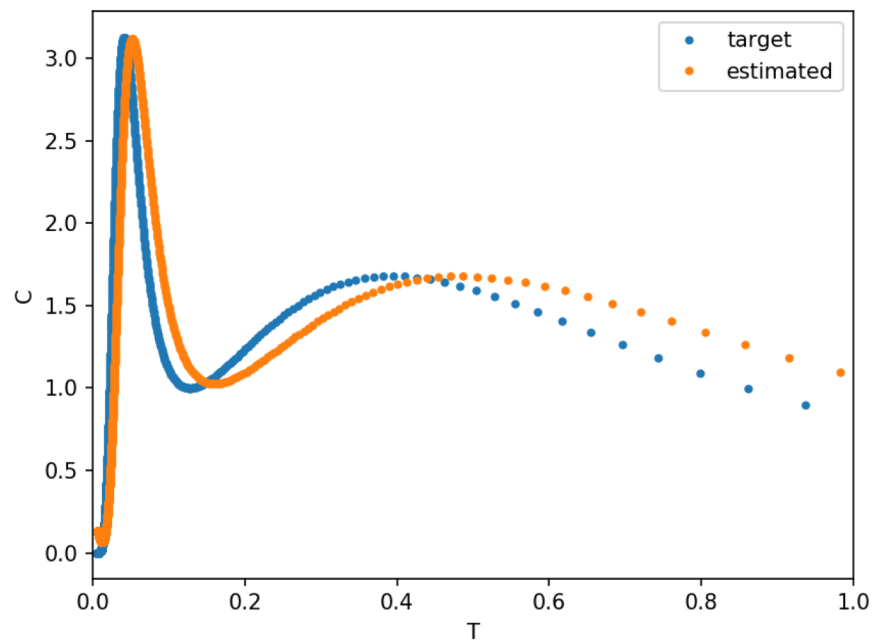
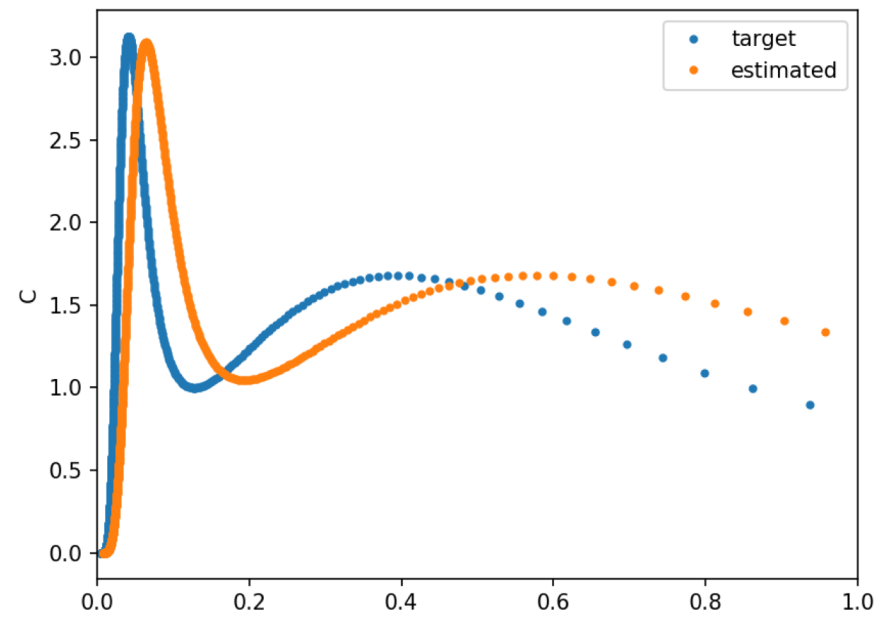


アプローチ方法（比熱による模型推定）

- Kitaev模型の比熱を用いて推定
 - $H\Phi$ のexample : <https://github.com/issp-center-dev/HPhi/tree/master/samples/TPQ/Kitaev>
- TPQアルゴリズム使用
- 評価関数 : $\delta = \sum [(C - C_{\text{target}})^2 + (T - T_{\text{target}})^2]$
- 12 sites, $J_x = J_y = J_z = -1$ の結果を再現するか？
- 区間 $[-2.1, -0.1]$ で20刻み（ $H=0$ になる $J_x = J_y = J_z = 0$ は除外）
- 最初の20 stepsは乱数、残りの80 stepsでBayesian最適化

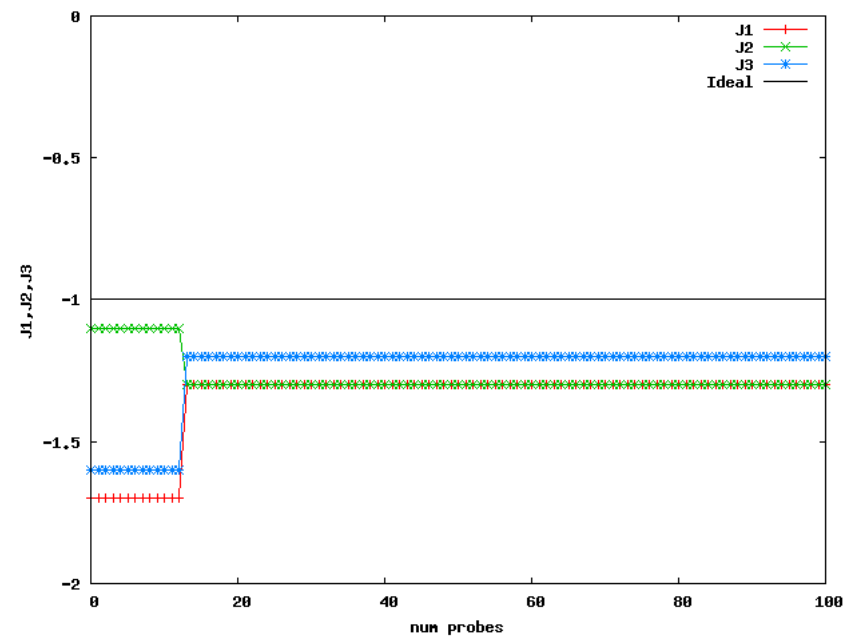
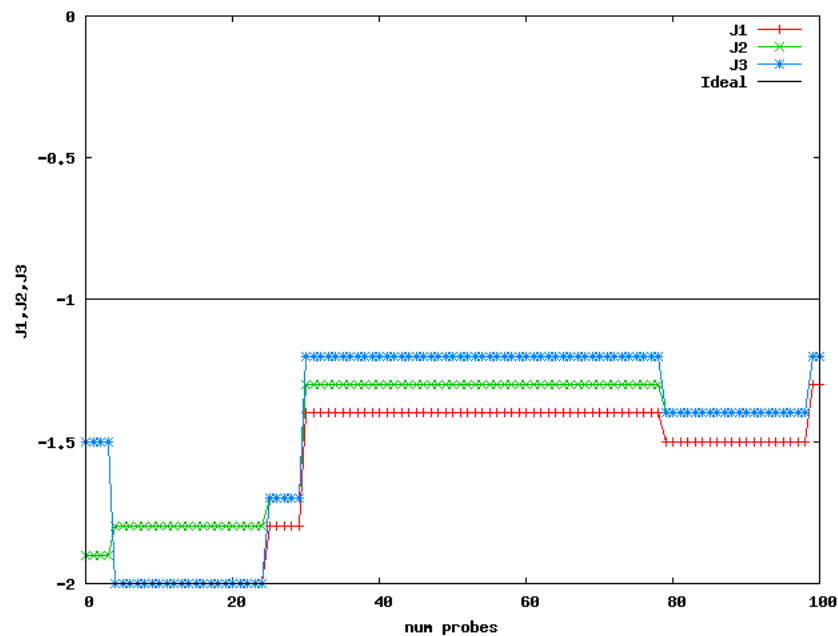
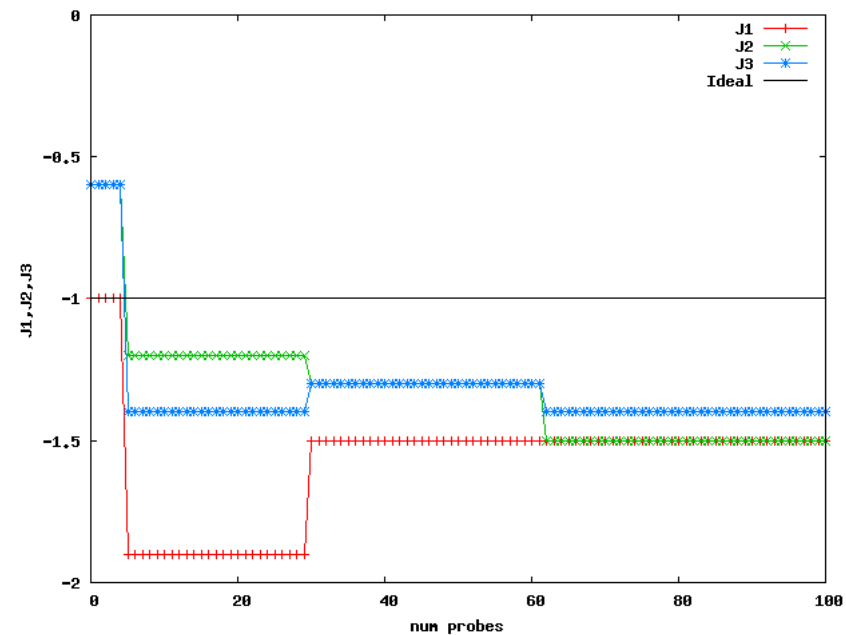
結果

- 乱数3種類考慮
- 一致しない…



結果

- $J_x = J_y = J_z = -1$ から
ややずれている...



まとめ

- 磁化を用いたモデルの推定
 - 結果は良好
 - 複数の乱数から始めても、期待通り収束？
 - スコア依存性も小さい？
- 比熱だと難しい？
 - 有限温度物理量だから？
 - 初めの乱数探索（現状20 stepsのみ）をもっと増やすべき？

困難だった点（あまり本質的でない…）

- Comboマニュアルなし→関数オプションが分かりづらい…
- Python力が足りなかった
 - HΦのTPQのデフォルト・スクリプトに0割りバグ…
 - matplotlibでinteractiveに図をdisplayせず、figure fileに描き出す方法が分からなかった
解決法：<https://stackoverflow.com/questions/15713279/calling-pylab-savefig-without-display-in-ipython>
`import matplotlib # 追加`
`matplotlib.use('Agg') # 追加`
`import matplotlib.pyplot as plt`
`#fig = plt.figure(figsize=(10, 6)) # コメントアウト`
- その他いろいろ…