

# 量子格子模型の 経路積分モンテカルロ法 ソルバーDSQSSの高度化

東大物性研 本山裕一

2019-04-03

物性研スパコン共同利用・CCMS合同研究会

# Outline

- 経路積分モンテカルロ法について
- DSQSS (Discrete Space Quantum Systems Solver) について
- DSQSS の高度化について
- まとめ・展望
- 付録

このスライドは DSQSS のsite (<https://issp-center-dev.github.io/dsqss>) にアップロードします

# マルコフ連鎖モンテカルロ法 (MCMC)

N. Metropolis, et al., J. Chem. Phys. **21**, 1087 (1953)

- 重み付き平均値を統計誤差の範囲内で厳密に計算する数値手法

$$\langle A \rangle = \sum_c A(c)W(c) / \sum_c W(c)$$

- 出現頻度が重み $W(c)$ に従うような配位列  $\{c_i\}$  を生成し、 $A(c)$ の平均を取る

- 今の配位から次の配位を確率的に生成していく (マルコフ連鎖)

$$c_1 \rightarrow c_2 \rightarrow \dots \rightarrow c_n \rightarrow \dots$$

- 次の配位を作るための確率 (遷移確率)  $p$  をうまく調整することで、任意の重み $W$  を表現できる

- 釣り合い条件  $W(c') = \sum_c W(c)p(c \rightarrow c')$

- 詳細釣り合い条件  $W(c')p(c' \xrightarrow{c} c) = W(c)p(c \rightarrow c')$

- これらの条件を満たすような  $p$  を作るアルゴリズム

- メトロポリス法、熱浴法、諏訪・藤堂法

- 他にエルゴード性と非周期性が必要

# 経路積分モンテカルロ法 (PIMC)

M. Suzuki, Prog. Theor. Phys. **56**, 1454 (1976), N. V. Prokofev, *et al.*, Sov. Phys. JETP **87**, 310 (1998)

- 量子系Hにおける有限逆温度 $\beta$ での物理量Aの期待値を計算したい

$$\langle \hat{A} \rangle = \frac{1}{Z} \text{Tr} \hat{A} e^{-\beta \hat{H}} \quad Z = \text{Tr} e^{-\beta \hat{H}}$$

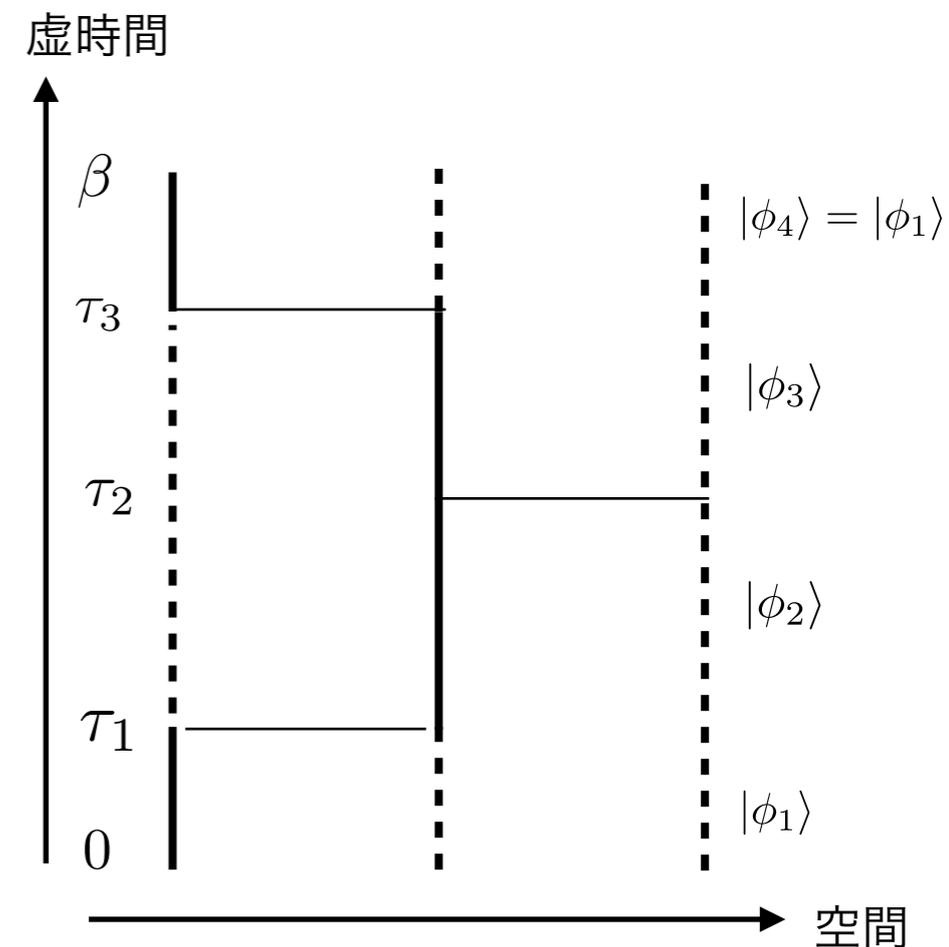
- MCMCを実行するためには分配関数Zをc-数 $W(c)$ (重み)の和で表す必要がある  $\rightarrow$  経路積分

$$\begin{aligned} Z = \text{Tr} e^{-\beta \hat{H}} &= \sum_{n=0}^{\infty} \sum_{\{\phi_k\}} \frac{1}{n!} \mathcal{T}_{\tau} \prod_{k=1}^n \int_0^{\beta} \langle \phi_{k+1} | -\hat{H} | \phi_k \rangle d\tau_k \\ &= \sum_c W(c) (d\tau)^n \end{aligned}$$

重み  $W(c) = \frac{1}{n!} \mathcal{T}_{\tau} \prod_{k=1}^n \langle \phi_{k+1} | -\hat{H} | \phi_k \rangle$

配位 (世界線)  $c = (n, \{\phi_k\}, \{\tau_k\}) \longrightarrow$

配位を効率よく更新したい



# 世界線の更新手法

- 格子量子モデルでは、世界線の更新手法として2つのメジャーな手法がある
- **ループアルゴリズム (LA)** H.G.Evertz, G. Lana, and M. Marcu, PRL **70**, 875 (1993)
  - 適用範囲 (系) が狭い
    - 反転対称性がある時のみ
      - 縦磁場のないXXZ 模型、half-filling のハードコアボーズハバード
  - ちゃんと動くケースでは収束が早い (自己相関時間が短い)
- **向き付きループアルゴリズム (DLA)** O.F.Syljuåsen and A.W.Sandvik, PRE **66**, 046701(2002)
  - 適用範囲がLAより広い ↖ DSQSS で使うアルゴリズム
  - 反転対称性があってもなくても動く

# 負符号問題 (Sign problem)

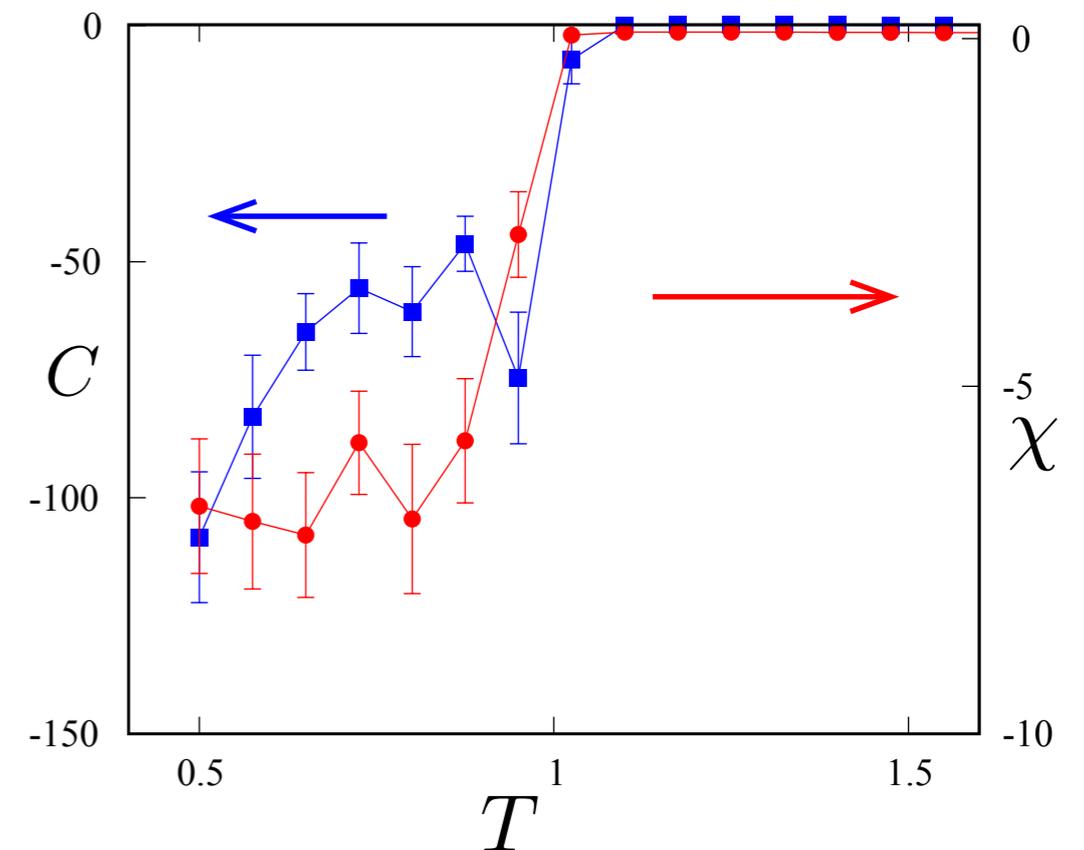
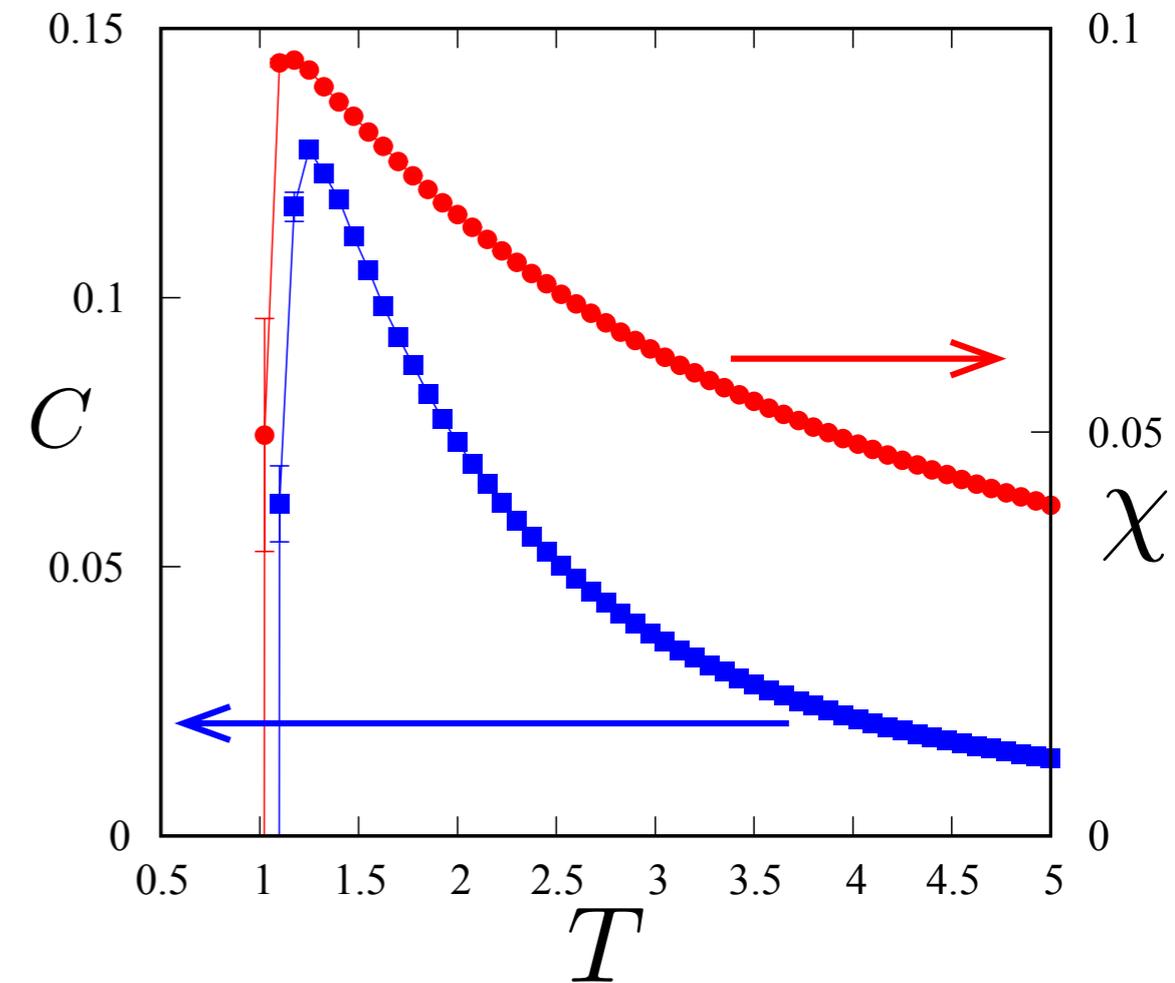
- PIMC などの量子モンテカルロ法における最大の障害
- 負や複素数の重みを持つ配位が生じうる
  - e.g., フラストレート反強磁性、フェルミオンハバード模型
  - 重みの絶対値を使ってMCMC を行い、符号や位相の情報は物理量とみなして期待値の計算で取り込む

$$\langle A \rangle = \frac{\sum_c A(c)W(c)}{\sum_c W(c)} = \frac{\sum_c A(c)S(c)|W(c)|}{\sum_c |W(c)|} \frac{\sum_c |W(c)|}{\sum_c S(c)|W(c)|} = \frac{\langle AS \rangle_+}{\langle S \rangle_+}$$

- 相対誤差が逆温度と体積に対して指数関数的に大きくなってしまいう
  - 実際の計算ではしばしば誤差の評価自体に失敗する

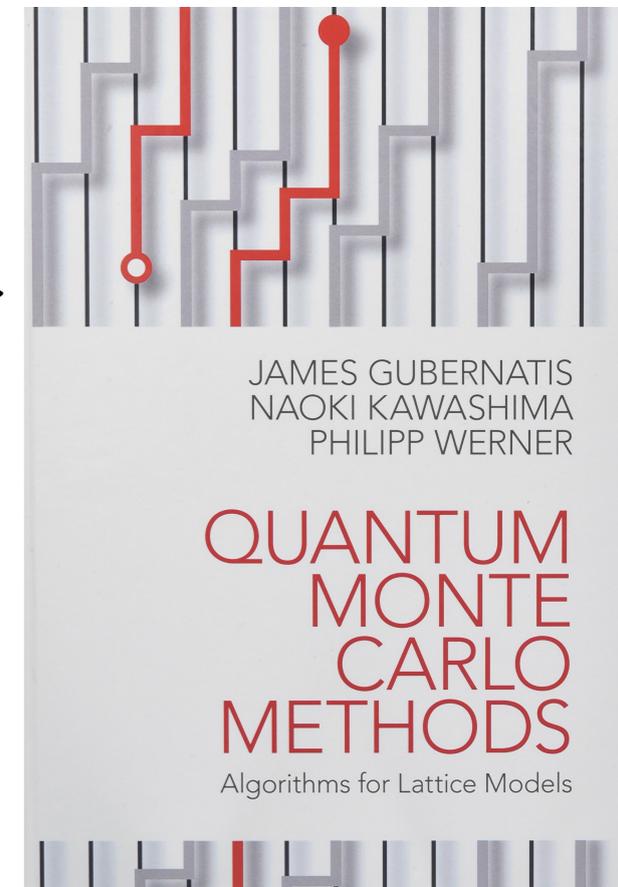
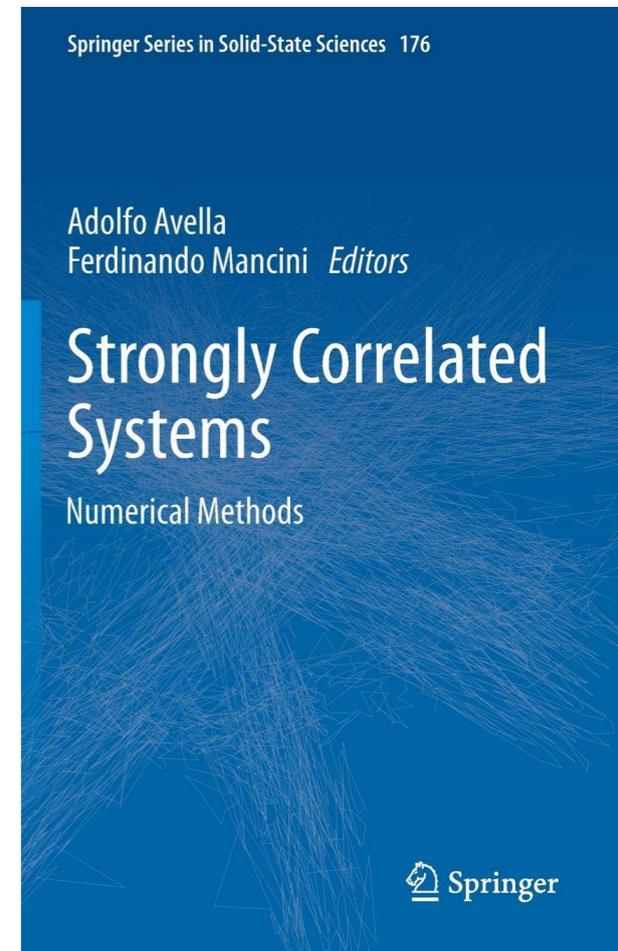
# 負符号問題

- $S=1/2$ カゴメ格子反強磁性ハイゼンベルグ模型
  - 結合定数  $J = 1$
  - スピン数  $N = 108 = 6 \times 6 \times 3$
- 青が比熱、赤が帯磁率
  - $T/J < 1$  ではまともに計算ができていない
  - $T/J > 1$  ではあまり問題はない
    - 実験のフィッティングなどには使える



# PIMC の参考図書

- S. Todo (藤堂真治) “Loop Algorithm”
  - Chapter 6 in “Strongly Correlated Systems: Numerical Methods”
    - Adolfo Avella and Ferdinando Mancini (Ed.)
    - Springer-Verlag Berlin (2013)
- R. G. Melko “Stochastic Series Expansion Quantum Monte Carlo”
  - Chapter 7 in “Strongly Correlated Systems: Numerical Methods”
    - Adolfo Avella and Ferdinando Mancini (Ed.)
    - Springer-Verlag Berlin (2013)
- J. Gubernatis, N. Kawashima (川島直輝), and P. Werner “Quantum Monte Carlo Methods: Algorithms for Lattice Models”
  - Cambridge University Press (2016)
- 原田健自 「量子モンテカルロ法の最近の発展」
  - 岩波書店 応用数理 VOL.15 NO.3 SEP. 2005 page 28-44



# Outline

- 経路積分モンテカルロ法について
- DSQSS (Discrete Space Quantum Systems Solver) について
- DSQSS の高度化について
- まとめ・展望
- 付録

このスライドは DSQSS のsite (<https://issp-center-dev.github.io/dsqss>) にアップロードします

# DSQSS (Discrete Space Quantum Systems Solver)

<https://issp-center-dev.github.io/dsqss/>

- 量子格子模型に対する経路積分モンテカルロ法ソルバーパッケージ
- DSQSS/DLA と DSQSS/PMWA からなる
  - DSQSS/DLA  今日のメイン
    - DLA (on-the-fly DLA) を実装したプログラム
    - 任意の格子・グラフの上の任意の模型について、QMC 計算を実行可能
      - いくつかの格子、模型は定義済み
      - 負符号が強いときには実行結果はかなり悪くなる (実行はできる)
  - DSQSS/PMWA
    - 大規模並列マルチワームアルゴリズム PMWA を実装したプログラム
      - 時空間をドメイン分割することで多数のワームを並列して動かす
      - ソース項 (e.g., 横磁場項) がゼロの極限を取る
      - A. Masaki-Kato *et al.*, PRL **112**, 140603(2014)
    - 現在は超立方格子上の  $S=1/2$  XXZ 模型 (= ハードコアボーズハバード模型) のみ計算可能

# DSQSS (Discrete Space Quantum Systems Solver)

<https://issp-center-dev.github.io/dsqss/>

- クレジット (2019-04-03 現在)

- 開発者 (五十音順)

太字：2018年度高度化Pj 関係者

- 加藤康之 (東京大学工学系研究科)
- **川島直輝 (東京大学物性研究所)**
- 坂倉耕太 (NEC)
- 鈴木隆史 (兵庫県立大学工学研究科)
- 原田健自 (京都大学情報学研究科)
- 正木晶子 (理研→日立研究所)
- **本山裕一 (東京大学物性研究所)**
- **吉見一慶 (東京大学物性研究所)**

- 協力者

- 大久保毅 (東京大学理学系研究科)
- **加藤岳生 (東京大学物性研究所)**

# DSQSS/DLA の計算例

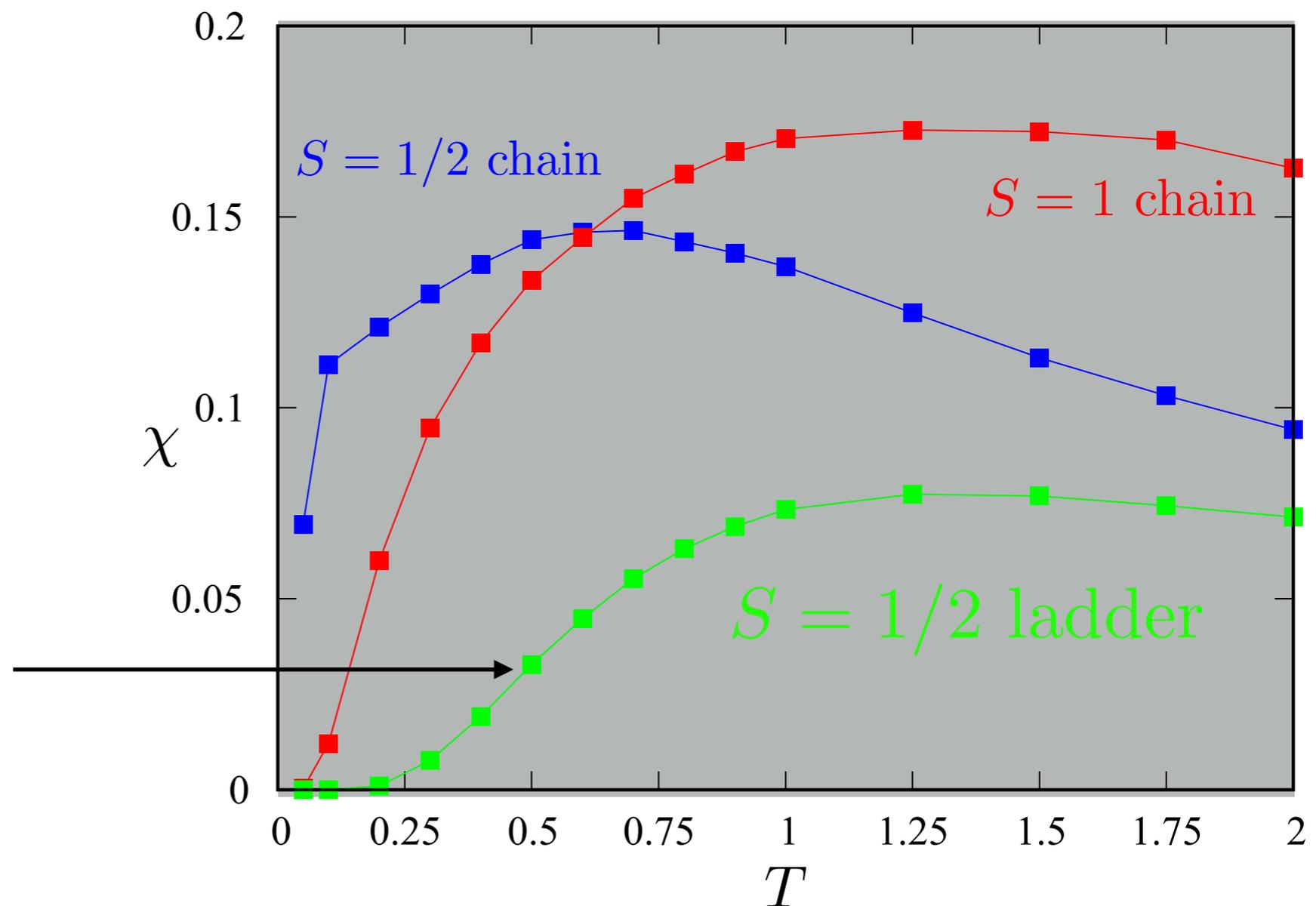
- 一次元反強磁性ハイゼンベルグ模型の帯磁率計算
  - $S=1/2$  chain (gapless),  $S=1$  chain (gapped),  $S=1/2$  ladder (gapped)

入力ファイル std.toml

```
[parameter]
beta = 2.0 ## inv.temp.
```

```
[hamiltonian]
model = 'spin'
M = 1 ## M = 2S
Jz = -1 ## antiferro
Jxy = -1 ## antiferro
```

```
[lattice]
lattice = 'hypercubic'
dim = 2
L = [32, 2]
bc = [true, false]
    #periodic, open
```



# DSQSS/DLA の計算例

S. Wessel and M. Troyer, PRL **95**, 127205 (2005)

- 三角格子ハードコアボーズハバード模型

入力ファイル std.toml

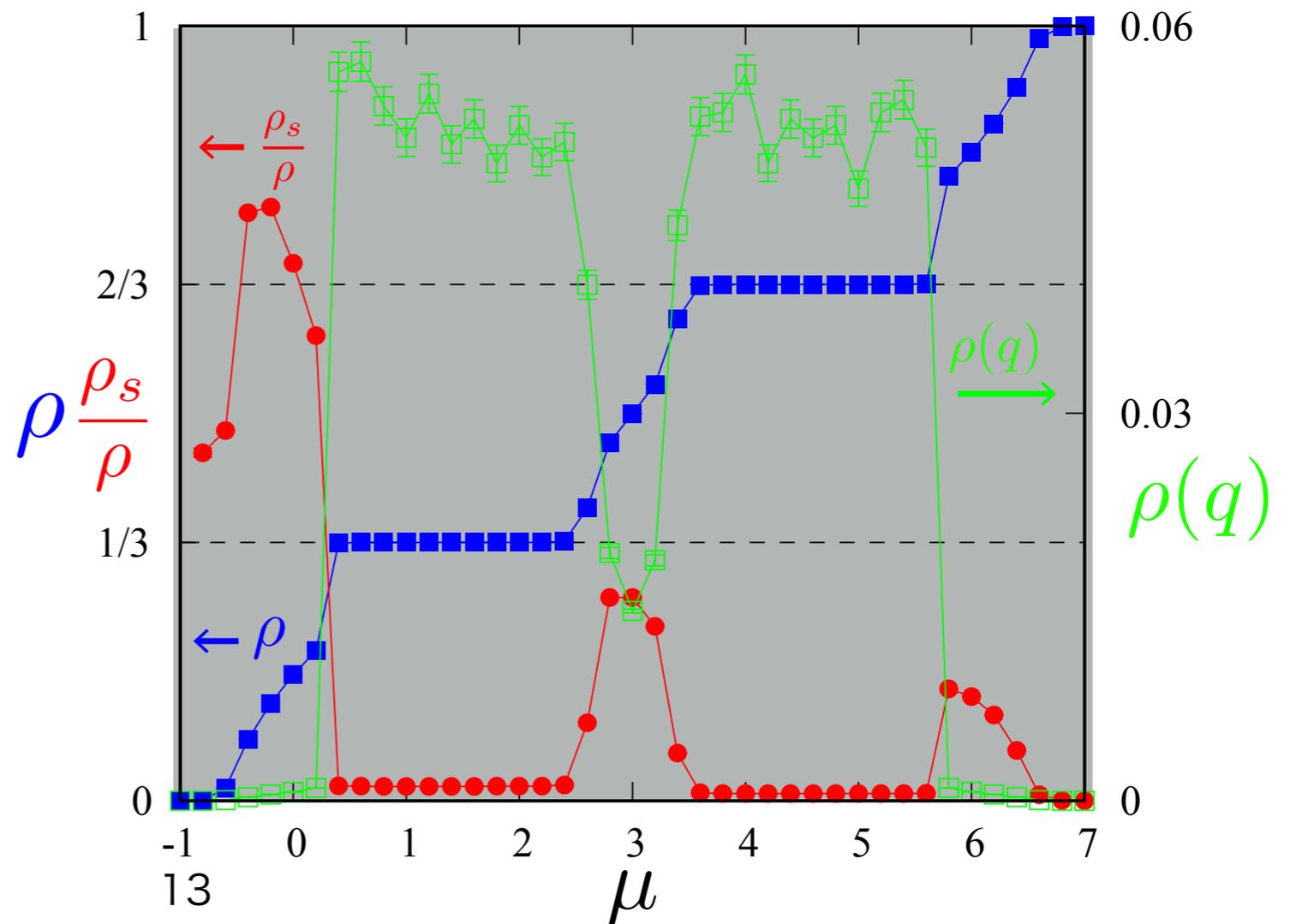
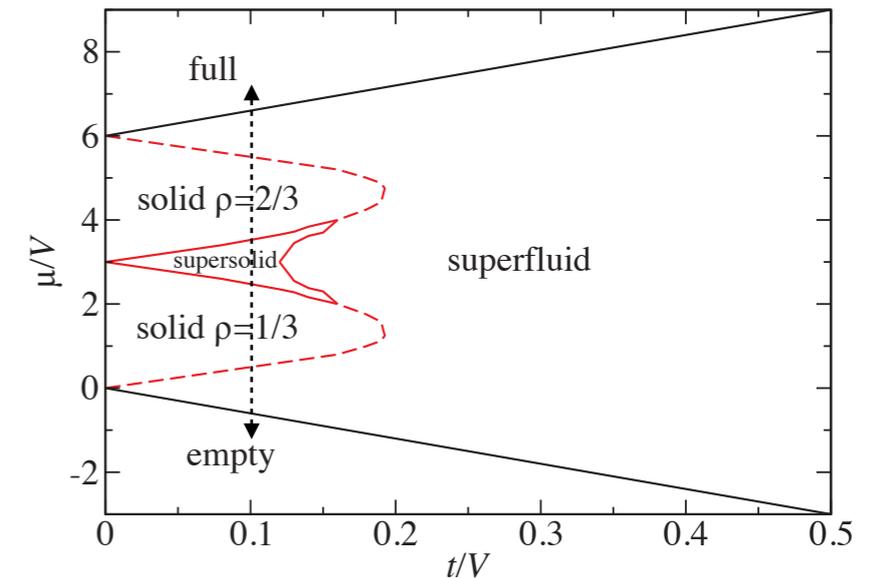
```
[parameter]
beta = 20.0 ## inv.temp.
```

```
[hamiltonian]
model = 'boson'
M = 1 ## number cutoff
t = 0.1
V = 1.0
mu = 0.0
```

```
[lattice]
lattice = 'triangular'
L = 12 ## 12x12
```

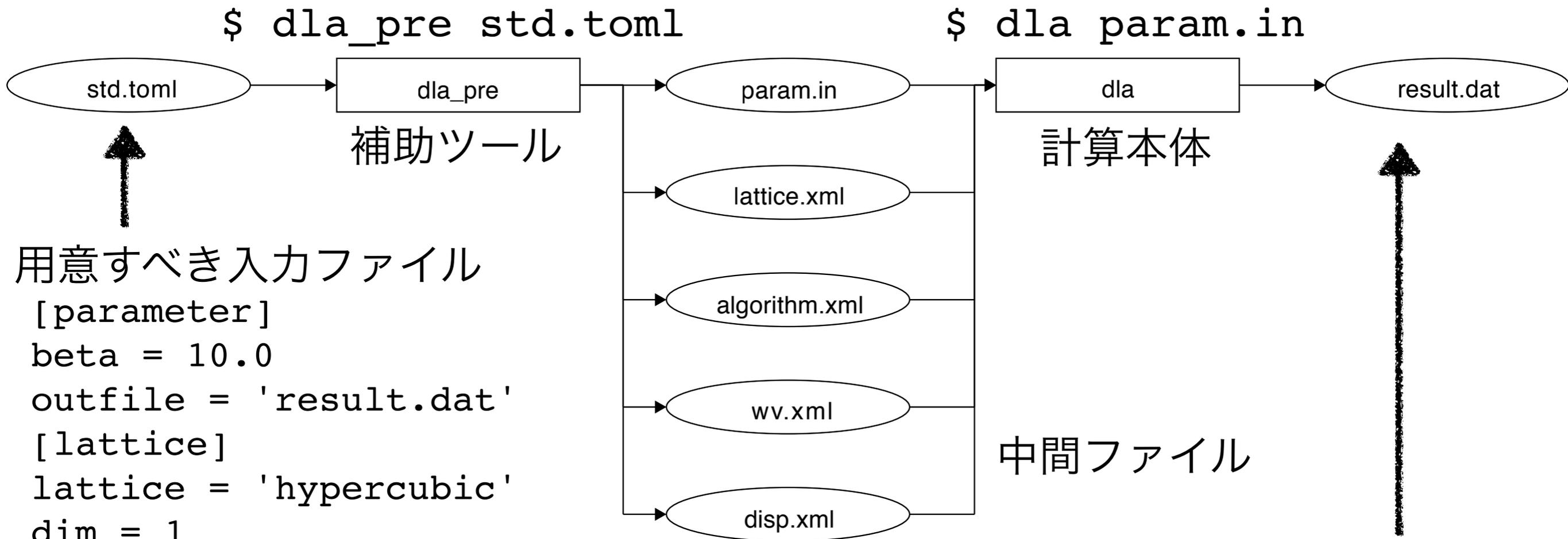
```
[kpoints]
ksteps = 1
```

q は 1/3, 2/3 固体の波数



# DSQSS/DLA のシンプルモード

- 定義済みの格子、モデルを計算したい時の計算フロー
  - 例：S=1/2 反強磁性ハイゼンベルグ鎖 (L=16,  $\beta = 10$ )



用意すべき入力ファイル

```
[parameter]
beta = 10.0
outfile = 'result.dat'
[lattice]
lattice = 'hypercubic'
dim = 1
L = 16
[hamiltonian]
model = 'spin'
M = 1      ## 2S=1
Jz = -1    ## antiferro
Jxy = -1   ## antiferro
```

```
...
R sign = 1.000000000e+00 0.000000000e+00
R anv = 2.95061250e+00 5.24981647e-03
R ene = -4.42679771e-01 6.18129258e-04
R spe = 1.48677303e-01 5.04906705e-02
...
```

最終結果

# DSQSS/DLA のシンプルモード

- 簡潔なTOML ファイル (シンプル入力ファイル) から dla\_pre というツールで dla の入力ファイルを作成する
- いくつかの格子や模型が定義済み
  - 格子 (境界条件を軸ごとに周期的・開条件で切替可能)
    - 超立方格子、三角格子、蜂の巣格子、カゴメ格子
  - 模型
    - 任意スピンのXXZ 模型、ソフトコアボーズハバード模型 (粒子数カットオフ付き)

$$\mathcal{H} = - \sum_{\langle ij \rangle} \left[ J_{ij}^z S_i^z S_j^z + \frac{J_{ij}^{xy}}{2} (S_i^+ S_j^- + S_i^- S_j^+) \right] + \sum_i \left[ D_i (S_i^z)^2 - h_i S_i^z \right]$$

$$\mathcal{H} = \sum_{\langle ij \rangle} \left[ -t_{ij} (b_i^\dagger b_j + h.c.) + V_{ij} n_i n_j \right] + \sum_i \left[ U_i \frac{n_i(n_i - 1)}{2} - \mu_i n_i \right]$$

# TOML

<https://github.com/toml-lang/toml>

- TOML (Tom's Obvious, Minimal Language)

- 設定ファイル向けのファイルフォーマット
- シンプルながら高機能
  - 人間が読み書きしやすい
- 多くの言語に入出力実装がある
  - 計算機が読み書きしやすい

```
[parameter]
beta = 10.0
outfile = 'result.dat'
[lattice]
lattice = 'hypercubic'
dim = 1
L = 16
[hamiltonian]
model = 'spin'
M = 1
Jz = -1
Jxy = -1
```

## 例：Python

```
In [1]: import toml
```

```
In [2]: d = toml.load('std.toml'); d
```

```
Out[2]:
```

```
{'parameter': {'beta': 10.0, 'outfile': 'result.dat'},
 'lattice': {'lattice': 'hypercubic', 'dim': 1, 'L': 16},
 'hamiltonian': {'model': 'spin', 'M': 1, 'Jz': -1, 'Jxy': -1}}
```

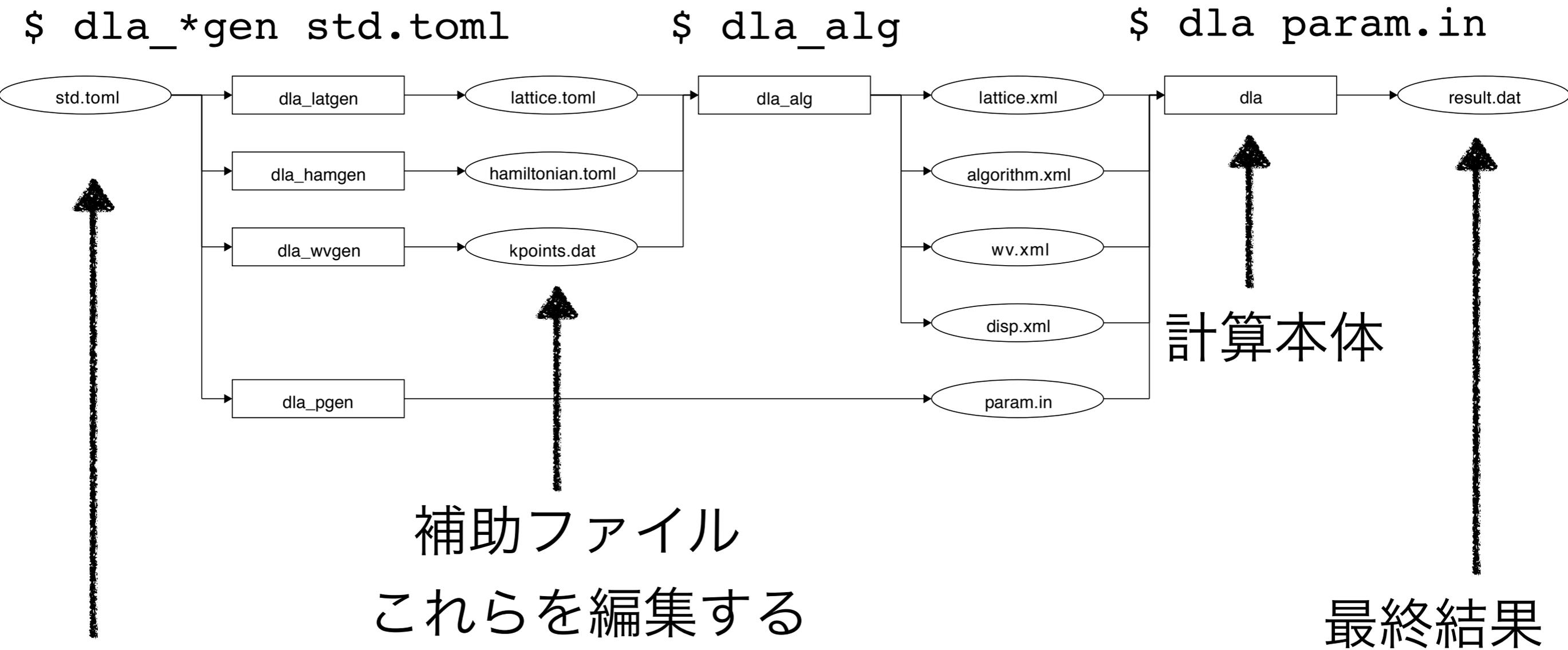
辞書（連想配列）が得られる

# DSQSS/DLA の物理量

- DSQSS/DLA で計算できる物理量
  - エネルギー、比熱、一様磁化、磁化（波数成分）、一様帯磁率、帯磁率（波数成分）、構造因子、構造因子（波数成分）、虚時間動的構造因子、虚時間グリーン関数、ヘリシティモデュラス
    - ボーズ系の場合、密度や超流動密度に適宜置き換える
- 計算結果はテキストファイルに出力される
- 例：反強磁性ハイゼンベルグダイマー  $\beta J=100$  のエネルギー密度と比熱
  - `R ene = -3.74436000e-01 5.64297400e-04`
  - `R spe = -3.92796000e-02 3.27280361e-01`
  - 数字はそれぞれ期待値と統計誤差
  - 先頭の R は Result の意味

# DSQSS/DLA のスタンダードモード

- ・ シンプルモードに用意されていない格子、模型を計算したい時の計算フロー

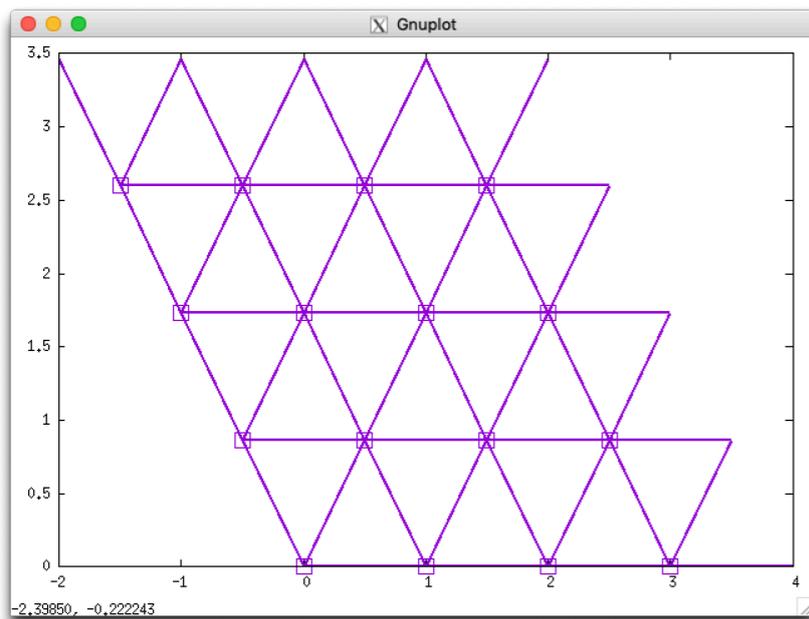


シンプルモードと同じファイル

# DSQSS/DLA のスタンダードモード

- 格子TOML ファイル

- ユニットセルと並進ベクトルを用いて格子を定義する
- 右は三角格子を表すファイル (抜粋)
- 下図を表示するgnuplot ファイルを出力する機能もある



```
[parameter] # size や境界条件、並進ベクトル
name = "triangular"
dim = 2
L = [ 4, 4, ]
bc = [ true, true, ]
basis = [ [ 1.0, 0.0, ],
          [ -0.5, 0.8660254037844387, ], ]
```

```
[unitcell] # ユニットセル
```

```
[[unitcell.sites]] # サイト情報
```

```
siteid = 0
```

```
type = 0
```

```
coord = [ 0.0, 0.0, ]
```

```
[[unitcell.bonds]] # ボンド情報
```

```
bondid = 0
```

```
type = 0
```

```
[unitcell.bonds.source]
```

```
siteid = 0
```

```
[unitcell.bonds.target]
```

```
siteid = 0
```

```
offset = [ 1, 0, ] # 隣のユニットセル
```

```
... 略 (ボンド2本) ...
```

# DSQSS/DLA のスタンダードモード

- ハミルトニアンTOML ファイル

- 局所ハミルトニアンの行列要素を指定する  $\langle 1, 0 | \mathcal{H}_0 | 0, 1 \rangle = -0.1$

- 右は実際のファイルからの抜粋

- 波数ファイル

- 波数ベクトルの要素を指定する

- 基底は格子の並進ベクトルの逆ベクトル

- 下は二次元系での一例

```
dim
2
```

```
kpoints
```

```
4
0 0 0
1 0 2
2 2 0
3 2 2
```

←  $\vec{k}_2 = 2\vec{g}_1 + 0\vec{g}_2$

```
...
[[interactions]]
id = 0
nbody = 2
N = [ 2, 2, ]
[[interactions.elements]]
istate = [ 0, 1, ] # 始状態
fstate = [ 1, 0, ] # 終状態
value = -0.1 # 行列要素

[[interactions.elements]]
...
```



# DSQSS/DLA のその他機能

- **セーブ&ロード**

- 指定秒数経過後に、チェックポイントファイルに情報を保存して終了することができる
- チェックポイントファイルから読み込んで計算を再開できる
- スパコンなどの共用計算機で効果的

- **乱数並列**

- MPI 実行すると、プロセス数だけ乱数並列を行う
- 統計誤差が、プロセス数の平方根の逆数になる
- 計算例で載せた2つの計算は物性研スパコンシステムC “Enaga” を利用している
  - ノード内40コアあるのでそれだけで誤差がだいたい1/6になる
    - 同じ精度なら計算時間が1/40
  - ボソンの計算は4ノード使ったので160 コア、だいたい 1/13

# Outline

- 経路積分モンテカルロ法について
- DSQSS (Discrete Space Quantum Systems Solver) について
- **DSQSS の高度化について**
- まとめ・展望
- 付録

このスライドは DSQSS のsite (<https://issp-center-dev.github.io/dsqss>) にアップロードします

# DSQSS の高度化について

- DSQSS ver1.2 および ver2 は2018年度東京大学物性研究所ソフトウェア開発・高度化プロジェクトの支援を受けた
  - インストール方法の整備
    - CMake を導入した
  - 入出力の整備
    - 補助ファイルの導入
      - 読み書きのしやすいフォーマットとしてTOML を採用した
    - 入力ファイル生成ツールの整備
      - Python で書き直し
    - スピン系ソルバとボーズ系ソルバの統合
      - 物理量定義の違いは入力パラメータで吸収
  - テストの導入
    - GitHub + Travis CI による自動テストも導入
  - 文書化
    - Sphinx を導入することで HTML と PDF を単一ソースから作るようにした

# まとめ・今後の展開

- 格子量子模型の経路積分モンテカルロ(PIMC)法ソルバー DSQSS を紹介した
  - DSQSS は on-the-fly 向き付きループアルゴリズム を実装した DSQSS/DLA と 並列マルチワームアルゴリズム を実装した DSQSS/PMWA を含む
  - DSQSS/DLA は任意の模型・格子のPIMC計算が実行できる
    - 負符号問題が起きると統計誤差がひどくなる
    - 模型や格子の表現力の柔軟性とファイル形式の複雑さに応じて3レベルの入力ファイルがある
      - シンプルモード、スタンダードモード、エキスパートモード
- 高度化で導入したことを紹介した
  - CMake, Python, テスト, Sphinx, Travis CI, …
- 今後の展開
  - MateriAppsLIVE! や物性研スパコンへのプレインスツール
  - ソフトウェア論文 (Comp. Phys. Comm. を予定)
  - ソフトウェア講習会 (少なくとも2019年度中に一回は物性研で開催する予定)

# Outline

- 経路積分モンテカルロ法について
- DSQSS (Discrete Space Quantum Systems Solver) について
- DSQSS の高度化について
- まとめ・展望
- **付録**

このスライドは DSQSS のsite (<https://issp-center-dev.github.io/dsqss>) にアップロードします

# 重み付き平均

- 次に示すような重み付き平均値を計算したい (W: 重み)

$$\langle A \rangle = \frac{\sum_c A(c)W(c)}{\sum_c W(c)}$$

- 統計物理では、カノニカル平均は重みがボルツマン重みの重み付き平均

$$W(c) = \exp(-\beta E(c))$$

- c は 配位 (configuration)、 $\beta$  は逆温度、Eはエネルギー
- 配位として、たとえば格子点上に並んだスピンの列 (組み合わせ)
- 多くの場合において、取りうる配位の総数は莫大なものとなる
  - イジング模型ではスピン数 N に対して  $2^N$  と指数関数的に増大
  - 単純な足し上げでは計算不可能!

# マルコフ連鎖モンテカルロ法 (MCMC)

- 重み付き平均値を統計誤差の範囲内で厳密に計算する数値手法

$$\langle A \rangle = \sum_c A(c)W(c) / \sum_c W(c)$$

- 出現頻度が重み $W(c)$  に従うような配位列  $\{c_i\}$  を生成し、 $A(c)$ の平均を取る

- 今の配位から次の配位を確率的に生成していく (マルコフ連鎖)

$$c_1 \rightarrow c_2 \rightarrow \dots \rightarrow c_n \rightarrow \dots$$

- 次の配位を作るための確率 (遷移確率)  $p$  をうまく調整することで、任意の重み $W$  を表現できる

- 釣り合い条件  $W(c') = \sum_c W(c)p(c \rightarrow c')$

- 詳細釣り合い条件  $W(c')p(c' \xrightarrow{c} c) = W(c)p(c \rightarrow c')$

- これらの条件を満たすような  $p$  を作るアルゴリズム

- メトロポリス法、熱浴法、諏訪・藤堂法

- 他にエルゴード性と非周期性が必要

# マルコフ連鎖モンテカルロ法 (MCMC)

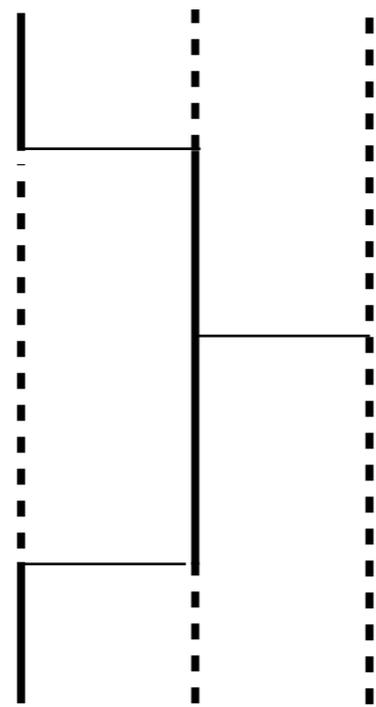
- MCMC では今の配位から次の配位を確率的に生成していく

$$c_1 \rightarrow c_2 \rightarrow \dots \rightarrow c_n \rightarrow \dots$$

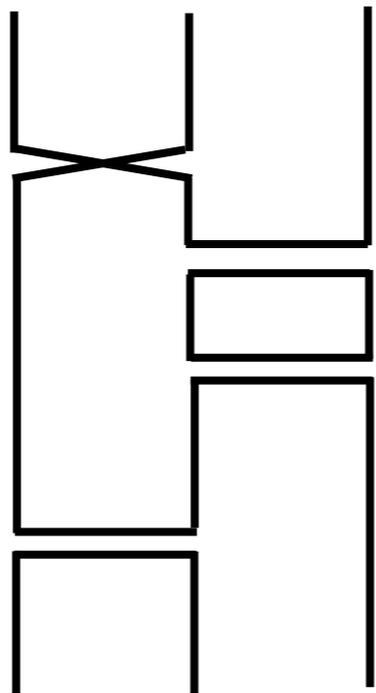
- 互いに相関がある (似ている)
- 相関が切れる (情報を忘れる) までに必要なステップ数が自己相関時間
- 配位を大きく変えると相関が切れやすいが、大きな変化はたいてい受理確率が低い
  - クラスターアルゴリズム：相関のある領域 (クラスター) に分割してまとめて更新する手法の総称
    - Swendseng-Wang アルゴリズム、Wolff アルゴリズム
    - ループアルゴリズム、向き付きループアルゴリズム

# ループアルゴリズム (LA)

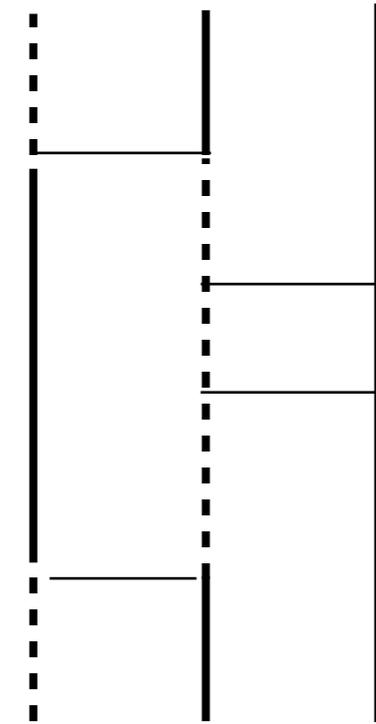
- スピン配位 (もともとの配位) の他に補助的なループ配位を考えて空間を広げる
- Swendsen-Wang アルゴリズムのPIMC 版



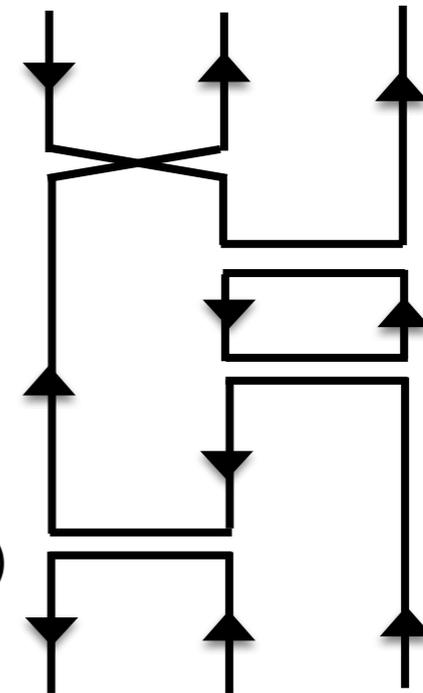
↓ ループ要素の挿入



→ ループ着色  
(ループフリップ)

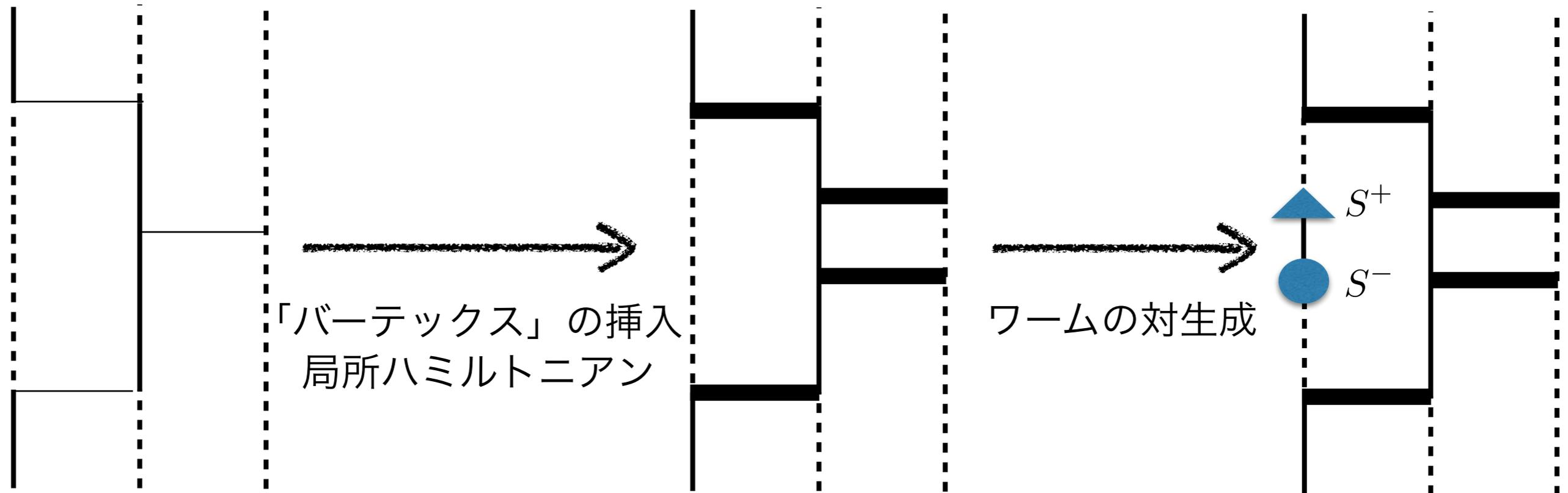


↑ スピンにもどす



# 向き付きループアルゴリズム (DLA)

O.F.Syljuåsen and A.W.Sandvik, PRE **66**, 046701 (2002)

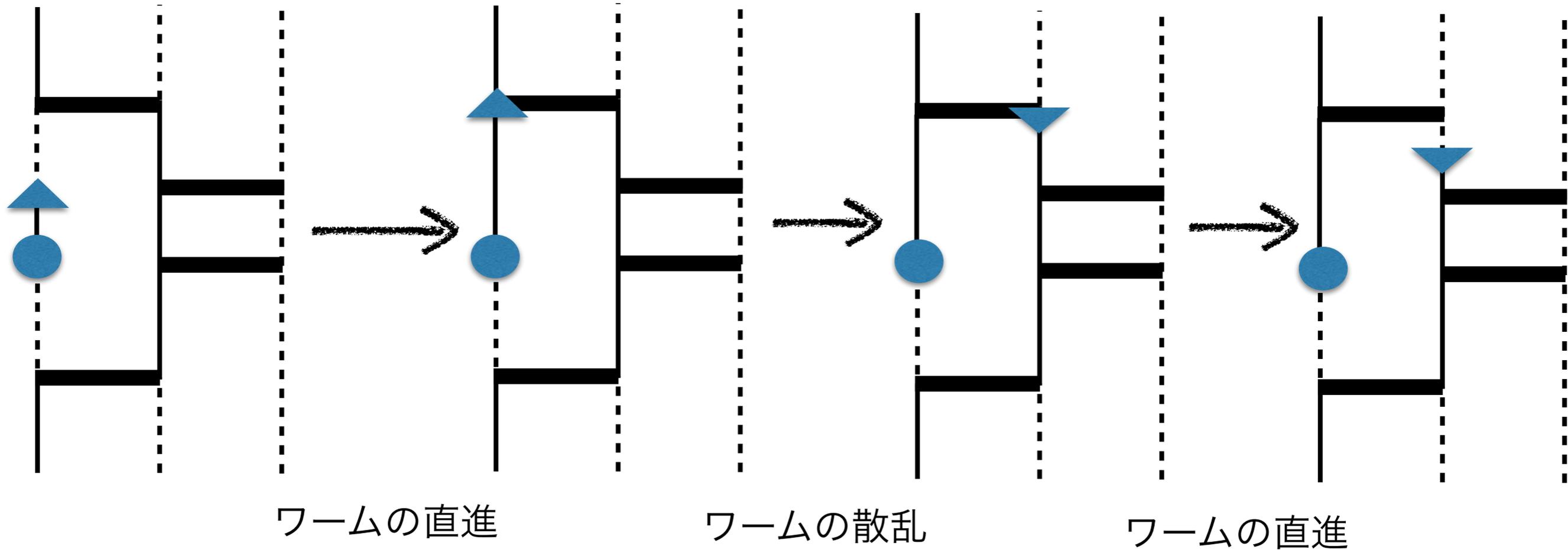


- ワームは生成消滅演算子
  - ワームの前後で状態が変わるので、ワームを動かすことで世界線全体が変わる
- ワームを導入するためにワームソース項（生成消滅演算子ひとつの項）をハミルトニアンに追加する
  - ワームがない時のみ物理量を測定する

例： $\eta S_i^x$

# 向き付きループアルゴリズム (DLA)

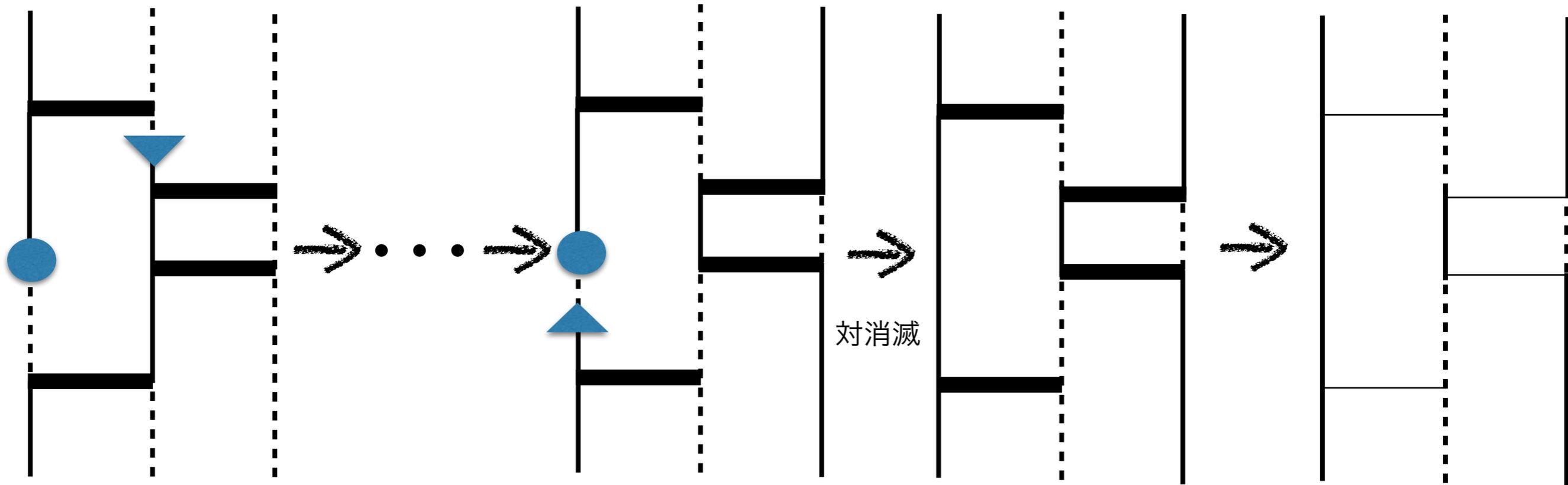
O.F.Syljuåsen and A.W.Sandvik, PRE **66**, 046701 (2002)



- ワームはバーテックスに当たると「散乱」される
- 散乱確率は（詳細）釣り合い条件を満たすように決められる
  - 例：諏訪・藤堂法

# 向き付きループアルゴリズム (DLA)

O.F.Syljuåsen and A.W.Sandvik, PRE **66**, 046701 (2002)



- ワームが戻ってきて対消滅するまで更新をする
- バーテックスをあらかじめバラ撒かずに、ワームを動かすのにあわせて生成するアルゴリズムもある (**on-the-fly DLA**)
- Y. Kato, T. Suzuki, and N. Kawashima, PRE **75**, 066703 (2007)
- 相互作用が強くてバーテックス密度がとても大きいときに有効

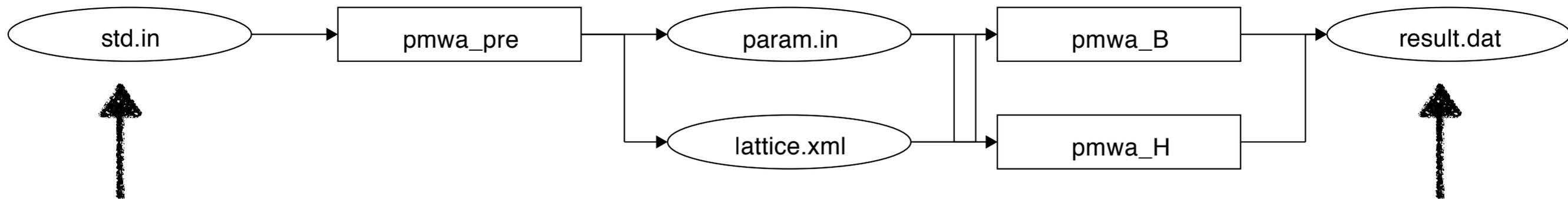
# DSQSS/DLA の入力ファイル

- DSQSS/DLA の入力ファイル (plaintext 1つ、XML 4つ)
  - ほとんどの場合、これらのファイルをユーザが直接編集する必要はない
    - エキスパート向け
  - パラメータファイル
    - 逆温度、MCサンプル数、入出力ファイル名などの計算パラメータを記すテキストファイル
  - 格子XML ファイル
    - 格子点のつながりなどを記すXML ファイル
  - アルゴリズムXML ファイル
    - ワームのバーテックスでの散乱確率などを記すXML ファイル
  - 波数XML ファイル
    - 構造因子などの計算に使う波数点を記すXML ファイル
    - 正確には波数と格子座標との内積を定義する
  - 相対座標 XML ファイル
    - 相関関数などの計算に使う、格子点对の相対座標を記すXML ファイル
    - 正確には同じ相対座標をもつ格子点对をグループ化する

# DSQSS/PMWA のワークフロー

\$ pmwa\_pre std.in

\$ pmwa\_H param.in



```
[System]
solver = PMWA
[Hamiltonian]
model_type = spin
Jxy = -1.0
Jz = -1.0
Gamma = 0.1    ## ソース項の係数

R ene = -0.5614703125 0.004336923499001907
R spe = -0.1126392874999807 0.2421583310628812

[Lattice]
lattice_type = square
D = 1
L = 8
Beta = 8.0
[Parameter]
outfile = result.dat
```

# DSQSS の高度化:入力の整備

- 直接の入力である XML ファイルフォーマットは極力いじらないようにした
- 入力ファイルジェネレータを一から書き直した
  - Python製
  - スクリプトが複数あるため、dsqss パッケージを作成して使う
  - 外部パッケージを適宜利用
    - Numpy, Scipy, TOMLパーサ
- XML はユーザが自分で書くには冗長なので、もう少し読み書きしやすいフォーマットとして TOMLを導入した
  - 読み書きしやすい
  - 仕様がだいぶ固まっている
  - 実績もそこそこある
  - 様々な言語で読み書き用のライブラリ実装がある

# DSQSS の高度化:インストール方法の整備

- CMake を導入した
  - 私見ではMakefile をユーザが直接編集するのは最終手段だと思っ  
てます
- Python ツールのインストールについて
  - `python (fullpath/)dla_pre.py` ではなく、パスを通して `dla_pre` と打つだけ  
にしたい
    - 実行ファイルのインストールディレクトリはdla 本体の実行ファイルと同じ場所  
にしたい
  - 共用計算機 (スパコン) で、あるユーザが他のユーザのために共有領域に  
インストールすることを考慮したい
    - ディレクトリなどの環境をあまり汚したくない
  - 意外と各種 PATH周りで考えることが多い
    - 読み込むべきdsqss パッケージはどこにあるのか?
    - スクリプトを起動するためのインタプリタのパスは?
    - 依存パッケージ(Numpy)などをどう解決するか?
- CMake と pip (setuptools) を駆使してできるだけ自動化した
  - 詳しくは実際のコードを参照

# DSQSS の高度化:テストの整備

- テストを導入した
  - ツールの出力が予想したものとなっているか
  - QMC 計算の結果が厳密対角化のものと整合が取れているか
    - 「ずれているとは言えない」ぐらいしか言えない
- GitHub リポジトリにTravis CI による自動テストを導入した
  - master ブランチにマージする前にチェックできる
- 今後の課題
  - dla, pmwa 本体のユニットテスト？
    - ソースコードそのものを整理しないと難しい

# DSQSS の高度化:文書の整備

- Sphinx による文書化を行った
  - HTML ドキュメントとPDF ドキュメントとを同一の原稿から作成可能
- 今後の課題
  - web 上のドキュメントを手で更新（置換）しているので自動化したい
    - Travis CI を利用した自動デプロイ
    - やりかたを整理しておくると他の高度化プロジェクトでも使える

# DSQSS の高度化:並列化

- DSQSS/DLA の並列化は乱数並列のみ
- パラメータ並列は未実装
  - 一組の入力を受け取って計算して、一組の出力を返す、という構造は DSQSS/DLA だけではない
  - 外側で用意するべき？
    - xargs, GNU Parallel, OACIS など
    - スパコンの場合、ジョブスケジューラとの兼ね合いがあって実は難しい
      - 適当にプロセスを作るとジョブのクリーンアップ処理と競合することがあるらしい
      - ツールとしてはベンダーが作るべき???
      - バルクジョブ@ISSP SC