

ALPS チュートリアル – ALPS ライブラリ

CMSI 神戸ハンズオン

ALPS Collaboration
<http://alps.comp-phys.org/>

ALPS

- 1 全体構成
- 2 ALPS/parameter
- 3 ALPS/alea
- 4 ALPS/lattice
- 5 ALPS/model

ALPS の階層構造

tools	XML manipulation Python binding GUI	
applications	MC QMC ED DMRG DMFT	
domain-specific libraries	lattice model observables scheduler	
numerics	random ublas Boost library	iterative eigenvalue solver
generic C++	graph	serialization XML/XSLT
C / Fortran	BLAS LAPACK MPI HDF5	

サードパーティーのライブラリ

- BLAS, LAPACK: 線形演算 (行列対角化、特異値分解など)
- MPI (Message Passing Interface): 並列計算のためのメッセージ通信
- HDF5 (Hierarchical Data Format): プラットフォーム非依存のバイナリファイル格納形式
- Boost C++ Library: 乱数, グラフ, シリアル化, など多くの有用なライブラリ群

ALPS/parameter ライブラリ

- パラメータの入出力のためのライブラリ
- 改行, セミコロン, コンマで変数を区別
- 四則演算、初等関数 (sin, cos, exp など) が使える
- π (PI), 虚数単位 (I)などを文字で指定
- C 風, C++風のコメント
- { } で囲んだ変数は異なるセット

```
LATTICE = "chain_lattice";  
L = 16,  
SEED = 2873  
// C++ style comment  
SWEEPS = 4096;  
THERMALIZATION = SWEEPS/8;  
/* C style comment */  
{ T = 2; Sq = 2*PI/3; }  
{ T = 1.8; }
```

ALPS/parameter を使ったコード例

```
#include <boost/foreach.hpp>
#include <alps/parameter.h>
int main() {
    std::ifstream fin;
    fin.open("parameters.txt");
    alps::ParameterList plist(fin);
    BOOST_FOREACH(alps::Parameters& p, plist) {
        double a = p["a"];
        double b = p.value_or_default("b", 0.5);
        ...
    }
    ...
}
```

ALPS/alea ライブラリ

■ マルコフ連鎖における平均値, 分散, 自己相関を計算するライブラリ

```
alps::RealObservable mag2("Magnetization^2");  
...  
mag2 << m * m; // in each MC step
```

■ ビンニング解析を用いた平均値, エラー, 自己相関時間の評価

```
std::cout << mag2 << std::endl;
```

■ ジャックナイフ法を用いた非線形量のエラー評価

```
alps::RealObsEvaluator binder = mag2 * mag2 / mag4;  
std::cout << binder.mean() << '□' << binder.error() << '\n';
```

ALPS/lattice ライブラリ

- 「格子構造」は数学的には「グラフ」で表現できる
 - $\text{site} \Leftrightarrow \text{vertex}$
 - $\text{bond} \Leftrightarrow \text{edge}$
- Boost Graph Library に対する「ラッパー」を提供
 - XML による「格子構造」の入出力
 - 「ユニットセル」による繰り返し構造の指定
 - 座標、パリティ、逆格子ベクトルなどの属性
- あらかじめ用意されている格子: "chain lattice", "square lattice", "triangular lattice", "honeycomb lattice", "simple cubic lattice", など

有限格子 + ユニットセルの埋め込み - 1

■ 格子の指定

```
<LATTICE name="chain lattice" dimension="1">  
  <BASIS><VECTOR> 1 </VECTOR></BASIS>  
</LATTICE>
```

■ ユニットセル

```
<UNITCELL name="simple1d" dimension="1" vertices="1">  
  <EDGE>  
    <SOURCE vertex="1" offset="0"/><TARGET vertex="1" offset="1"/>  
  </EDGE>  
</UNITCELL>
```

有限格子 + ユニットセルの埋め込み - 2

■ サイズ、境界条件の指定

```
<LATTICEGRAPH name = "chain lattice">  
  <FINITELATTICE>  
    <LATTICE ref="chain lattice"/>  
    <PARAMETER name="L"/>  
    <EXTENT size ="L"/>  
    <BOUNDARY type="periodic"/>  
  </FINITELATTICE>  
  <UNITCELL ref="simple1d"/>  
</LATTICEGRAPH>
```

■ より複雑な格子の作り方

■ ALPS Lattice HOWTO:

<http://alps.comp-phys.org/mediawiki/index.php/Tutorials:LatticeHOWTO>

ALPS/model ライブラリ

- XML を使ってハミルトニアンを定義する
 - 量子数や演算子の定義
 - シンボリックな表現を使って, ハミルトニアンのサイト項やボンド項を定義

```
Jz*Sz(i)*Sz(j)+Jxy/2*(Splus(i)*Sminus(j)+Sminus(i)*Splus(j))
```

- 作成した局所ハミルトニアンは行列の形で取り出せる
- ALPS Model HOWTO:
<http://alps.comp-phys.org/mediawiki/index.php/Tutorials:ModelHOWTO>
- あらかじめ用意されている模型: "spin", "boson Hubbard", "hardcore boson", "fermion Hubbard", "spinless fermions", "Kondo lattice", "t-J"