Manual of the analysisHAL_miya

Takaya Miyamoto
Yukawa Institute for Theoretical Physics, Kyoto University

2018 年 6 月 2 日

目次

はじめに

このマニュアルは未完成です。
適宜、項目を増やしていく予定です

1 コード全体の概要

本コードは、HAL QCD 法を用いた Lattice QCD potential の計算とその解析を目的として構築されています。使用する言語は C/C++ もしくは python3 で、基本的な設計として以下を想定しています。

基本設計

C/C++

Single hadron correlator 及び NBS 波動関数の入出力とポテンシャルの計算・出力のために用いる

python3

C/C++ で作られたポテンシャルの解析のために用いる

- (例 1) ポテンシャルのフィッティング
- (例 2) 位相差、T 行列等の観測量の計算

なお、C/C++ では外部ライブラリ（boost など）を使わないように設計してありますが、python3 では少なくとも Numpy と Scipy、できれば Matplotlib と npmath もインストールしていることを想定しています。

また、python3 コードはできるだけ python2 と互換性のあるように設計してありますが、確実に動くという保証はありません。

2 C/C++ コード

この章では、analysisHAL_miya}の中の C/C++ コードで実装されているクラス・関数等の詳細と使い方を解説します。基本的には、共通のルーチンをベースクラスやテンプレートクラスにまとめ、実際に用いるクラスはそれらを継承したものにしています。現在（Version 2）実装されているベースクラス・テンプレートクラスを以下に示します。

C/C++ コードで実装されているベースクラス・テンプレートクラス

- ComplexField_BASE
  複素数の場の量を扱うベースクラス
- STATISTICS<X>
  いくつかのサンプルについて、統計計算をするテンプレートクラス
- MATRIX_TEMPLATE_BASE<X>
  いくつかのサンプルについて、統計計算をするテンプレート（ベース）クラス

この中でも特に重要なのが複素数の場の量を扱うための「ComplexField_BASE」クラスであり、2.1 章にて詳しく解説します。また 2.2 章では、次に重要な「STATISTICS<X>」テンプレートクラスについて詳しく見ていきます。その他のベースクラスは、??章にて解説します。

2.1 ComplexField_BASE クラス

Single hadron correlator や NBS 波動関数などは、時空間やスピンなどの index を持つ複素場として定義されており、HAL QCD 法ではそれらの量を組み合わせることによりポテンシャルを計算しますが、この複素場を直感的に扱えるクラスとして「ComplexField_BASE」というベースクラスを用意しています。この章では、このベースクラスの実装と実際の使い方を解説します。

ComplexField_BASE クラスのメンバ変数は次のようなもので、全て protected になっています。

ComplexField_BASE クラスのメンバ変数

- std::complex<double> *m_field
  複素数の場を格納する配列のポインタ
- int m_xSIZE
  空間（x 方向）の大きさ
- int m_ySIZE
  空間（y 方向）の大きさ
- int m_zSIZE
  空間（z 方向）の大きさ
- int m_tSIZE
  時間（t 方向）の大きさ
- int m_aSIZE
  時空間の index の内側を回る自由度の大きさ
- int m_bSIZE
  時空間の index の外側を回る自由度の大きさ

以下では、ComplexField_BASE クラスのメンバ関数について解説していきます。

2.1.1 ComplexField_BASE クラスのインスタンス化とオブジェクトの初期化

ComplexField_BASE クラスのインスタンス化のために、次の 4 種類のコンストラクタを用意してあります。

(1) `ComplexField_BASE psi;`
(2) `ComplexField_BASE psi(psi_in);`
(3) `ComplexField_BASE psi(aSIZE, xSIZE, ySIZE, zSIZE, tSIZE, bSIZE);`
(4) `ComplexField_BASE psi(aSIZE, LSIZE, tSIZE, bSIZE);`

(1) では、全ての自由度の大きさがゼロのオブジェクト `psi` が作られます。(2) はコピーコンストラクタで、 引数で与えた `psi_in` と同じサイズ、同じ値を持つオブジェクトが作られます。

2.2 Statistics<X> テンプレートクラス

最終的なポテンシャルの結果は、異なるゲージ配位の下で計算された量の統計平均として定義されますが、この統計操作を直感的に扱えるクラスとして「Statistics<X>」というテンプレートクラスを用意しています。

3 python3 コード

3.1 概要

4 簡単な計算コード例と解析の手順

4.1 テストデータについて