

# LATTICEEASY(日本語訳)

Haruto Yokoyama

2021 年 3 月 17 日

# 目次

第 1 章	概要	2
第 2 章	記法と慣習	3
第 3 章	LATTICEEASY のファイル	4

# 第 1 章

## 概要

LATTICEASY は C 発展する宇宙においてスカラー場との相互作用の格子シミュレーションをする C++ のプログラムです。プログラムは簡単にパラメータを変えて動かすことができ、簡単に評価のために新しいモデルを挿入可能なようにデザインされています。LATTICEASY2.0 ではこれらのシミュレーションを 1-3 の次元で一つの変数を変えるだけで簡単に行うことができます。また、LATTICEASY の並列処理のバージョンとして CLUSTEREASY があります。

プログラムは Web サイトの <http://www.science.smith.edu/departments/Physics/fstaff/gfelder/latticeeasy> にて使用可能です。これは自由に誰でも使い、修正することが可能です。詳細は第 8 章:基本的には我々のクレジットと連絡先を明記していただければ自由に変更を加えていただいて問題ありません。

もし何らかの質問やコメントがありましたら、[gfelder@email.smith.edu](mailto:gfelder@email.smith.edu) までご連絡ください。我々はあなたにとってプログラムがどんな風に動き、あなたが何らかの問題を解決できれば光栄です。バグや改善点などプログラムに関してご連絡いただければ幸いです。

この資料は 4 つのメインセクションに分かれています。4 章は LATTICEASY の使い方について記載されており、どのようにプログラムをコンパイルし、走らせるか、そしてどのように与えられたモデルに対して適切なパラメータをセットするか、最終的にどのように新しいモデルをプログラムを走らせるために作るのが記載されています。5 章ではプログラムのアウトプットについて記載されています。そこにはアウトプットの関数と出力されるファイルに何が記載されているかを述べています。6 章ではプログラムで用いる式について記載されています。このセクションの大半はプログラムを使う上で必要はありません。しかし、どのようにしてプログラムが動いているのか、何が起きているのかを知る上では必要となってきます。このセクションで最も重要なパートは変数をプログラムでリスケールすることができることです。式を簡単にするためにプログラムは場や時空に対してリスケールされた値を用い、これらのリスケールについては 6 章にて説明がなされています。これらのリスケールは 4 章や 5 章を通して使われています。なのでそれらのセクションを読むときにより理解理解しやすくなるでしょう。7 章は LATTICEASY の並列処理について記載されています。

メインセクションに比べてマイナーなセクションがいくつかあります。2 章は記法と慣習について、3 章は LATTICEASY ファイルのリストとそれぞれが何をしているかについて説明しています。8 章はプログラムの使用方法について。この"クレジット"セクションは `latticeeasy.cpp` の一番上に記載してください。

## 第 2 章

# 記法と慣習

それぞれの値はプランク単位である  $M_p \approx 1.22 \times 10^{19}$  GeV で与えられています。FRW 計量は  $g_{\mu\nu} = \text{diag}\{1, -a^2, -a^2, -a^2\}$  となり、これに対する時間は物理単位でかかれ、距離は共動座標で測られています。プログラムでリスケールされた変数は下付き文字で  $pr$  がつけられています。また、繰り返しの変数は特に明記せず和が取られます。上付きドットは実時間での微分を表し、プライム記号はプログラム時間  $t_{pr}$  での微分を表します。

場の一般的な特性は、フィールド変数  $f$  を用いて示されます。 $f_i$  のような下付き文字は必要に応じて用いられます。 $F_k$  と  $f_k$  はそれぞれ連続と離散の  $f$  のフーリエ変換を指します。これらに関しては 6.3 にて議論します。

スケールファクターと時間は利便性のため、始め 1 と 0 にセットされています。

この文書は LATTICEEASY に含まれるファイルや、プログラムの変数や関数を参照していることが多くあります。これらのファイル名は `latticeeasy.cpp` のようにタイプライター体で書かれます。プログラムによって作成されるアウトプットファイルも同じ方法で記載されます。プログラム中の関数や変数もタイプライターフォントですが、*nflds* や *potentialenergy()* のように斜体になっています。(ただし、プログラム変数が数式の中に出てくる場合は、読みやすいように通常の数式の形式で表示されます。) 関数は常に括弧で表示されますが、関数の引数は通常含まれません。

## 第 3 章

# LATTICEEASY のファイル

LATTICEEASY は以下のファイルから構成されます。

- latticeeasy.cpp** `latticeeasy` の `main()` 関数とグローバル変数の宣言が含まれています。(すべて `latticeeasy.h` で `extern` として宣言されています。)
- initialize.cpp** 初期化に関することをする `initialize()` 関数を含みます。フィールドの揺らぎの初期条件の設定、スケールファクターの微分値の初期値の設定、各種パラメーターの初期化などがあります。
- evolution.cpp** 場やその微分値やスケールファクターやその微分値の増幅に関する関数が含まれています。またこのファイルは場のエネルギー勾配をも計算する関数が含まれています。
- output.cpp** 与えられた量を計算し保存する関数が含まれています。これらの量には場の平均値、分散、スケールファクターとその微分値、エネルギー密度、スペクトル、ヒストグラムが含まれています。また、このファイルには、格子のイメージを定期的に保存する機能も含まれています。
- ffteasy.cpp** フーリエ変換をするコードが含まれています。FFTEASY は、Gary Felder 氏が執筆した独立したルーチンのセットで、<http://www.science.smith.edu/departments/Physics/fstaff/gfelder/ffteasy> に記載されています。
- latticeeasy.h** 実行するためのすべてのパラメータが記載されています。グリッドサイズやタイムステップなどの変数に加え、どの出力をどのくらいの頻度で保存するかなどの変数もあります。
- model.h** それぞれのモデルに関する情報が含まれています。これらのファイルのライブラリは、"models" ディレクトリに含まれています。これらのファイルのどれかをプログラムディレクトリにコピーし、`model.h` という名前に変更することで、そのモデルを実行することができます。
- makefile** プログラムをコンパイルするためのもの。デフォルトは `g++` を使うためにセットされていますが、簡単に修正できます。