

Lattice Inflation Manual

目次

第 I 部	一般論	7
第 1 章	Lattice シミュレーションとは	8
1.1	概要	8
1.2	場の定義	9
1.3	Lattice シミュレーションの利点	10
1.4	リスケーリング	10
第 2 章	離散化	12
2.1	離散フーリエ変換	12
2.1.1	定義	12
2.1.2	逆格子ベクトルの取り方の等価性	13
2.1.3	FFT	14
2.2	微分演算子	16
2.3	有効運動量	17
2.4	フーリエモードの対応関係	18
2.5	解像度と Horizon-Crossing	19
第 3 章	初期条件	21
3.1	背景量	21
3.2	実空間での場の値を求める方法	22
3.3	ゆらぎ	23
3.3.1	解析解を再現する確率分布	23
3.3.2	ガウス分布の実装	24

3.3.3	等方的初期条件	26
3.3.4	場の微分のゆらぎ	26
3.3.5	リスケーリング	27
第 4 章	時間発展の方法	28
4.1	一般論	28
4.2	Euler 法	30
4.2.1	概要	30
4.2.2	実装	31
4.3	Leap-Frog 法	32
4.3.1	概要	32
4.3.2	実装	32
4.4	4 次の Runge-Kutta 法	33
4.4.1	概要	33
4.4.2	実装	33
第 5 章	アウトプット	36
5.1	背景量	36
5.2	ゆらぎの PDF	37
5.3	パワースペクトル	37
5.3.1	パワースペクトル	37
第 II 部	単一場インフレーション	40
第 6 章	セットアップと実装表式	41
6.1	セットアップのまとめ	41
6.2	初期条件	42
6.2.1	インフラトンゆらぎ	42
6.3	発展方程式	43
6.4	アウトプット	43
6.5	インフラトンポテンシャル	43
6.6	インプットパラメータ	43
第 7 章	ファイル構造	44

7.1	時間発展	44
7.1.1	単一場インフレーション	44
7.1.2	アクシオン-U(1) インフレーション	45
7.1.3	時間発展	46
7.2	アウトプット	47
7.2.1	背景量	47
第 8 章	コードレビュー	49
第 III 部	Axion-U(1) インフレーション	50
第 9 章	セットアップと実装表式	51
9.1	セットアップのまとめ	52
9.2	初期条件	52
9.3	発展方程式	52
9.4	アウトプット	52
9.5	インフラトンポテンシャル	52
9.6	インプットパラメータ	52
9.6.1	PBH 関係量	53
第 10 章	ファイル構造	56
第 11 章	コードレビュー	57
第 IV 部	利用に関して	58
第 12 章	利用上の注意事項	59
12.1	使用権限	59
12.2	コーラン条件	63
12.3	必要メモリ	63
第 13 章	利用方法	64
13.1	必要なアプリケーション	64
13.2	Lattice シミュレーションの実行	64

目次	4
13.3 グラフの描画	64
付録 A 計算過程の明記	66
A.1 第 I 部	66
A.2 第 II 部	69
参考文献	71

はじめに

本マニュアルは単一場インフレーション・Axion-U(1) インフレーションの Lattice シミュレーションコード「SFInflation」「axion-U(1)」を理解するためのマニュアル。一応 Lattice を知らない人向けだが、プログラミング言語についての知識は範囲外。

コードは C++ で書かれていて LATTICEEASY[1] のコードを適宜変更したものを用いている。LATTICEEASY のコードについての詳細は <https://www.felderbooks.com/latticeeasy/latticeeasydocs/latticeeasydocs.html> を参照。C++ の知識については私は「ロベールの C++ 入門講座」で勉強した。

インフレーションの Lattice シミュレーションとして主に参考にしたのは Angelo Caravano 氏の Simulating the inflationary Universe: from single-field to the axion-U(1) model[2] あるいは [3, 4, 5]。その他の参考は LATTICEEASY や CosmoLattice[6]。

注意事項として LatticeQCD とは根本的に違い、おそらく N 体シミュレーションと呼ばれるものに近い方の Lattice です。また、コードは「美しさ」の「う」の字くらいしかないので大いに改善の余地が残されていることをご了承ください。

本マニュアルの構成は以下の通り。

- 第 I 部：Lattice simulation の一般論
- 第 II 部：単一場インフレーションのシミュレーションとコード解説
- 第 III 部：Axion-U(1) インフレーションのシミュレーションとコード解説
- 第 IV 部：シミュレーションを実行する方法・注意事項
- 付録：計算過程の明記など

記法・規約

本論文を通して用いる単位系は、自然単位系 $c = k_B = \hbar = 1$ ・換算プランク質量を $M_{\text{pl}} = 1$ とする単位系である。

時間変数は共形時間 τ を用い、共形時間に関する微分はドット $\dot{}$ で表し、プログラム上の無次元化された共形時間 τ_{pr} に関する微分はプライム $'$ で表す。このようにプログラム上で扱う無次元化された量には”pr”の添字をつける。

議論すべき重要な仮定は曲率ゆらぎを無視していること。

用いる物理定数は以下の通り。

- 換算 Planck 質量

$$M_{\text{pl}} \equiv \sqrt{\frac{1}{8\pi G}} = 2.435 \times 10^{18} \text{ GeV}$$

- 太陽質量

$$M_{\odot} = 1.988 \times 10^{30} \text{ kg}$$

第I部

一般論

第 1 章

Lattice シミュレーションとは

1.1 概要

Lattice シミュレーションとは、有限な長さの一边を持つ立方領域上での連続場の動力学を数値的にシミュレーションする手法であり、連続な立方領域を N^3 個の格子点に離散化し各格子点上で運動方程式を解く。

格子を特徴づけるのに一般的に用いられる量は

- L : 立方領域の一边の（共動的な）長さ
- N : 一边の格子点数
- dx : （共動的な）格子間隔。 $dx = \frac{L}{N-1} \approx \frac{L}{N}$

の 3 つである。

連続的な立方領域の位置ベクトル

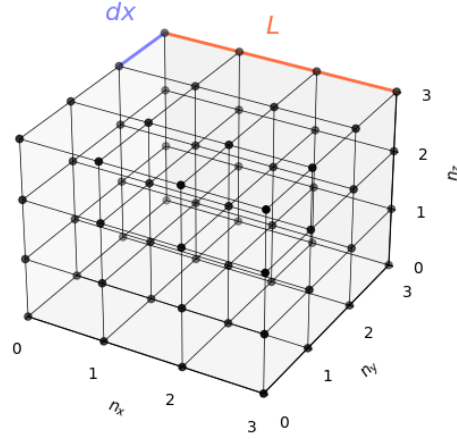
$$\boldsymbol{x} = \sum_i x_i \boldsymbol{e}_i \quad , \quad x_i = [0, L]$$

に対し、離散的な立方領域つまり（実）格子空間の位置ベクトルを

$$\boldsymbol{n} = \sum_i n_i \boldsymbol{e}_i \quad , \quad n_i \in \{0, 1, \dots, N-1\} \quad (1.1)$$

と定義する。但し基底ベクトル \boldsymbol{e}_i の定義は変更されていて、Lattice 上では格子間隔 dx に等しい大きさを持つ軸ベクトルとなる（格子で見れば 1 マス分なのでここでは基底ベクトルと呼んでいる）。

これらを踏まえて、図 1.1 に $N = 4$ の場合の 3 次元格子を示しておく。

図 1.1 $N = 4$ の場合の 3 次元格子

1.2 場の定義

時空上の連続場の値 $f(\tau, \mathbf{x})$ は、格子シミュレーションでは各格子点上で定義する。

$$f(\tau, \mathbf{x}) \longrightarrow f(\tau, \mathbf{n})$$

以下では場合に応じて時間依存性の表記を省略する。このようにスカラー場は N^3 個の値を持ち、各格子点上で運動方程式を解くことで場の発展を追う。格子空間の境界条件として**周期的境界条件**を課し、場の満たす条件式は

$$f(\tau, \mathbf{n}) = f(\tau, \mathbf{n} + N\mathbf{e}_x) = f(\tau, \mathbf{n} + N\mathbf{e}_y) = f(\tau, \mathbf{n} + N\mathbf{e}_z) \quad (1.2)$$

となる。

ゆらぎによって場は各格子点上で異なる値を持ち、背景量を取り出したければ格子平均値を計算すれば良い。

$$\bar{f}(\tau) \equiv \langle f(\tau) \rangle = \frac{1}{N^3} \sum_{\mathbf{n}} f(\tau, \mathbf{n}) \quad (1.3)$$

以降、格子シミュレーションの文脈においては $\langle \dots \rangle$ を格子平均の意味で用いる。

一方、ゆらぎの値を取り出したければ平均値を差し引いて

$$\delta f(\tau, \mathbf{n}) \equiv f(\tau, \mathbf{n}) - \bar{f}(\tau) \quad (1.4)$$

で計算される。これは宇宙論的摂動論におけるゆらぎの定義（空間依存しない平均値からのずれ）である。

1.3 Lattice シミュレーションの利点

格子シミュレーションでは各格子点で異なる場の値を取ることが許されているので、ゆらぎが場の値に原理的に組み込まれている。このゆらぎは値が大きくなることが許され、線形摂動論が破れるような場合も取り扱うことができ**非摂動論的な計算が可能**である。

しかし注意が必要なこともあり、ゆらぎが十分小さい時は線形摂動論と同様の結果が得られると期待されるが、プログラムで扱う変数（の型）が持つ有効数字の桁によって、ゆらぎの値が何桁まで信頼できるか決定される。例えば C++ の float 型は約 8 桁の有効数字を持ち、この 8 桁が場の値の有効桁となる。ここでゆらぎが平均値から 4 桁小さい場合、ゆらぎの有効桁は 4 桁となる。

$$f(\tau, \mathbf{n}) = 1.0002745 \quad \longrightarrow \quad \delta f(\tau, \mathbf{n}) = 2.745 \times 10^{-4}$$

このようにゆらぎの有効桁は変数の型よりも少ないため、ゆらぎが小さい場合（厳密には時間発展の最中に 1 度でも小さくなった場合）は信頼できる桁に注意が必要である。

1.4 リスケーリング

プログラム上で数値的に扱うために時空座標を無次元化（リスケーリング）する必要がある。ここで導入するリスケーリングの定義は [1] や [2] とは異なり、私のオリジナルの必要最小限な定義である。リスケーリングパラメータは無次元化のための質量次元を持つ量（インフラトン質量 m ）とスケール因子の冪を調整する s の 2 つで

$$d\tau_{\text{pr}} = ma^s d\tau, \quad \mathbf{x}_{\text{pr}} = m\mathbf{x} \quad (1.5)$$

のように無次元化する。私は $s = 1$ を主に用いるが、以下では s の値を指定せずに計算を進める（ s の選択がシミュレーションに与える影響はあまりわかっていない）。

この無次元化によって、立方領域の一辺の長さ L や格子間隔 dx はプログラム上では

$$L_{\text{pr}} = mL, \quad dx_{\text{pr}} = m dx \quad (1.6)$$

で表される。また時間微分も変更を受け、共形時間 τ による微分（ドット $\dot{}$ ）とプログラム

時間 τ_{pr} による微分（プライム'）の関係は*¹

$$\dot{a} = ma^s a', \quad \ddot{a} = m^2 a^{2s} \left[a'' + s \frac{(a')^2}{a} \right] \quad (1.7)$$

$$\dot{f} = ma^s f', \quad \ddot{f} = m^2 a^{2s} \left[f'' + s \frac{a'}{a} f' \right] \quad (1.8)$$

ただし f は場や時間依存の量を表す。空間微分演算子は

$$\nabla_{\text{pr}} = \frac{\nabla}{m} \quad (1.9)$$

の関係がある。

プログラム上のエネルギー密度と圧力を以下で定義する。

$$\rho_{\text{pr}} \equiv \frac{\rho}{m^2 a^{2s}}, \quad p_{\text{pr}} \equiv \frac{p}{m^2 a^{2s}} \quad (1.10)$$

これに伴って、インフラトンポテンシャルに対しても

$$V(\phi)_{\text{pr}} \equiv \frac{V(\phi)}{m^2 a^{2s}} \quad (1.11)$$

を定義する。

ちなみに LATTICEEASY では、場の値の規格化・空間微分でスケール因子の冪を調整するというさらに多いリスケーリングパラメータを用いている。

*¹ 計算は付録を参照

第 2 章

離散化

本章では Lattice シミュレーションに伴う離散化の影響・実装方法をフーリエ変換と微分演算子に関して説明する。2.1 節で離散フーリエ変換を定義、2.2 節で微分演算子の離散化法を説明、2.3 節で物理的な運動量に対応する Lattice 上の有効運動量を定義、2.4 節で物理的なフーリエモードと Lattice 上の離散フーリエモードの対応関係を導出、2.5 で Lattice 上の Horizon-Crossing や Lattice シミュレーションの限界を議論する。

2.1 離散フーリエ変換

2.1.1 定義

Lattice シミュレーションでは実空間を離散化して実空間上で運動方程式を解くが、ゆらぎのパワースペクトルなどを計算するためにはフーリエ空間に移る必要がある。そのためにフーリエ変換を離散化するが、まずは逆格子ベクトル κ を定義する。

$$\begin{aligned}\kappa(\mathbf{m}) &= \frac{2\pi}{L}\mathbf{m}, \quad m_i \in \left\{-\frac{N}{2} + 1, \dots, \frac{N}{2}\right\} \\ \kappa_{\text{pr}}(\mathbf{m}) &= \frac{\kappa}{m} = \frac{2\pi}{L_{\text{pr}}}\mathbf{m}\end{aligned}\tag{2.1}$$

これを踏まえて離散フーリエ変換 (Discrete Fourier Transformation, DFT) とその逆変換を以下のように定義する。^{*1}

$$f(\tau, \boldsymbol{\kappa}) = \sum_{\mathbf{n}} f(\tau, \mathbf{n}) e^{-\frac{2\pi i}{N} \mathbf{m} \cdot \mathbf{n}} \quad (2.2)$$

$$f(\tau, \mathbf{n}) = \frac{1}{N^3} \sum_{\mathbf{m}} f(\tau, \boldsymbol{\kappa}) e^{\frac{2\pi i}{N} \mathbf{m} \cdot \mathbf{n}} \quad (2.3)$$

これは LATTICEASY[1] での定義と同じである。上記のように、実空間上とフーリエ空間上の場の区別を引数で行うこととする。実空間での場はリスケーリングの影響を受けず、DFT における位相が

$$\frac{2\pi}{N} \mathbf{m} \cdot \mathbf{n} = \boldsymbol{\kappa} \cdot d\mathbf{x} \mathbf{n} = \boldsymbol{\kappa}_{\text{pr}} \cdot d\mathbf{x}_{\text{pr}} \mathbf{n}$$

でリスケーリングの影響を受けないことから、フーリエモードもリスケーリングの影響を受けない。

$$f_{\text{pr}}(\mathbf{n}) = f(\mathbf{n}) \quad , \quad f_{\text{pr}}(\boldsymbol{\kappa}) = f(\boldsymbol{\kappa}) \quad (2.4)$$

フーリエモード $f(\boldsymbol{\kappa})$ は一般に複素数であり、 N^3 個の格子点を考慮するとその自由度は $2N^3$ である。しかし格子空間あるいは実空間における場の値 $f(\mathbf{n})$ が実数である場合、その自由度は N^3 でありフーリエモードの自由度も N^3 に削減される。 $f(\mathbf{n})$ が実場であることによる $f(\boldsymbol{\kappa})$ に対する条件式は

$$f^*(\boldsymbol{\kappa}) = f(-\boldsymbol{\kappa}) \quad , \quad \text{ただし} \quad \frac{2\pi}{L} \left(-\frac{N}{2} \right) \equiv \frac{2\pi}{L} \frac{N}{2} \quad (2.5)$$

である。^{*2}

2.1.2 逆格子ベクトルの取り方の等価性

(2.1) でベクトル \mathbf{m} の各成分は正負で定義した。これは (空間 2 次元の場合に置き換えれば) 図 2.1 の右に対応し、逆格子ベクトルの各大きさに対して全方向を考慮している。しかし $m_i \in \{0, \dots, N-1\}$ で定義する方法もあり、これは逆格子ベクトルの各成分が正の領域のみを考えていて図 2.1 の左に対応する。これらの逆格子ベクトルの定義は、フー

^{*1} 計算は付録を参照

^{*2} 計算は付録を参照

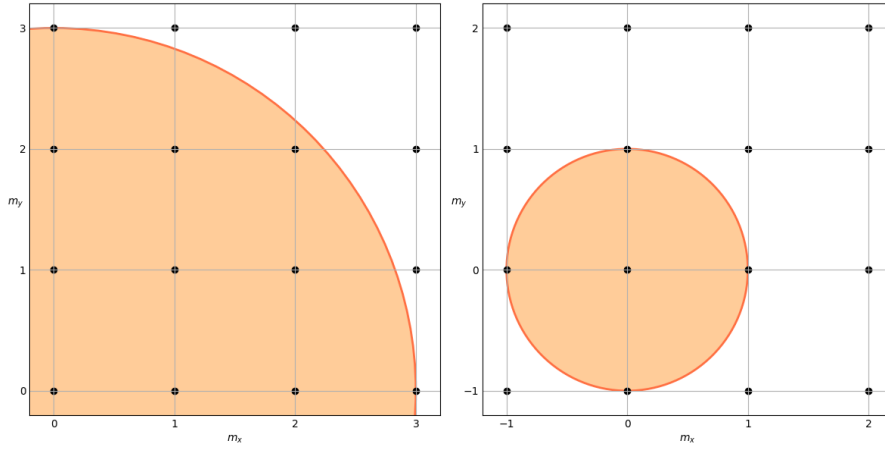


図 2.1 $N = 4$ かつ 2 次元での逆格子ベクトルのイメージ。正の成分（左）と全方向網羅（右）。

リエモードと後に定義する有効運動量ベクトルにおいては等価である。なぜならフーリエ変換の位相について

$$\begin{aligned}
 \exp \left[-\frac{2\pi i}{N} \mathbf{m} \cdot \mathbf{n} \right] &= \exp \left[-\frac{2\pi i}{N} (\mathbf{m} - \mathbf{N} + \mathbf{N}) \cdot \mathbf{n} \right] \\
 &= \exp \left[-\frac{2\pi i}{N} (\mathbf{m} - \mathbf{N}) \cdot \mathbf{n} \right] \exp [-2\pi i (n_x + n_y + n_z)] \\
 &= \exp \left[-\frac{2\pi i}{N} (\mathbf{m} - \mathbf{N}) \cdot \mathbf{n} \right]
 \end{aligned}$$

の関係があることから、 $m_i \in \{\frac{N}{2} + 1, \dots, N - 1\}$ に対して (2.1) の $m_i \in \{-\frac{N}{2} + 1, \dots, -1\}$ が対応している。私が (2.1) の定義を用いるのは、同じ $|\kappa|$ の値を持つ逆格子ベクトルあるいはフーリエモードのサンプル数を増やすためである。

2.1.3 FFT

プログラム上で DFT を実行することを考えると、愚直に (2.2)(2.3) をそのまま実装した場合は処理に長い時間がかかる。そこで高速フーリエ変換 (Fast Fourier Transformation, FFT) がよく用いられる。これは DFT を高速で処理するための離散フーリエ変換の 1 つの表現であり、**格子点数 N が 2 の累乗の時のみ成立** する。Lattice シミュレーションで $N = 512, 1024$ などがよく用いられるのはこの FFT の制限によるものである。以下で 1 次元の場合の FFT の表式を導く。

FFT では N 点の DFT を $N/2$ 点の 2 つの DFT に分解する。複素関数 h_k の離散複素

フーリエ変換は

$$\begin{aligned}
 H_n &= \sum_{k=0}^{N-1} W^{nk} h_k, \quad W = e^{\frac{2\pi i}{N}} \\
 &= \sum_{k=0}^{\frac{N}{2}-1} W^{n(2k)} h_{2k} + \sum_{k=0}^{\frac{N}{2}-1} W^{n(2k+1)} h_{2k+1} \\
 &= \sum_{k=0}^{\frac{N}{2}-1} e^{\frac{2\pi i}{N} nk} h_{2k} + W^n \sum_{k=0}^{\frac{N}{2}-1} e^{\frac{2\pi i}{N} nk} h_{2k+1} \\
 &\equiv H_n^e + W^n H_n^o
 \end{aligned}$$

ここで上付き添字の”e”は偶数 (even) を表し、”o”は奇数 (odd) を表す。 $N = 2^l$ で累乗で表されれば、上の分解を繰り返すことで和の記号を省略することができる。簡単な例として $N = 4$ の場合を具体的に見ると

$$\begin{aligned}
 H_n &= \sum_{k=0}^{\frac{N}{2}-1} e^{\frac{2\pi i}{N} nk} h_{2k} + W^n \sum_{k=0}^{\frac{N}{2}-1} e^{\frac{2\pi i}{N} nk} h_{2k+1} \\
 &= \sum_{k=0}^{\frac{N}{4}-1} e^{\frac{2\pi i}{N} n(2k)} h_{2(2k)} + \sum_{k=0}^{\frac{N}{4}-1} e^{\frac{2\pi i}{N} n(2k+1)} h_{2(2k+1)} \\
 &\quad + W^n \left[\sum_{k=0}^{\frac{N}{4}-1} e^{\frac{2\pi i}{N} n(2k)} h_{2(2k)+1} + \sum_{k=0}^{\frac{N}{4}-1} e^{\frac{2\pi i}{N} n(2k+1)} h_{2(2k+1)+1} \right] \\
 &= \sum_{k=0}^{\frac{N}{4}-1} e^{\frac{2\pi i}{N} nk} h_{4k} + W^{2n} \sum_{k=0}^{\frac{N}{4}-1} e^{\frac{2\pi i}{N} nk} h_{4k+2} \\
 &\quad + W^n \sum_{k=0}^{\frac{N}{4}-1} e^{\frac{2\pi i}{N} nk} h_{4k+1} + W^{3n} \sum_{k=0}^{\frac{N}{4}-1} e^{\frac{2\pi i}{N} nk} h_{4k+3} \\
 &\stackrel{N=4}{=} h_0 + W^{2n} h_2 + W^n h_1 + W^{3n} h_3 \\
 &\equiv H_n^{ee} + W^{2n} H_n^{eo} + W^n H_n^{oe} + W^{3n} H_n^{oo}
 \end{aligned}$$

最終的な h の下付き添字と H の上付き添字には関係があり、「 e, o の順番を反転させ、 $e = 0, o = 1$ として 2 進数を作り、10 進数に直すと h の下付き添字に等しくなる」。コードは LATTICEASY のものを用いたが、アルゴリズムはあまりわかっていないので説明しない。

2.2 微分演算子

格子シミュレーションでは各格子点上で場の値を定義し運動方程式を解くが、異なる点での場の値は独立に発展するわけではない。なぜなら空間微分項によって、隣接格子点の情報が運動方程式に組み込まれるからである。これを実現するために、連続空間における空間微分演算子を離散化する必要がある。離散化に伴って微分は差分で定義されるが、その差分の取り方には任意性がある。ここではよく用いられる差分の取り方を列举しておく。

1 階微分を離散化する場合、解像度（実際の連続空間を離散化した時のピクセルあるいは dx の小ささ、 dx が小さいほど連続空間に近づき解像度が高くなる。モザイクアートのモザイク数が多いほど写真に近づくことと同じ。）を最大化するためには、注目格子点 \mathbf{n} の最隣接格子点 $\mathbf{n} \pm \mathbf{e}_i$ を用いればよく

$$\partial_i f(\mathbf{n}) = \frac{f(\mathbf{n} + \mathbf{e}_i) - f(\mathbf{n} - \mathbf{e}_i)}{2dx} \quad (2.6)$$

となる。プログラム上で実装する場合は

$$\partial_{i,\text{pr}} f(\mathbf{n}) = \frac{f(\mathbf{n} + \mathbf{e}_i) - f(\mathbf{n} - \mathbf{e}_i)}{2dx_{\text{pr}}} \quad (2.7)$$

のようにプログラム上の格子間隔 dx_{pr} を用いれば良い。

(2.6) の 1 階差分をもう一度適用して得られるラプラシアン of 離散化は

$$\nabla^2 f(\mathbf{n}) = \frac{1}{(2dx)^2} \sum_{i=x,y,z} \sum_{c=\pm 2} [f(\mathbf{n} + c\mathbf{e}_i) - f(\mathbf{n})] \quad (2.8)$$

である。しかしこの定義ではラプラシアンの解像度が最大化されていない。ラプラシアンの解像度が最大化されるような定義は最隣接格子点を用いて

$$\nabla^2 f(\mathbf{n}) = \frac{1}{dx^2} \sum_{i=x,y,z} \sum_{c=\pm 1} [f(\mathbf{n} + c\mathbf{e}_i) - f(\mathbf{n})] \quad (2.9)$$

と表される。

2.3 有効運動量

格子上的有効運動量ベクトルを、物理的な運動量ベクトルに対応するものとして定義する。連続空間では微分演算子のフーリエ変換で運動量ベクトルが現れるので、有効運動量ベクトルも同様に微分演算子の離散フーリエ変換で定義する。思想は連続空間でのフーリエモードの運動方程式を再現することで、そうすれば観測量であるパワースペクトルを再現できると期待する。(2.9) に対する有効運動量ベクトルは

$$\begin{aligned}
 \text{DFT}[\nabla^2 f(\mathbf{n})](\boldsymbol{\kappa}) &= \sum_{\mathbf{n}} \frac{1}{dx^2} \sum_{i=x,y,z} [f(\mathbf{n} + \mathbf{e}_i) + f(\mathbf{n} - \mathbf{e}_i) - 2f(\mathbf{n})] e^{-\frac{2\pi i}{N} \mathbf{m} \cdot \mathbf{n}} \\
 &= \frac{1}{dx^2} \sum_{\mathbf{n}} \sum_{i=x,y,z} \left[f(\mathbf{n} + \mathbf{e}_i) e^{-\frac{2\pi i}{N} \mathbf{m} \cdot (\mathbf{n} + \mathbf{e}_i - \mathbf{e}_i)} \right. \\
 &\quad \left. + f(\mathbf{n} - \mathbf{e}_i) e^{-\frac{2\pi i}{N} \mathbf{m} \cdot (\mathbf{n} + \mathbf{e}_i - \mathbf{e}_i)} - 2f(\mathbf{n}) e^{-\frac{2\pi i}{N} \mathbf{m} \cdot \mathbf{n}} \right] \\
 &= \frac{1}{dx^2} \sum_{\mathbf{n}} \sum_{i=x,y,z} \left[f(\mathbf{n}) e^{-\frac{2\pi i}{N} \mathbf{m} \cdot (\mathbf{n} - \mathbf{e}_i)} \right. \\
 &\quad \left. + f(\mathbf{n}) e^{-\frac{2\pi i}{N} \mathbf{m} \cdot (\mathbf{n} + \mathbf{e}_i)} - 2f(\mathbf{n}) e^{-\frac{2\pi i}{N} \mathbf{m} \cdot \mathbf{n}} \right] \\
 &= \frac{1}{dx^2} f(\boldsymbol{\kappa}) \sum_{i=x,y,z} \left[2 \cos \left(\frac{2\pi m_i}{N} \right) - 2 \right] \\
 &= - \left[\frac{2}{dx} \sqrt{\sin^2 \left(\frac{\pi m_x}{N} \right) + \sin^2 \left(\frac{\pi m_y}{N} \right) + \sin^2 \left(\frac{\pi m_z}{N} \right)} \right]^2 f(\boldsymbol{\kappa})
 \end{aligned}$$

より

$$[k_{\text{eff}}(\mathbf{m})]_i = \frac{2}{dx} \sin \left(\frac{\pi m_i}{N} \right) \quad (2.10)$$

同様に (2.6)(2.8) に対する有効運動量ベクトルは

$$[k_{\text{eff}}(\mathbf{m})]_i = \frac{1}{dx} \sin \left(\frac{2\pi m_i}{N} \right) \quad (2.11)$$

である。 dx が共動的な長さであるので、これらの運動量も共動的な量である。

有効運動量ベクトルのリスケージングは逆格子ベクトルと同様に

$$\mathbf{k}_{\text{eff,pr}} = \frac{\mathbf{k}_{\text{eff}}}{m} \quad (2.12)$$

2.4 フーリエモードの対応関係

物理的には無限空間かつ連続空間でフーリエモード $f(\mathbf{k})$ が計算される。一方、Lattice シミュレーションでは有限空間かつ離散空間で離散フーリエモード $f(\boldsymbol{\kappa})$ が計算される。ここでは $f(\mathbf{k})$ と $f(\boldsymbol{\kappa})$ の対応関係を求める。

まず有限化に伴う補正を考える。ある量に対して 2 乗の空間平均が体積 V に依存しないことを要求する [1]。

$$\begin{aligned}\langle f(\mathbf{x})^2 \rangle &= \frac{1}{V} \int d^3x \int \frac{d^3k_1}{(2\pi)^3} \int \frac{d^3k_2}{(2\pi)^3} f(\mathbf{k}_1) f(\mathbf{k}_2) e^{i(\mathbf{k}_1 + \mathbf{k}_2) \cdot \mathbf{x}} \\ &= \frac{1}{V} \int \frac{d^3k}{(2\pi)^3} f(\mathbf{k}) f(-\mathbf{k}) \\ &= \frac{1}{V} \frac{1}{(2\pi)^3} \int d^3k |f(\mathbf{k})|^2\end{aligned}$$

ただし 2 つ目の $=$ では Dirac のデルタ関数を用い、3 目の $=$ では実空間での場の値 $f(\mathbf{x})$ が実数であることから得られる $f^*(\mathbf{k}) = f(-\mathbf{k})$ を用いた。ここで結果が体積 V に依存しないためには $|f(\mathbf{k})| \propto V^{\frac{1}{2}} = L^{\frac{3}{2}}$ である必要がある。

次に離散化に伴う補正を考える。

$$f(\mathbf{k}) = \int d^3x f(\mathbf{x}) e^{-i\mathbf{k} \cdot \mathbf{x}} \approx (dx)^3 \sum_{\mathbf{n}} e^{-\frac{2\pi i}{N} \mathbf{m} \cdot \mathbf{n}} = (dx)^3 f(\boldsymbol{\kappa})$$

より離散化に伴って dx^3 の因子が必要である。

最終的に、物理的なフーリエモードと Lattice 上のフーリエモードの関係は

$$f(\boldsymbol{\kappa}) = \frac{L^{\frac{3}{2}}}{dx^3} f(\mathbf{k}) = m^{\frac{3}{2}} \frac{L_{\text{pr}}^{\frac{3}{2}}}{dx_{\text{pr}}^3} f(\mathbf{k}) \quad (2.13)$$

この関係は、以降でシミュレーションにおけるゆらぎの初期条件を求める際や、シミュレーションのパワースペクトルから物理的なパワースペクトルを得る際に用いる。

2.5 解像度と Horizon-Crossing

Lattice シミュレーションにおける「解像度」とは dx のことである。Lattice シミュレーションでは例えば $\frac{dx}{2}$ の距離スケールのゆらぎは原理上扱うことができず、格子間隔 dx が Lattice シミュレーションで扱える最小の距離スケールである。 dx が小さければ小さい程解像度が高く、解像度を上げるには格子点数 N を大きくする必要がある。

インフレーションの Lattice シミュレーションでは、 L あるいは dx が共動的な長さを表す（物理的な長さを表すようにする設定もあるのかも）ため、上で述べた解像度があるという間に低く（悪く）なってしまう長い間の時間発展を追うことはできない。追うことができるのは数 e-folds で、インフレーション開始から終了まで Lattice シミュレーションすることは実質的に不可能である（無限のメモリ容量と無限の計算力があればその限りではないけれども）。

具体的には、共動的な距離スケール l と Horizon スケールは

$$l : \frac{1}{aH}$$

の比較で行い、インフレーション中はスケール因子が（宇宙時間 t に関しては）指数関数的に増加するため、ホライズンスケールが急激に小さくなり共動的な距離スケール l がすぐ super-horizon になる。Lattice シミュレーションでは数 e-folds のうちに最小の距離スケール dx が super-horizon になり、sub-horizon の物理が追えなくなってしまう。ただし、インフレーションにおける量子ゆらぎが sub-horizon で古典的に扱って良いかどうかは議論の余地あり。

次に Lattice シミュレーションにおける Horizon-Crossing を説明する。理論的によく議論される Horizon-Crossing は、共動的な運動量スケール k と Horizon スケールの逆数で

$$\begin{array}{ll} k < aH & \text{Super - Horizon} \\ k > aH & \text{Sub - Horizon} \end{array}$$

のように評価される。2.3 節で述べたように、Lattice シミュレーションでは物理的な運動量に対応するものとして有効運動量が存在するので、Lattice シミュレーションにおけ

る Horizon-Crossing は有効運動量 k_{eff} で評価する。プログラム上の量で表せば

$$\begin{aligned} k_{\text{eff,pr}} < a^{s-1} a' & \quad \text{Super - Horizon} \\ k_{\text{eff,pr}} > a^{s-1} a' & \quad \text{Sub - Horizon} \end{aligned}$$

で評価する。

Lattice シミュレーションでは小さい k_{eff} から順に Horizon-Crossing していき、最後に最大の k_{eff} が Horizon-Crossing する。この最大の k_{eff} が Horizon-Crossing すると、Sub-Horizon の物理を見ることができなくなるので、これを Lattice シミュレーションが追える最終時刻とする。つまり、「最大の k_{eff} が Horizon-Crossing する時刻あたりまで Lattice シミュレーションで追える」。また、Sub-Horizon の物理を見るために「シミュレーション開始時刻で、最小の k_{eff} が Horizon スケール程度であるように設定する」。以上 2 つの基準を考慮すれば、開始時刻での Hubble パラメータの値から L がおよそ決定され、 N に応じて最大の k_{eff} が決まりシミュレーションで追える期間が決定される。

具体的には $k_{\text{eff,min}} \sim \frac{2\pi}{L}$ なので

$$k_{\text{eff,min}} \lesssim [aH]_{\text{ini}} \iff L \gtrsim \frac{2\pi}{H_{\text{ini}}}$$

で L が決まり（上の条件は最小の k_{eff} がシミュレーション開始時刻で少し Super-Horizon になるようにしている）、このように選んだ L と使用しているコンピュータで可能な最大の N から

$$k_{\text{eff,max}} \sim \frac{1}{dx}$$

のように最大の k_{eff} が決まる。この $k_{\text{eff,max}}$ が Horizon-Crossing する e-folds は、インフレーションが slow-roll していて Hubble パラメータが一定であると近似（仮定）して

$$\begin{aligned} k_{\text{eff,max}} \sim a_{\text{end}} H_{\text{ini}} & \iff a_{\text{end}} \sim \frac{1}{H_{\text{ini}} dx} \\ & \iff N_{e,\text{end}} \sim \ln \left(\frac{1}{H_{\text{ini}} dx} \right) \end{aligned}$$

でおよそ決定される。厳密には $k_{\text{eff,max}}$ の表式（あるいは k_{eff} の定義）に依存している。上の式から分かることは、**格子点数 N を 2 倍したとしても $N_{e,\text{end}}$ は $\ln 2 \approx 0.3$ しか延長されない**ということである。最後に注意書きとして、本節では次章で述べるスケール因子の初期条件 $a_{\text{ini}} = 1$ を用いている。

第 3 章

初期条件

本章では Lattice シミュレーションにおける背景的な量（時間変数 τ, a, H, N_e 、場の平均値）とゆらぎについて初期値の取り方を説明する。ゆらぎについては CosmoLattice[6] と同じ手法を用いて決定する。3.1 節で背景的な量の初期条件を説明し、3.2 節でゆらぎのフーリエモードから実空間での場の値を得る方法を説明し、3.3 節でゆらぎのフーリエモードの初期条件を説明する。

3.1 背景量

- ・スケール因子

Lattice シミュレーション開始時のスケール因子は 1 と規格化する。

$$a_{\text{ini}} = 1 \quad (3.1)$$

- ・e-folds

スケール因子の初期値に合わせて、Lattice シミュレーションにおける e-folds の初期値は 0 である。

$$N_{e,\text{ini}} = 0 \quad (3.2)$$

- ・共形時間

共形時間は de-Sitter 宇宙の関係式から決定する。

$$\tau_{\text{ini}} = -\frac{1}{a_{\text{ini}} H_{\text{ini}}} = -\frac{1}{H_{\text{ini}}} \quad (3.3)$$

de-Sitter 宇宙の関係式を用いるため、Lattice シミュレーション開始時には slow-roll 近似が十分良い精度で成立している必要がある。

- ハッブルパラメータ

ハッブルパラメータは Friedmann 方程式から決めるが、ゆらぎの寄与は無視する。

$$H_{\text{ini}} = \sqrt{\frac{\rho_{\text{ini}}}{3}} = m \sqrt{\frac{\rho_{\text{pr,ini}}}{3}} \quad (3.4)$$

- スケール因子の微分

スケール因子の微分はリスケーリングの影響を受ける。スケール因子のプログラム時間による微分は

$$a'_{\text{ini}} = \frac{\dot{a}_{\text{ini}}}{ma_{\text{ini}}^s} = \frac{a_{\text{ini}}^2}{ma_{\text{ini}}^s} H_{\text{ini}} = \frac{H_{\text{ini}}}{m} = \sqrt{\frac{\rho_{\text{pr,ini}}}{3}} \quad (3.5)$$

- 場の平均値

場の値は背景量とゆらぎから初期値を決定する（後述）。インフラトン場に関する背景量 $\bar{\phi}, \dot{\bar{\phi}}$ は、背景インフレーションから決める。この背景場の値によって Lattice シミュレーション開始時の e-folds が決定される。場の平均値 $\bar{\phi}$ はリスケーリングの影響を受けないが、場の微分は時間微分からリスケーリングの影響を受ける。インプットパラメータである場の微分の初期値 $\dot{\bar{\phi}}_{\text{ini}}$ に対して、プログラム上の実装は

$$\bar{\phi}'_{\text{ini}} = \frac{\dot{\bar{\phi}}_{\text{ini}}}{ma_{\text{ini}}^s} = \frac{\dot{\bar{\phi}}_{\text{ini}}}{m} \quad (3.6)$$

3.2 実空間での場の値を求める方法

上で述べたように、場の初期値は背景量とゆらぎから決定する。ゆらぎの初期条件の決定方法は次節で述べるが、結果としてはゆらぎの離散フーリエ変換 $\delta f(\boldsymbol{\kappa})$ とその時間微分 $\delta f'(\boldsymbol{\kappa})$ が与えられる。これらから実空間の場の値を求めるために、まずゆらぎのフーリエ変換を逆離散フーリエ変換 (2.3) して実空間のゆらぎを求める。このようにして得られた各格子点でのゆらぎに背景量を加えることで場の初期値を決定する。

$$\begin{aligned} \delta\phi(\boldsymbol{\kappa}) &\xrightarrow{(2.3)} \delta\phi(\boldsymbol{n}) \xrightarrow{+\bar{\phi}} \bar{\phi} + \delta\phi(\boldsymbol{n}) \equiv \phi(\boldsymbol{n}) \\ \delta\phi'(\boldsymbol{\kappa}) &\xrightarrow{(2.3)} \delta\phi'(\boldsymbol{n}) \xrightarrow{+\bar{\phi}'} \bar{\phi}' + \delta\phi'(\boldsymbol{n}) \equiv \phi'(\boldsymbol{n}) \end{aligned}$$

3.3 ゆらぎ

Lattice シミュレーションにおけるゆらぎの初期条件として「**解析的な計算を再現できるようなゆらぎであること**」を課す。具体的には、ゆらぎの主な観測量であるパワースペクトルを再現するような量子ゆらぎを初期条件として課す。私のコードではインフラトンゆらぎの初期条件に slow-roll 近似の解析解である Hankel 関数を用いたが、これはつまり **slow-roll 近似と線形摂動論が良い精度で成立している時刻を Lattice シミュレーションの開始時刻に設定する必要がある**ことを指す。

本節ではモード関数として指数関数を採用し、確率密度分布がガウス分布となるようなゆらぎの実装方法を説明する。モード関数の一般化は第 2 部と第 3 部で行う。

3.3.1 解析解を再現する確率分布

解析的に得られるインフラトンパワースペクトルは sub-horizon で

$$\mathcal{P}_{\delta\phi}(k) = |\delta\phi(k)|^2 = \frac{1}{2a^2 \sqrt{k^2 + m_{\text{eff}}^2}} \quad , \quad m_{\text{eff}} = \frac{d^2 V(\phi)}{d\phi^2} \quad (3.7)$$

(無次元パワースペクトルでは k^3 が掛かり $\mathcal{P} \propto k^2$ となる)。ここでインフラトンゆらぎのモード関数は

$$\delta\phi(\tau, k) = \frac{1}{a\sqrt{2(k^2 + m_{\text{eff}}^2)}} e^{-i\sqrt{k^2 + m_{\text{eff}}^2}\tau} \quad (3.8)$$

である。

(3.7) の値を分散として出すような $|\delta\phi(k)|$ の分布は Rayleigh 分布

$$\text{PDF}[|\delta\phi|] = \left(\frac{\mathcal{P}_{\delta\phi}}{2}\right)^{-1} |\delta\phi| \exp\left[-\frac{1}{2} \left(\frac{\mathcal{P}_{\delta\phi}}{2}\right)^{-1} |\delta\phi|^2\right] \quad (3.9)$$

である*1。ただし”確率変数”の範囲は $0 \leq |\delta\phi|$ である。コード上で欲しいのは複素数 $\delta\phi(k)$ であり、 $|\delta\phi(k)|$ が上記の Rayleigh 分布になるような $\delta\phi(k)$ の分布は $N(0, \frac{\mathcal{P}_{\delta\phi}}{2})$ の複素ガウス分布である。ここで言う複素ガウス分布とは、複素数 $\delta\phi(k)$ の実部 $\text{Re}[\delta\phi(k)]$ と虚部 $\text{Im}[\delta\phi(k)]$ が各々独立に $N(0, \frac{\mathcal{P}_{\delta\phi}}{2})$ に従うという意味である。

*1 計算は付録を参照

$\text{Re}[\delta\phi(k)], \text{Im}[\delta\phi(k)]$ の同時確率分布は*2

$$\text{PDF} [\text{Re}[\delta\phi(k)], \text{Im}[\delta\phi(k)]] = \frac{1}{2\pi \frac{\mathcal{P}_{\delta\phi}}{2}} \exp \left[-\frac{(\text{Re}[\delta\phi(k)])^2 + (\text{Im}[\delta\phi(k)])^2}{2 \frac{\mathcal{P}_{\delta\phi}}{2}} \right] \quad (3.10)$$

3.3.2 ガウス分布の実装

(3.10) でフーリエモードの実部と虚部がガウス分布に従えば良いことがわかった。ここでは一様乱数 X, Y からガウス乱数（ガウス分布に従う確率変数）を得る公式を求める。

一様乱数の生成

LATTICEEASY で用いられている生成方法と同じく、「PM(Park-Miller) minimum standard random number generator」を用いる。これは

$$a = 7^5 = 16807, \quad m = 2^{31} - 1 = 2147483647$$

を用いて

$$X_{n+1} = (a \times X_n) \bmod m$$

によって疑似乱数を生成する方法である（線形合同法と似てる）。ただし、プログラム上では数値が大きくなりオーバーフロー（変数の最大値を上回り正しくない結果を出すこと）してしまうので

$$q = 127773, \quad r = 2836 \quad (m = aq + r)$$

を用いて

$$X_{n+1} = \begin{cases} a \times [X_n \bmod q] - r \times [\text{quotient of } X_n/q] & X_n \geq 0 \\ a \times [X_n \bmod q] - r \times [\text{quotient of } X_n/q] + m & X_n \leq 0 \end{cases}$$

を計算することで同様の結果が得られる。

*2 計算は付録を参照

一様乱数からガウス乱数の生成公式

X, Y を $[0, 1]$ の一様乱数とする。同時確率分布は

$$\text{PDF}[X, Y] = 1 \quad , \quad (0 \leq X, Y \leq 1) \quad (3.11)$$

であり、天下りの的に

$$r = \sqrt{-2 \ln X} \quad , \quad \theta = 2\pi Y \quad (3.12)$$

と変数変換すると

$$\begin{aligned} \int_0^1 dX \int_0^1 dY &= \int_0^1 dX \frac{e^{\ln X}}{X} \int_0^1 dY \\ &= \int_1^0 -\frac{dX}{X} e^{-(-\ln X)} \int_0^1 dY \frac{2\pi}{2\pi} \\ &= \int_0^\infty dr r e^{-\frac{r^2}{2}} \frac{1}{2\pi} \int_0^{2\pi} d\theta \\ &= \int_0^\infty dr r \int_0^{2\pi} d\theta \frac{1}{2\pi} e^{-\frac{r^2}{2}} \\ &= \int_{-\infty}^\infty dx \int_{-\infty}^\infty dy \frac{1}{2\pi} e^{-\frac{x^2+y^2}{2}} \end{aligned}$$

最後の「=」で極座標からデカルト座標

$$x = r \cos \theta \quad , \quad y = r \sin \theta$$

に座標変換した。

x, y は分散 1 のガウス分布に従うので、(3.10) のガウス分布のためには標準偏差 $\sqrt{\frac{\mathcal{P}_{\delta\phi}}{2}}$ を掛けて

$$\begin{aligned} \text{Re}[\delta\phi(k)] &= \sqrt{\mathcal{P}_{\delta\phi}} \sqrt{-\ln X} \cos(2\pi Y) \\ \text{Im}[\delta\phi(k)] &= \sqrt{\mathcal{P}_{\delta\phi}} \sqrt{-\ln X} \sin(2\pi Y) \end{aligned}$$

よってシミュレーションで実装すべき複素数 $\delta\phi(k)$ は

$$\delta\phi(k) = \sqrt{\mathcal{P}_{\delta\phi}} \sqrt{-\ln X} e^{i2\pi Y} \quad (3.13)$$

ここで (3.7) より $\sqrt{\mathcal{P}_{\delta\phi}}$ は解析的な $|\delta\phi(k)|$ であるのでモード関数で表され、以降はモード関数を指定すれば (3.13) に則ってガウス分布の $\delta\phi(k)$ を実装できる。

3.3.3 等方的初期条件

(3.13) の実装では $e^{-i\sqrt{k^2+m_{\text{eff}}^2}\tau}$ の波を考えており、これは 1 方向に進む波のみを考慮していることに対応する（ゆらぎの初期条件に対して解析的なモード関数は絶対値で効いてくるため指数関数の位相の符号の違いは (3.13) には現れないが、場の微分のゆらぎの初期条件ではその違いが明示的に効いてくる、後述する）。1 方向に進む波では宇宙原理の等方性に逆らうため、CosmoLattice[6] でも採用されている「等方的初期条件」を用いる。等方的初期条件では上述した波と逆方向の $e^{i\sqrt{k^2+m_{\text{eff}}^2}\tau}$ も考慮する。この時の $\delta\phi(k)$ は

$$\delta\phi(k) = \frac{1}{\sqrt{2}} \sqrt{\mathcal{P}_{\delta\phi}} \left(\sqrt{-\ln X_1} e^{i2\pi Y_1} + \sqrt{-\ln X_2} e^{i2\pi Y_2} \right) \quad (3.14)$$

ここで $\sqrt{2}$ で割ったのは、ガウス分布の加法性を考慮して $|\delta\phi(k)|$ の分散が $\mathcal{P}_{\delta\phi}$ となるようにするためである。

ガウス分布の加法性とは、2 つのガウス分布 $N(\mu_1, \sigma_1^2), N(\mu_2, \sigma_2^2)$ に従う確率変数 x_1, x_2 に対して $x_1 + x_2$ が $N(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$ に従うことである。これを考慮して (3.14) は分散 $\mathcal{P}_{\delta\phi}$ に従う。式的には

$$x_{1,2} = \sqrt{\frac{\mathcal{P}_{\delta\phi}}{2}} \sqrt{-\ln X_{1,2}} e^{i2\pi Y_{1,2}}$$

の実部と虚部が各々 $N(0, \frac{\mathcal{P}_{\delta\phi}}{4})$ に従うため、(3.14) つまり $x_1 + x_2$ の実部と虚部は各々 $N(0, \frac{\mathcal{P}_{\delta\phi}}{2})$ に従う。よって (3.14) の $\delta\phi(k)$ は、分散が $\mathcal{P}_{\delta\phi}$ となる $|\delta\phi(k)|$ の Rayleigh 分布を導く。

3.3.4 場の微分のゆらぎ

場の微分のゆらぎはモード関数の微分から決める。(3.8) のモード関数に対しては

$$\begin{aligned} \partial_\tau \delta\phi(k) &= \partial_\tau \left[\frac{1}{a\sqrt{2(k^2+m_{\text{eff}}^2)}} e^{\pm i\sqrt{k^2+m_{\text{eff}}^2}\tau} \right] \\ &= \left[-\mathcal{H} - \frac{\partial_\tau \sqrt{k^2+m_{\text{eff}}^2}}{\sqrt{k^2+m_{\text{eff}}^2}} \pm i\sqrt{k^2+m_{\text{eff}}^2} \right] \frac{1}{a\sqrt{2(k^2+m_{\text{eff}}^2)}} e^{\pm i\sqrt{k^2+m_{\text{eff}}^2}\tau} \\ &\approx \left[-\mathcal{H} \pm i\sqrt{k^2+m_{\text{eff}}^2} \right] \frac{1}{a\sqrt{2(k^2+m_{\text{eff}}^2)}} e^{\pm i\sqrt{k^2+m_{\text{eff}}^2}\tau} \end{aligned}$$

と計算される。ただし LATTICEEASY[1] と同様に 2 項目の時間依存性 $\partial_\tau \sqrt{k^2+m_{\text{eff}}^2}$ を無視した。

(3.14) のように等方的初期条件を考慮して

$$\partial_\tau \delta\phi(k) = -\mathcal{H}\delta\phi(k) - i\sqrt{k^2 + m_{\text{eff}}^2} \sqrt{\frac{\mathcal{P}_{\delta\phi}}{2}} \left(\sqrt{-\ln X_1} e^{i2\pi Y_1} - \sqrt{-\ln X_2} e^{i2\pi Y_2} \right) \quad (3.15)$$

ここで右辺の $\delta\phi(k)$ は場のゆらぎで実装した (3.14) の値である。

3.3.5 リスケーリング

得られた表式をプログラム上で実装するためのリスケーリングを考慮する。(2.13) よりプログラム上で実装するフーリエモードは

$$\begin{aligned} \delta\phi(\boldsymbol{\kappa}) &= \frac{1}{\sqrt{2}} |\delta\phi|_{\text{pr}} \left(\sqrt{-\ln X_1} e^{i2\pi Y_1} + \sqrt{-\ln X_2} e^{i2\pi Y_2} \right) \\ |\delta\phi|_{\text{pr}} &= m \frac{L_{\text{pr}}^{\frac{3}{2}}}{dx_{\text{pr}}^3} \frac{1}{\sqrt{2}} \left(k_{\text{eff,pr}}^2 + \frac{m_{\text{eff}}^2}{m^2} \right)^{-\frac{1}{4}} \end{aligned} \quad (3.16)$$

ここで 1 式目はインフラトンゆらぎのモード関数に依らない一般的な表式で、2 式目の $|\delta\phi|_{\text{pr}}$ は sub-horizon 極限のモード関数 (3.8) を用いリスケーリング (2.12)(2.13) を考慮した表式である。またスケール因子の初期値が 1 であることも用いた。

プログラム上で実装するゆらぎの微分は

$$\begin{aligned} \delta\phi'(\boldsymbol{\kappa}) &= -a'_{\text{ini}} \delta\phi(\boldsymbol{\kappa}) \\ &\quad - i\sqrt{k_{\text{eff,pr}}^2 + \frac{m_{\text{eff}}^2}{m^2}} \frac{|\delta\phi|_{\text{pr}}}{\sqrt{2}} \left(\sqrt{-\ln X_1} e^{i2\pi Y_1} - \sqrt{-\ln X_2} e^{i2\pi Y_2} \right) \end{aligned} \quad (3.17)$$

である。

一様乱数 X_1, X_2, Y_1, Y_2 は各 $\boldsymbol{\kappa}$ に対して異なる値を用いる。また、 $\delta\phi'$ で用いる X_1, X_2, Y_1, Y_2 は同じ $\boldsymbol{\kappa}$ に対して $\delta\phi$ と同じ値を用いる。この場合、ゆらぎの生成に使用している自由度（一様乱数の数）は $2N^3$ であり、 $\delta\phi(\boldsymbol{n}), \delta\phi'(\boldsymbol{n})$ の自由度の数と一致する。 $2N^3$ の数え方は、1 つの $\boldsymbol{\kappa}$ に対して $\delta\phi, \delta\phi'$ で 4 つであり、 $\boldsymbol{\kappa}$ が格子点数と同数の N^3 個あるので $4N^3$ 。最後にフーリエモードの条件式 (2.5) から、実際に値を決めるの（独立な値）は $\kappa_z > 0$ のフーリエモードのみであり、最終的な自由度は半分の $2N^3$ である。(LATTICEASY[1] では正負方向の波の振幅を一致させるために $X_1 = X_2$ としているが、この場合は自由度が少ないためにガウス分布が正しく実現できないと考えられている)

第 4 章

時間発展の方法

本章では Lattice シミュレーション（より一般には常微分方程式の数値計算）で用いる時間発展あるいは時間積分の手法の概要を説明する。ただし具体的に実装する式などは第 II 部と題 III 部に記す。4.1 節で時間発展の概要を説明し、4.2 節以降で各手法について説明する。

4.1 一般論

常微分方程式の解法はいわゆる初期値問題であり、（場あるいは関数の）初期値を与えれば微分方程式（つまり時間発展の振る舞い）によって初期時刻から dt 後の値が得られ、これを繰り返せば原理的に任意の時刻における値を得ることができる。解析的に解いた場合は $dt \rightarrow 0$ の極限での解を得ることができていて厳密解となるが、数值的に解く場合は $dt \rightarrow 0$ の極限を取ることが不可能であるため厳密解からの誤差が生じる。この誤差は時間発展を繰り返すたびに積み重なっていく。以降は 1 回の時間発展における有限な時間間隔を Δt のように Δ を用いて表記する。（[この段落の日本語変かも](#)）

時間発展に用いる手法は様々あり、各手法で得られる数値解が厳密解からどの程度外れているのかを「何次の精度」という言葉で表す。これは 1 回の時間発展で

$$1 \text{ step Error} \equiv |y(t + \Delta t) - y_{i+1}| = \mathcal{O}(\Delta t^{p+1}) \quad (4.1)$$

の誤差を生じる場合に「 p 次の精度」とあると言う。ここで $y(t)$ が厳密解で y_i が数値解である。つまり 1 回の時間発展で誤差が時間間隔 Δt の $p + 1$ 乗のオーダーである場合に p 次の精度がある。なぜ p 次なのかというと、初期時刻から最終時刻まで N 回時間発展

した時の大域誤差（最終的な誤差）が

$$\mathcal{O}(\Delta t^{p+1}) \times N = C\Delta t^p(\Delta t N) = C\Delta t^p \Delta t_{\text{tot}} \quad (4.2)$$

と表されるからである。大域誤差に対して 1 回の時間発展で現れる誤差 (4.1) を局所誤差と呼ぶ。[（この内容はどこぞの web サイトを参照した）](#)

時間発展の手法は例えば

- Euler 法 (1 次)
- Leap-Frog 法
- Runge-Kutta 法

が存在する。私のコードで用いているのは 4 次の Runge-Kutta 法である。また、私のコードで用いている時間変数は共形時間 τ （プログラム上ではこれに対応するプログラム時間 τ_{pr} ）であり、その 1 回の時間間隔 $\Delta\tau_{\text{pr}}$ は一定にしている。

時間発展の手法は大抵の場合 1 階微分方程式に対する解法となっている。しかし物理学において解くべきは時間に関する 2 階微分方程式

$$\frac{d^2x(t)}{dt^2} = f(t, x) \quad (4.3)$$

で、これに 1 階微分方程式の解法を適用するために $v(t) = \frac{dx}{dt}$ と $x(t)$ を独立に扱う。こうすることで (4.3) は

$$\frac{dx(t)}{dt} = v(t) \quad , \quad \frac{dv(t)}{dt} = f(t, x) \quad (4.4)$$

のように 2 つの 1 階微分方程式になる。

4.2 Euler 法

4.2.1 概要

ここでは最も単純な「1 つの 1 変数 1 階微分方程式」に対する Euler 法を説明する。Lattice シミュレーションで用いるための一般化は次の 4.2.2 節で説明する。ここで考える微分方程式を

$$\frac{dy(t)}{dt} = f(t, y) \quad (4.5)$$

とする。Euler 法では時間微分を離散化して差分で定義し

$$\begin{aligned} f(t, y) &= \frac{y(t + \Delta t) - y(t)}{\Delta t} \\ \therefore y(t + \Delta t) &= y(t) + \Delta t f(t, y(t)) \end{aligned} \quad (4.6)$$

このような上式 1 式目の差分の取り方を「前進差分」と呼び、2 式目の Euler 法を「前進 Euler 法」と呼ぶ。この方法では時刻 t の情報のみから時刻 $t + \Delta t$ の情報を得ることができるようになっていて、「陽解法」と呼ぶ。

一方、以下のような「後退差分」で差分を定義する方法もあり

$$\begin{aligned} f(t, y) &= \frac{y(t) - y(t - \Delta t)}{\Delta t} \\ y(t + \Delta t) &= y(t) + \Delta t f(t + \Delta t, y(t + \Delta t)) \end{aligned} \quad (4.7)$$

このような Euler 法を「後退 Euler 法」と呼ぶ。この方法では、時刻 $t + \Delta t$ の情報を得るのに時刻 t の情報だけでは足りず他の方程式を解く必要がある。これを陽解法に対して「陰解法」と呼ぶ。

Euler 法の精度は 1 次であり、 Δt を小さくしてもあまり誤差は小さくならない。これは (4.6) が 1 次の Taylor 展開

$$y(t + \Delta t) \approx y(t) + \Delta t \frac{dy}{dt} \quad (4.8)$$

に対応しているからである。また、厳密解の近似として考えれば

$$\begin{aligned} y(t + \Delta t) &= y(t) + \int_0^{\Delta t} d\tilde{t} f(t + \tilde{t}, y(t + \tilde{t})) \\ &\approx y(t) + \Delta t f(t, y(t)) \end{aligned} \quad (4.9)$$

でこれはつまり図 4.1 のような近似をしていると解釈できる。

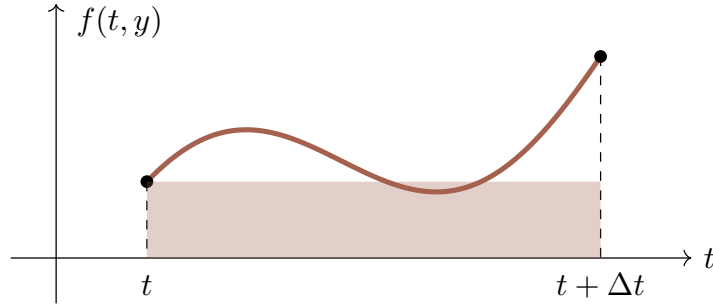


図 4.1 Euler 法の解釈（これほど f が変化するような Δt は大きすぎて本当は良くない）

4.2.2 実装

ここでは Lattice シミュレーションの時間発展に Euler 法を適用することを考える。4.1 節の最後で述べたように、1 つの 2 階微分方程式を 2 つの 1 階微分方程式に分解する。つの場合 $\phi(\tau, \mathbf{n})$ を考えた場合、各格子点で場の運動方程式を解くため $2N^3$ 個の 1 階微分方程式が存在する（ ϕ に対して N^3 個と ϕ' に対して N^3 個）。さらにスケール因子 a とその微分 a' に対する 1 階微分方程式が存在するため、合計 $2N^3 + 2$ 個の 1 階微分方程式に対して Euler 法を適用する。

$2N^3 + 2$ 個の微分方程式を

$$\frac{da(\tau_{\text{pr}})}{d\tau_{\text{pr}}} = a' \quad , \quad \frac{da'(\tau_{\text{pr}})}{d\tau_{\text{pr}}} = F^a(\tau_{\text{pr}}) \quad (4.10)$$

$$\frac{d\phi(\tau_{\text{pr}}, \mathbf{n})}{d\tau_{\text{pr}}} = \phi'(\mathbf{n}) \quad , \quad \frac{d\phi'(\tau_{\text{pr}}, \mathbf{n})}{d\tau_{\text{pr}}} = F^\phi(\tau_{\text{pr}}, \mathbf{n}) \quad (4.11)$$

とすると、時刻 τ_{pr} の情報 $a(\tau_{\text{pr}}), a'(\tau_{\text{pr}}), \phi(\tau_{\text{pr}}, \mathbf{n}), \phi'(\tau_{\text{pr}}, \mathbf{n})$ から

$$\begin{aligned} a(\tau_{\text{pr}} + \Delta\tau_{\text{pr}}) &= a(\tau_{\text{pr}}) + \Delta\tau_{\text{pr}} a'(\tau_{\text{pr}}) \\ a'(\tau_{\text{pr}} + \Delta\tau_{\text{pr}}) &= a'(\tau_{\text{pr}}) + \Delta\tau_{\text{pr}} F^a(\tau_{\text{pr}}) \\ \phi(\tau_{\text{pr}} + \Delta\tau_{\text{pr}}, \mathbf{n}) &= \phi(\tau_{\text{pr}}, \mathbf{n}) + \Delta\tau_{\text{pr}} \phi'(\tau_{\text{pr}}, \mathbf{n}) \\ \phi'(\tau_{\text{pr}} + \Delta\tau_{\text{pr}}, \mathbf{n}) &= \phi'(\tau_{\text{pr}}, \mathbf{n}) + \Delta\tau_{\text{pr}} F^\phi(\tau_{\text{pr}}, \mathbf{n}) \end{aligned} \quad (4.12)$$

と得られる。注意が必要なのは、例えば F^a が ϕ' に依存し F^ϕ が a' に依存している場合、 a' を発展させるために $\phi'(\tau_{\text{pr}})$ を用いれば今度は ϕ' を発展させるための $a'(\tau_{\text{pr}})$ が存在しなくなってしまう。このような問題の解決のためには a, a', ϕ, ϕ' の値を保存しておくダミー配列が必要になり、（特に N^3 個の要素を持つ ϕ, ϕ' によって）メモリが圧迫されるので Lattice シミュレーションでは致命的である。

4.3 Leap-Frog 法

4.3.1 概要

Leap-Frog 法はもともと 2 階微分方程式に適用するような手法である（1 階微分方程式に適用する Leap-Frog 法は知らないだけであるかも）。ただし 2 つの 1 階微分方程式に分解することは他の手法と同様である。ここでは (4.4) に対する Leap-Frog 法を説明する。Leap-Frog 法は位置 x と速度 v の参照時刻を $\frac{\Delta t}{2}$ だけずらし、交互に時間発展させる手法であり (4.4) から

$$\begin{aligned} x(t + \Delta t) &= x(t) + \Delta t v \left(t + \frac{\Delta t}{2} \right) \\ v \left(t + \frac{\Delta t}{2} \right) &= v \left(t - \frac{\Delta t}{2} \right) + \Delta t f(t, x) \end{aligned} \quad (4.13)$$

のように時間発展後の値を得ることができる。ただし一般的には $f(t, x)$ が速度 v にも依存する場合が考えられ、そのような場合には (4.13)2 式目最終項が時刻 t の情報だけで表されなくなり、Leap-Frog 法の実装が困難になる。

Leap-Frog 法の精度は 2 次であり、Runge-Kutta 法にはない「symplectic 性（エネルギー保存が良い性質）」を持っている。また、Leap-Frog 法の 4 次精度への一般化があり吉田法と呼ばれている。

4.3.2 実装

Euler 法と同様に (4.10)(4.11) に対する時間発展後の表式は

$$\begin{aligned} a(\tau_{\text{pr}} + \Delta\tau_{\text{pr}}) &= a(\tau_{\text{pr}}) + \Delta\tau_{\text{pr}} a' \left(\tau_{\text{pr}} + \frac{\Delta\tau_{\text{pr}}}{2} \right) \\ a' \left(\tau_{\text{pr}} + \frac{\Delta\tau_{\text{pr}}}{2} \right) &= a' \left(\tau_{\text{pr}} - \frac{\Delta\tau_{\text{pr}}}{2} \right) + \Delta\tau_{\text{pr}} F^a(\tau_{\text{pr}}) \\ \phi(\tau_{\text{pr}} + \Delta\tau_{\text{pr}}, \mathbf{n}) &= \phi(\tau_{\text{pr}}, \mathbf{n}) + \Delta\tau_{\text{pr}} \phi' \left(\tau_{\text{pr}} + \frac{\Delta\tau_{\text{pr}}}{2}, \mathbf{n} \right) \\ \phi' \left(\tau_{\text{pr}} + \frac{\Delta\tau_{\text{pr}}}{2}, \mathbf{n} \right) &= \phi' \left(\tau_{\text{pr}} - \frac{\Delta\tau_{\text{pr}}}{2}, \mathbf{n} \right) + \Delta\tau_{\text{pr}} F^\phi(\tau_{\text{pr}}, \mathbf{n}) \end{aligned} \quad (4.14)$$

となる。Leap-Frog 法では、Euler 法で問題となったようなダミー配列によるメモリ圧迫の心配はない。なぜなら参照時刻をずらしているおかげで交互に時間発展させることがで

きるからである。この利点は 4.3.1 節で注意した「 $f(t, x)$ が v に依存しない場合」によって保証されているものであるが、多分この条件は往々にして満たせない。(エイスケさんのコードではなんか解決できてた。忘れた。)

4.4 4 次の Runge-Kutta 法

4.4.1 概要

4 次の Runge-Kutta 法では 1 回の Δt 時間発展を行うのに、4 回の細かい時間発展を計算する。(4.5) に対する 4 次の Runge-Kutta 法は

$$y(t + \Delta t) = y(t) + \Delta t \left(\frac{S_1}{6} + \frac{S_2}{3} + \frac{S_3}{3} + \frac{S_4}{6} \right) \quad (4.15)$$

ここで $S_1 \sim S_4$ は

$$\begin{aligned} S_1 &= f(t, y(t)) & , & \quad S_2 = f\left(t + \frac{\Delta t}{2}, y(t) + \frac{\Delta t}{2} S_1\right) \\ S_3 &= f\left(t + \frac{\Delta t}{2}, y(t) + \frac{\Delta t}{2} S_2\right) & , & \quad S_4 = f(t + \Delta t, y(t) + \Delta t S_3) \end{aligned} \quad (4.16)$$

で求められる量である。

4.4.2 実装

上で見た最も単純な場合の Runge-Kutta 法 (4.15)(4.16) でもある程度複雑だが、Lattice シミュレーションに適用する場合にはより複雑になる。(4.10)(4.11) に対する時間発展の表式は

$$\begin{aligned} a(\tau_{\text{pr}} + \Delta\tau_{\text{pr}}) &= a(\tau_{\text{pr}}) + \Delta\tau_{\text{pr}} \left(\frac{S_1^a}{6} + \frac{S_2^a}{3} + \frac{S_3^a}{3} + \frac{S_4^a}{6} \right) \\ a'(\tau_{\text{pr}} + \Delta\tau_{\text{pr}}) &= a'(\tau_{\text{pr}}) + \Delta\tau_{\text{pr}} \left(\frac{S_1^{a'}}{6} + \frac{S_2^{a'}}{3} + \frac{S_3^{a'}}{3} + \frac{S_4^{a'}}{6} \right) \\ \phi(\tau_{\text{pr}} + \Delta\tau_{\text{pr}}, \mathbf{n}) &= \phi(\tau_{\text{pr}}, \mathbf{n}) + \Delta\tau_{\text{pr}} \left(\frac{S_1^\phi}{6} + \frac{S_2^\phi}{3} + \frac{S_3^\phi}{3} + \frac{S_4^\phi}{6} \right) \\ \phi'(\tau_{\text{pr}} + \Delta\tau_{\text{pr}}, \mathbf{n}) &= \phi'(\tau_{\text{pr}}, \mathbf{n}) + \Delta\tau_{\text{pr}} \left(\frac{S_1^{\phi'}}{6} + \frac{S_2^{\phi'}}{3} + \frac{S_3^{\phi'}}{3} + \frac{S_4^{\phi'}}{6} \right) \end{aligned} \quad (4.17)$$

である。ただし S は各変数に対して (4.16) と同様に

$$\begin{aligned} S_1^a &= a' & , & \quad S_2^a = a' + \frac{\Delta\tau_{\text{pr}}}{2} S_1^{a'} \\ S_3^a &= a' + \frac{\Delta\tau_{\text{pr}}}{2} S_2^{a'} & , & \quad S_4^a = a' + \Delta\tau_{\text{pr}} S_3^{a'} \end{aligned} \quad (4.18)$$

$$\begin{aligned} S_1^{a'} &= F^a(a, a', \phi, \phi') \\ S_2^{a'} &= F^a\left(a + \frac{\Delta\tau_{\text{pr}}}{2} S_1^a, a' + \frac{\Delta\tau_{\text{pr}}}{2} S_1^{a'}, \phi + \frac{\Delta\tau_{\text{pr}}}{2} S_1^\phi, \phi' + \frac{\Delta\tau_{\text{pr}}}{2} S_1^{\phi'}\right) \\ S_3^{a'} &= F^a\left(a + \frac{\Delta\tau_{\text{pr}}}{2} S_2^a, a' + \frac{\Delta\tau_{\text{pr}}}{2} S_2^{a'}, \phi + \frac{\Delta\tau_{\text{pr}}}{2} S_2^\phi, \phi' + \frac{\Delta\tau_{\text{pr}}}{2} S_2^{\phi'}\right) \\ S_4^{a'} &= F^a\left(a + \Delta\tau_{\text{pr}} S_3^a, a' + \Delta\tau_{\text{pr}} S_3^{a'}, \phi + \Delta\tau_{\text{pr}} S_3^\phi, \phi' + \Delta\tau_{\text{pr}} S_3^{\phi'}\right) \end{aligned} \quad (4.19)$$

$$\begin{aligned} S_1^\phi &= \phi' & , & \quad S_2^\phi = \phi' + \frac{\Delta\tau_{\text{pr}}}{2} S_1^{\phi'} \\ S_3^\phi &= \phi' + \frac{\Delta\tau_{\text{pr}}}{2} S_2^{\phi'} & , & \quad S_4^\phi = \phi' + \Delta\tau_{\text{pr}} S_3^{\phi'} \end{aligned} \quad (4.20)$$

$$\begin{aligned} S_1^{\phi'} &= F^\phi(a, a', \phi, \phi') \\ S_2^{\phi'} &= F^\phi\left(a + \frac{\Delta\tau_{\text{pr}}}{2} S_1^a, a' + \frac{\Delta\tau_{\text{pr}}}{2} S_1^{a'}, \phi + \frac{\Delta\tau_{\text{pr}}}{2} S_1^\phi, \phi' + \frac{\Delta\tau_{\text{pr}}}{2} S_1^{\phi'}\right) \\ S_3^{\phi'} &= F^\phi\left(a + \frac{\Delta\tau_{\text{pr}}}{2} S_2^a, a' + \frac{\Delta\tau_{\text{pr}}}{2} S_2^{a'}, \phi + \frac{\Delta\tau_{\text{pr}}}{2} S_2^\phi, \phi' + \frac{\Delta\tau_{\text{pr}}}{2} S_2^{\phi'}\right) \\ S_4^{\phi'} &= F^\phi\left(a + \Delta\tau_{\text{pr}} S_3^a, a' + \Delta\tau_{\text{pr}} S_3^{a'}, \phi + \Delta\tau_{\text{pr}} S_3^\phi, \phi' + \Delta\tau_{\text{pr}} S_3^{\phi'}\right) \end{aligned} \quad (4.21)$$

である。ただし a, a', ϕ, ϕ' は全て時刻 τ_{pr} における値であり、(4.20)(4.21) の S は本来 \mathbf{n} を引数に持ち各格子点について定義される量である。

Euler 法でも述べたように、正しく時間発展させるためには a, a', ϕ, ϕ' の値を保存しておくダミー配列が必要になる。以下で (4.17) から (4.21) の計算をする手順を説明しつつ、必要になるダミー配列の要素数も求める。

① まず時刻 τ_{pr} の情報 a, a', ϕ, ϕ' を持っているとして、 $S_1^a, S_1^{a'}, S_1^\phi, S_1^{\phi'}$ を計算する。この時、計算した S_1 の値を保存しておくために $2N^3 + 2$ 個の変数を持つダミー配列 ($S_1^a, S_1^{a'}$ が 1 個ずつ、 $S_1^\phi, S_1^{\phi'}$ が N^3 個ずつ) が必要になる。

$$a, a', \phi, \phi' \rightarrow S_1^a, S_1^{a'}, S_1^\phi, S_1^{\phi'} \text{ (Dummy List 1)}$$

②元々持っていた時刻 τ_{pr} の情報 a, a', ϕ, ϕ' と先ほど計算した $S_1^a, S_1^{a'}, S_1^\phi, S_1^{\phi'}$ から $S_2^a, S_2^{a'}, S_2^\phi, S_2^{\phi'}$ を計算する。この時、計算した S_2 の値を保存しておくために $2N^3 + 2$ 個の変数を持つ新たなダミー配列が必要になる。ここで時刻 τ_{pr} の情報 a, a', ϕ, ϕ' を書き換えていけないのは (4.17) や今後の S の計算でこれらの情報が必要になるからで、 $S_1^a, S_1^{a'}, S_1^\phi, S_1^{\phi'}$ を書き換えていけないのは S_2 の計算で S_1 の情報が必要だからである。

$$a, a', \phi, \phi' \text{ and } S_1^a, S_1^{a'}, S_1^\phi, S_1^{\phi'} \rightarrow S_2^a, S_2^{a'}, S_2^\phi, S_2^{\phi'} \text{ (Dummy List 2)}$$

③元々持っていた時刻 τ_{pr} の情報 a, a', ϕ, ϕ' と先ほど計算した $S_2^a, S_2^{a'}, S_2^\phi, S_2^{\phi'}$ から $S_3^a, S_3^{a'}, S_3^\phi, S_3^{\phi'}$ を計算する。この時 $S_1^a, S_1^{a'}, S_1^\phi, S_1^{\phi'}$ は (4.17) の時間発展を除けばもう使わないので、 S_1 の情報を $2N^3 + 2$ 個の変数を持つ新たなダミー配列に保存し、 S_3 の情報を元々 S_1 を保存していたダミー配列に上書きする。

$$S_1^a, S_1^{a'}, S_1^\phi, S_1^{\phi'} \text{ (Dummy List 1)} \rightarrow \frac{S_1^a}{6}, \frac{S_1^{a'}}{6}, \frac{S_1^\phi}{6}, \frac{S_1^{\phi'}}{6} \text{ (Dummy List 3)}$$

$$a, a', \phi, \phi' \text{ and } S_2^a, S_2^{a'}, S_2^\phi, S_2^{\phi'} \rightarrow S_3^a, S_3^{a'}, S_3^\phi, S_3^{\phi'} \text{ (Dummy List 1)}$$

④元々持っていた時刻 τ_{pr} の情報 a, a', ϕ, ϕ' と先ほど計算した $S_3^a, S_3^{a'}, S_3^\phi, S_3^{\phi'}$ から $S_4^a, S_4^{a'}, S_4^\phi, S_4^{\phi'}$ を計算する。この時、先程と同じように S_2 の情報を S_1 の情報が入っているダミー配列に追加し、 S_4 の情報を元々 S_2 を保存していたダミー配列に上書きする。

$$S_2 \text{ (Dummy List 2)} \rightarrow \frac{S_1}{6} + \frac{S_2}{3} \text{ (Dummy List 3)}$$

$$a, a', \phi, \phi' \text{ and } S_3^a, S_3^{a'}, S_3^\phi, S_3^{\phi'} \rightarrow S_4^a, S_4^{a'}, S_4^\phi, S_4^{\phi'} \text{ (Dummy List 2)}$$

⑤最後に S_3, S_4 の情報を S_1 と S_2 の情報が入っているダミー配列に追加し、この配列と時刻 τ_{pr} の情報 a, a', ϕ, ϕ' を用いて (4.17) を通して時刻 $\tau_{\text{pr}} + \Delta\tau_{\text{pr}}$ の情報を得る。これで得られる情報で時刻 τ_{pr} の配列を上書きして良い。

$$\begin{aligned} S_3, S_4 \text{ (Dummy List 1, 2)} &\rightarrow \frac{S_1}{6} + \frac{S_2}{3} + \frac{S_3}{3} + \frac{S_4}{6} \text{ (Dummy List 3)} \\ &\rightarrow a, a', \phi, \phi' \text{ at } \tau_{\text{pr}} + \Delta\tau_{\text{pr}} \text{ (Main List)} \end{aligned}$$

上記のような手順で $\Delta\tau_{\text{pr}}$ の時間発展を計算するが、ここで必要になったダミー配列の容量（メモリ）は変数 a, a', ϕ, ϕ' の 3 つ分である。つまり 4 次の Runge-Kutta 法を実行するためには、扱う変数（スケール因子と場の値）の 4 倍のメモリが必要になる。このメモリ圧迫は Lattice シミュレーションでは致命的だよね（泣）。

第 5 章

アウトプット

本章では Lattice シミュレーションで得られたプログラム上の値から、物理量を取り出すための方法を説明する。5.1 節で取り出す背景量とその取り出し方を説明し、5.3 節でゆらぎの PDF（確率分布関数）の計算方法を説明し、5.3 節で物理的なパワースペクトルの計算方法を説明する。

5.1 背景量

Lattice シミュレーションにおいて場に関する背景量は、各格子点での値を足し合わせた後に格子点数で割るという格子平均で計算される。計算式は

$$\langle f \rangle \equiv \frac{1}{N^3} \sum_{\mathbf{n}} f(t, \mathbf{n}) \quad (5.1)$$

である。以降はよく出力される量について、プログラム上の値から計算する方法を記す。

• e-folds

Lattice シミュレーション開始から経過した e-folds は

$$N_e = \ln \left(\frac{a}{a_{\text{ini}}} \right) = \ln a \quad (5.2)$$

• Hubble パラメータ

出力する Hubble パラメータはインフラトン質量で無次元化する。(1.7) を用いて

$$\frac{H}{m} = \frac{\partial_t a}{ma} = \frac{\partial_\tau a}{ma^2} = \frac{ma^s a'}{ma^2} = a^{s-2} a' \quad (5.3)$$

• 場の平均値

場 ϕ の値はリスケーリングの影響を受けないので (5.1) の格子平均を用いて

$$\bar{\phi} = \langle \phi \rangle \quad (5.4)$$

• 場の微分の平均値

場 ϕ の微分は時間微分からリスケーリングの影響を受けるので (1.8) を考慮する。また Hubble パラメータと同様にインフラトン質量で無次元化して

$$\frac{\overline{\partial_t \phi}}{m} = \frac{\langle \partial_\tau \phi \rangle}{ma} = \frac{ma^s \langle \phi' \rangle}{ma} = a^{s-1} \langle \phi' \rangle \quad (5.5)$$

• キュムラント

5.2 ゆらぎの PDF

5.3 パワースペクトル

5.3.1 パワースペクトル

[2] に基づいてパワースペクトルの計算方法を述べる。場 f に対して解析的なパワースペクトルは

$$\mathcal{P}_f(k) = |f(k)|^2$$

で計算される。フーリエモードの関係式 (??) を用いると、物理的なパワースペクトルを再現するためには

$$\mathcal{P}_f(k) = \frac{dx^6}{L^3} |f(\kappa)|^2$$

ここで、格子上では場のフーリエモードが逆格子ベクトル κ （あるいは有効運動量ベクトル \mathbf{k}_{eff} ）の方向にも依存するので、大きさにのみ依存する量を取り出すために逆格子ベクトルの方向で平均を取る。つまり

$$\mathcal{P}_f(k) = \frac{dx^6}{L^3} \frac{\sum_{\kappa} |f(\kappa)|}{\sum_{\kappa}} \quad (5.6)$$

で \sum_{κ} は大きさ κ を持つ任意の方向の逆格子ベクトル κ での和を表す。この和は連続空間では球面積分に対応し、格子上は離散化されているため幅を持たせて球殻で平均を取る。その幅は逆格子ベクトルの大きさの最小値である $\frac{2\pi}{L}$ とする。つまり、

$$\sum_{\kappa} = \sum_{[\frac{2\pi}{L}i, \frac{2\pi}{L}(i+1))}, \quad |\kappa| \times \frac{L}{2\pi} \in [i, i+1) \quad (5.7)$$

のように表現できる。

ここまで議論したパワースペクトルの計算は LATTICEEASY[1] で用いられている方法である。物理的な運動量に格子上で対応するのは有効運動量 (??)(??) であり、[2] ではこのことを考慮して (5.6) で計算した量は物理的な運動量 k_{eff} における情報とした。つまり

$$\mathcal{P}_f(k_{\text{eff}}) = \frac{dx^6}{L^3} \frac{\sum_{\kappa} |f(\kappa)|}{\sum_{\kappa}}, \quad k_{\text{eff}} = \frac{\sum_{\kappa} |\mathbf{k}_{\text{eff}}(\kappa)|}{\sum_{\kappa}} \quad (5.8)$$

で計算した。無次元パワースペクトルに直したければ

$$P_f(k_{\text{eff}}) = \frac{k_{\text{eff}}^3}{2\pi^2} \mathcal{P}_f(k_{\text{eff}}) \quad (5.9)$$

とすれば良い。LATTICEEASY[1] では k_{eff} ではなく κ を掛け算している。

前節で導入したリスケーリングを考慮して、プログラム上の値で表すと

$$\mathcal{P}_f(k_{\text{eff}}) = \frac{1}{m^3} \frac{dx_{\text{pr}}^6}{L_{\text{pr}}^3} \frac{\sum |f_{\text{pr}}(\kappa)|}{\sum} \quad (5.10)$$

$$P_f(k_{\text{eff}}) = \frac{k_{\text{eff,pr}}}{2\pi^2} \frac{dx_{\text{pr}}^6}{L_{\text{pr}}^3} \frac{\sum |f_{\text{pr}}(\kappa)|}{\sum} \quad (5.11)$$

ただし和 \sum と $k_{\text{eff,pr}}$ については

$$\sum = \sum_{[\frac{2\pi}{L_{\text{pr}}}i, \frac{2\pi}{L_{\text{pr}}}(i+1))}, \quad |\kappa_{\text{pr}}| \times \frac{L_{\text{pr}}}{2\pi} \in [i, i+1) \quad (5.12)$$

$$k_{\text{eff,pr}} = \frac{\sum |k_{\text{eff,pr}}(\kappa_{\text{pr}})|}{\sum} \quad (5.13)$$

である。

第Ⅱ部

単一場インフレーション

第 6 章

セットアップと実装表式

本章では、単一場インフレーションの Lattice シミュレーションで用いた詳細なセットアップと、実際に用いた実装表式を説明する。第 I 部で既に説明した内容については触れない。6.1 節で用いたセットアップのまとめを説明、6.2 節で初期条件の説明、6.3 節で実装する発展方程式の導出、6.4 節で出力する量の説明、6.5 節でインフラトンポテンシャルの計算、6.6 節でインプットすべきパラメータの説明をする。

ここでは概念的なセットアップと理論的な計算を記すため、具体的（あるいは技術的）なコードの解説は 7 章と 8 章を参照すべき。

6.1 セットアップのまとめ

単一実スカラー場インフレーションを考えるので、扱う変数はインフラトン場 ϕ ・その時間微分 ϕ' ・スケール因子 a ・その時間微分 a' である。全格子点数が N^3 個であることを考慮すれば、変数の数は $2N^3 + 2$ 個である。

インフラトンゆらぎは場の値 ϕ, ϕ' に含まれているが、計量ゆらぎについては小さいと無視し、平坦な FLRW 計量を考える。これは線形摂動論において計量ゆらぎは slow-roll パラメータに比例するので、インフラトンが十分 slow-roll していれば計量ゆらぎは無視できるからである [2]。今は計量ゆらぎを無視していて、この近似のためにシミュレーション中は十分 slow-roll している必要がある。

6.2 初期条件

6.2.1 インフラトンゆらぎ

3.3 節で量子ゆらぎを再現するインフラトンゆらぎの実装方法を見た。私は単一場インフレーションのシミュレーション「SFInflation」で、モード関数として Hankel 関数を用いる。これは 3.3 節で用いた sub-horizon 極限のモード関数 (3.8) の一般化であり、super-horizon のゆらぎも考慮できるモード関数である。ただし Hankel 関数を用いた場合でも、slow-roll 近似を課しているため **Lattice シミュレーション開始時に十分 slow-roll かつ線形摂動論が有効でなければならない**ことは変わらない。

Hankel 関数を用いた場合の解析的なインフラトンパワースペクトルは

$$\mathcal{P}_{\delta\phi}(k) = \frac{-\pi\tau}{4a^2} |H_\nu^{(1)}(-k\tau)|^2 \quad (6.1)$$

であり、インフラトンゆらぎのモード関数は

$$\delta\phi(k) = \frac{\sqrt{-\pi\tau}}{2a} H_\nu^{(1)}(-k\tau) \quad (6.2)$$

である。ここで ν は slow-roll の 0 次近似で評価して

$$\nu = \sqrt{\frac{9}{4} - \frac{m_{\text{eff}}^2}{H^2}} \quad , \quad m_{\text{eff}} = \frac{d^2 V(\phi)}{d\phi^2} \quad (6.3)$$

を用いる。Hankel 関数の引数を χ で表すと (2.12)(3.3)(3.4) より

$$\chi = \frac{k_{\text{eff}}}{H_{\text{ini}}} = \frac{k_{\text{eff,pr}}}{a'_{\text{ini}}} = k_{\text{eff}} \sqrt{\frac{3}{\rho_{\text{pr,ini}}}} \quad (6.4)$$

のようにプログラム上の値で与えられる。

この解析解を (3.16) に用いて、プログラム上で実装するインフラトンゆらぎ（のフリーモード）の初期条件は

$$\begin{aligned} \delta\phi(\boldsymbol{\kappa}) &= \frac{1}{\sqrt{2}} |\delta\phi|_{\text{pr}} \left(\sqrt{-\ln X_1} e^{i2\pi Y_1} + \sqrt{-\ln X_2} e^{i2\pi Y_2} \right) \\ |\delta\phi|_{\text{pr}} &= m \frac{L_{\text{pr}}^{\frac{3}{2}}}{dx_{\text{pr}}^3} \frac{1}{2} \sqrt{\frac{\pi}{a'_{\text{ini}}}} |H_\nu(\chi)| \end{aligned} \quad (6.5)$$

ただし Hankel 関数の第 1 種と第 2 種を表す上付き添字 (1), (2) は、この式では違いが現れないので省略した。第 1 種 Hankel 関数と第 2 種 Hankel 関数は各々 sub-horizon 極限

で $e^{\mp ik\tau}$ に比例するため、3.3.3 節の等方的初期条件から両方の寄与を考慮している。これは時間微分のゆらぎを考える時に重要になる。

インフラトン場の時間微分のゆらぎは 3.3.4 節のようにモード関数の微分から求める。Hankel 関数は Bessel 関数 J_ν と Nuemann 関数 N_ν を用いて

$$H_\nu^{(1)} = J_\nu + iY_\nu \quad , \quad H_\nu^{(2)} = J_\nu - iY_\nu$$

と書け、これらの満たす微分漸化式を用いて計算される (3.17) に対応するプログラム上の微分ゆらぎは

$$\begin{aligned} \delta\phi'(\boldsymbol{\kappa}) = & \left[\left(\nu - \frac{3}{2} \right) a'_{\text{ini}} - k_{\text{eff,pr}} \frac{J_\nu J_{\nu-1} + Y_\nu Y_{\nu-1}}{|H_\nu|^2} \right] \delta\phi(\boldsymbol{\kappa}) \\ & + ik_{\text{eff,pr}} \frac{J_{\nu-1} Y_\nu + J_\nu Y_{\nu-1}}{|H_\nu|^2} \frac{|\delta\phi|_{\text{pr}}}{\sqrt{2}} \left[\sqrt{-\ln X_1} e^{i2\pi Y_1} - \sqrt{-\ln X_2} e^{i2\pi Y_2} \right] \end{aligned} \quad (6.6)$$

ここで Hankel 関数・Bessel 関数・Neumann 関数の引数は全て χ である*1。

6.3 発展方程式

6.4 アウトプット

6.5 インフラトンポテンシャル

6.6 インプットパラメータ

*1 計算は付録を参照

第 7 章

ファイル構造

7.1 時間発展

7.1.1 単一場インフレーション

スケール因子の発展方程式はその定義から

$$\frac{da}{d\tau} = a' \quad (7.1)$$

スケール因子の微分の発展方程式は、共形時間での Raychaudhuri 方程式 (2.13)

$$\frac{da'}{d\tau} = \frac{a^3}{6} \langle \rho - 3p \rangle \quad (7.2)$$

を用いる。ただしエネルギー密度と圧力は (??) より

$$\rho = \frac{(\phi')^2}{2a^2} - \frac{\phi \nabla^2 \phi}{2a^2} + V(\phi) \quad (7.3)$$

$$p = \frac{(\phi')^2}{2a^2} + \frac{\phi \nabla^2 \phi}{6a^2} - V(\phi) \quad (7.4)$$

ここで空間微分をラプラシアンのみにするために $(\nabla \phi)^2(\mathbf{n}) = -\phi(\mathbf{n}) \nabla^2 \phi(\mathbf{n})$ を仮定した [2]。また、格子シミュレーションでは場の値は背景量とゆらぎに分ける必要がないので、インフラトン場の値は $\phi(\tau, \mathbf{n})$ で表されることを用いた。

また Friedmann 方程式は、発展に伴って破れないかを確認しておく。具体的には以下の量が 1 に十分近い値を持つかどうか調べる (consistency condition)。

$$\frac{3(a')^2}{a^4 \langle \rho \rangle} = 1 \quad (7.5)$$

インフラトン場の発展方程式はその定義から

$$\frac{\partial \phi}{\partial \tau} = \phi' \quad (7.6)$$

インフラトン場の微分の発展方程式は、その作用 (??) から

$$\frac{\partial \phi'}{\partial \tau} = -2 \frac{a'}{a} \phi' + \nabla^2 \phi - a^2 \frac{dV}{d\phi} \quad (7.7)$$

7.1.2 アクシオン-U(1) インフレーション

独立な値は、単一場インフレーションの場合にゲージ場 $A_\mu(\mathbf{n})$ とその微分 $A'_\mu(\mathbf{n})$ が加わるので合計 $10N^3 + 2$ 個である。

スケール因子に関する発展方程式は (7.1) と (7.2) である。ここでゲージ場について $p = \frac{1}{3}\rho$ の関係があるので、 $\langle \rho - 3p \rangle$ にゲージ場は寄与しない。

インフラトン場の発展方程式は (7.6) であり、インフラトン場の微分の発展方程式は作用 (4.1) より

$$\frac{\partial \phi'}{\partial \tau} = -2 \frac{a'}{a} \phi' + \nabla^2 \phi - a^2 \frac{dV}{d\phi} - \frac{1}{a^2} \frac{\alpha}{f} (\nabla \times \mathbf{A}) \cdot (\mathbf{A}' - \nabla A_0) \quad (7.8)$$

ゲージ場の発展方程式は定義より

$$\frac{\partial A_\mu}{\partial \tau} = A'_\mu \quad (7.9)$$

ゲージ場の微分の発展方程式は Lorentz ゲージ条件

$$\partial^\mu A_\mu = 0 \quad \longrightarrow \quad A'_0 = \nabla \cdot \mathbf{A} \quad (7.10)$$

を課しつつ (4.1) より

$$\frac{\partial A'_0}{\partial \tau} = \nabla^2 A_0 + \frac{\alpha}{f} \nabla \phi \cdot (\nabla \times \mathbf{A}) \quad (7.11)$$

$$\frac{\partial \mathbf{A}'}{\partial \tau} = \nabla^2 \mathbf{A} + \frac{\alpha}{f} \phi' (\nabla \times \mathbf{A}) - \frac{\alpha}{f} (\nabla \phi) \times (\mathbf{A}' - \nabla A_0) \quad (7.12)$$

これらは (4.9) と (4.10) を非摂動論的に一般化した方程式である。

Friedmann 方程式の consistency condition (7.5) と同じく、Lorentz ゲージ条件も発展に伴って破れないか確かめておく。consistency condition は [2]

$$\frac{\partial^\mu A_\mu}{\sqrt{\sum_\rho |\partial^\rho A_\rho|^2}} = \frac{-A'_0 + \nabla \cdot \mathbf{A}}{\sqrt{|A'_0|^2 + |\partial_i A_i|^2}} \ll 1 \quad (7.13)$$

7.1.3 時間発展

最後に A.4 節の時間発展に関する量を書き直す。独立な値のうち時間微分で定義されているものは無次元化の影響を受ける。

$$a', \phi', A'_\mu \longrightarrow \check{a}, \check{\phi}, \check{A}_\mu \quad (7.14)$$

よって、スケール因子・インフラトン場・ゲージ場の発展方程式 (7.1)(7.6)(7.9) は

$$\frac{\partial a}{\partial \tau_{\text{pr}}} = \check{a} \quad (7.15)$$

$$\frac{\partial \phi}{\partial \tau_{\text{pr}}} = \check{\phi} \quad (7.16)$$

$$\frac{\partial A_\mu}{\partial \tau_{\text{pr}}} = \check{A}_\mu \quad (7.17)$$

となる。

Raychaudhuri 方程式 (7.2) は

$$\frac{\partial \check{a}}{\partial \tau_{\text{pr}}} = -s \frac{(\check{a})^2}{a} + \frac{a^3}{6} \langle \rho_{\phi, \text{pr}} - 3p_{\phi, \text{pr}} \rangle \quad (7.18)$$

ここでプログラム上におけるインフラトン場のエネルギー密度と圧力は

$$\rho_{\phi, \text{pr}} = \frac{(\check{\phi})^2}{2a^2} - \frac{\phi \nabla_{\text{pr}}^2 \phi}{2a^{2+2s}} + V(\phi)_{\text{pr}} \quad (7.19)$$

$$p_{\phi, \text{pr}} = \frac{(\check{\phi})^2}{2a^2} + \frac{\phi \nabla_{\text{pr}}^2 \phi}{6a^{2+2s}} - V(\phi)_{\text{pr}} \quad (7.20)$$

Friedmann 方程式の consistency condition(7.5) は

$$\frac{3(\check{a})^2}{a^4 \langle \rho_{\text{pr}} \rangle} = 1 \quad (7.21)$$

ただしエネルギー密度は単一場インフレーションの場合は (7.19) だが、アクシオン-U(1) インフレーションの場合はゲージ場の寄与

$$\rho_{A, \text{pr}} = \frac{1}{2a^{4+2s}} \left[\sum_i (a^s \check{A}_i - \partial_{\text{pr}, i} A_0)^2 + \frac{1}{2} \sum_{i, j} (\partial_{\text{pr}, i} A_j - \partial_{\text{pr}, j} A_i)^2 \right] \quad (7.22)$$

を加える。

単一場インフレーションにおけるインフラトン場の微分の発展方程式 (7.7) は

$$\frac{\partial \check{\phi}}{\partial \tau_{\text{pr}}} = -(s+2) \frac{\check{a}}{a} \check{\phi} + \frac{\nabla_{\text{pr}}^2 \phi}{a^{2s}} - a^2 \frac{dV_{\text{pr}}}{d\phi} \quad (7.23)$$

アクシオン-U(1) インフレーションにおけるインフラトン場の微分の発展方程式 (7.8) は

$$\frac{\partial \check{\phi}}{\partial \tau_{\text{pr}}} = -(s+2) \frac{\check{a}}{a} \check{\phi} + \frac{\nabla_{\text{pr}}^2 \phi}{a^{2s}} - a^2 \frac{dV_{\text{pr}}}{d\phi} - \frac{1}{a^{2+2s}} \frac{\alpha}{f} (\nabla_{\text{pr}} \times \mathbf{A}) \cdot (a^s \check{\mathbf{A}} - \nabla_{\text{pr}} A_0) \quad (7.24)$$

ゲージ場の微分の発展方程式 (7.11)(7.12) は

$$\frac{\partial \check{A}_0}{\partial \tau_{\text{pr}}} = -s \frac{\check{a}}{a} \check{A}_0 + \frac{\nabla_{\text{pr}}^2 A_0}{a^{2s}} + \frac{1}{a^{2s}} \frac{\alpha}{f} (\nabla_{\text{pr}} \phi) \cdot (\nabla_{\text{pr}} \times \mathbf{A}) \quad (7.25)$$

$$\frac{\partial \check{\mathbf{A}}}{\partial \tau_{\text{pr}}} = -s \frac{\check{a}}{a} \check{\mathbf{A}} + \frac{\nabla_{\text{pr}}^2 \mathbf{A}}{a^{2+2s}} + \frac{1}{a^s} \frac{\alpha}{f} \check{\phi} (\nabla_{\text{pr}} \times \mathbf{A}) - \frac{1}{a^{2s}} \frac{\alpha}{f} (\nabla_{\text{pr}} \phi) \times (a^s \check{\mathbf{A}} - \nabla_{\text{pr}} A_0) \quad (7.26)$$

Lorentz ゲージ条件の consistency condition (7.13) は

$$G \equiv \frac{-a^s \check{A}_0 + \nabla_{\text{pr}} \cdot \mathbf{A}}{\sqrt{a^{2s} |\check{A}_0|^2 + |\partial_{\text{pr},i} A_i|^2}} \quad (7.27)$$

7.2 アウトプット

7.2.1 背景量

単一場インフレーション

Friedmann 方程式の条件式 (7.21) は

$$E \equiv \frac{3(\check{a})^2}{a^4 \langle \rho_{\text{pr}} \rangle} \quad (7.28)$$

として計算し、この値が 1 からどの程度ずれているかで Friedmann 方程式が満たされているかを確認する。 E が 1 に十分近ければ、Hubble パラメータをエネルギー密度から計算した場合と (??) で計算した場合が等価となる。

スローロール・パラメータ ϵ は (??) より

$$\epsilon = \frac{3}{2} \left(1 + \frac{p}{\rho} \right) = \frac{3}{2} \left(1 + \frac{\langle p_{\text{pr}} \rangle}{\langle \rho_{\text{pr}} \rangle} \right) \quad (7.29)$$

エネルギー密度と圧力の表式は、単一場インフレーションの場合は (7.19)(7.20) を用い、アクシオン-U(1) インフレーションの場合はゲージ場の寄与 (7.22) を加える。

アクション-U(1) インフレーション

単一場インフレーションの場合に加えて相互作用に関する量とゲージ固定の条件式を出力する。有効結合定数 ξ は (4.18) より

$$\xi = \frac{\alpha \langle \dot{\phi} \rangle}{2fH} = \frac{\alpha}{2f} \frac{\langle \dot{\phi} \rangle}{\frac{H}{m}} = \frac{\alpha}{2f} \frac{a \langle \check{\phi} \rangle}{\check{a}} \quad (7.30)$$

また背景量へのゲージ場の影響を測る条件式 (??)(??) は

$$\frac{H^2}{26\pi \langle \dot{\phi} \rangle} \frac{e^{\pi\xi}}{\xi^{\frac{3}{2}}} = \frac{m \left(\frac{H}{m}\right)^2}{26\pi \frac{\langle \dot{\phi} \rangle}{m}} \frac{e^{\pi\xi}}{\xi^{\frac{3}{2}}} = \frac{ma^{s-3}(\check{a})^2}{26\pi \langle \check{\phi} \rangle} \left(\frac{\alpha}{2f} \frac{a \langle \check{\phi} \rangle}{\check{a}}\right)^{-\frac{3}{2}} \exp\left(\frac{\alpha}{2f} \frac{a \langle \check{\phi} \rangle}{\check{a}}\right) \quad (7.31)$$

$$\frac{H}{146} \frac{e^{\pi\xi}}{\xi^{\frac{3}{2}}} = \frac{ma^{s-2}\check{a}}{146} \left(\frac{\alpha}{2f} \frac{a \langle \check{\phi} \rangle}{\check{a}}\right)^{-\frac{3}{2}} \exp\left(\frac{\alpha}{2f} \frac{a \langle \check{\phi} \rangle}{\check{a}}\right) \quad (7.32)$$

で計算する。

ゲージ固定の条件式は (7.27) を用いる。

第 8 章

コードレビュー

第 III 部

Axion-U(1) インフレーション

第 9 章

セットアップと実装表式

本章では、Axion-U(1) インフレーションの Lattice シミュレーションで用いた詳細なセットアップと、実際に用いた実装表式を説明する。第 I 部で既に説明した内容については触れない。9.1 節で用いたセットアップのまとめを説明、9.2 節で初期条件の説明、9.3 節で実装する発展方程式の導出、9.4 節で出力する量の説明、9.5 節でインフラトンポテンシャルの計算、9.6 節でインプットすべきパラメータの説明をする。

ここでは概念的なセットアップと理論的な計算を記すため、具体的（あるいは技術的）なコードの解説は 10 章と 11 章を参照すべき。

9.1 セットアップのまとめ

9.2 初期条件

9.3 発展方程式

9.4 アウトプット

9.5 インフラトンポテンシャル

9.6 インプットパラメータ

ゲージ場

ゲージ場もインフラトンゆらぎと同様に初期条件を課す。ただし [2] に基づいて時間成分 A_0, A'_0 は 0 とする。空間成分は偏光モードに分解して

$$\mathbf{A}(\boldsymbol{\kappa}) = \epsilon_+(\boldsymbol{\kappa})A_+(\boldsymbol{\kappa}) + \epsilon_-(\boldsymbol{\kappa})A_-(\boldsymbol{\kappa}) \quad (9.1)$$

インフラトンゆらぎに解析解 (??) を用いたのと同じく、ゲージ場のモード関数には (4.20)(4.21) を用いる。よってゲージ場の初期条件は、偏光を表す $\lambda = \pm$ を用いて

$$A_\lambda(\boldsymbol{\kappa}) = \frac{1}{\sqrt{2}}|A_\lambda| \left[\sqrt{-\ln X_{\lambda,1}} e^{i2\pi Y_{\lambda,1}} + \sqrt{-\ln X_{\lambda,2}} e^{i2\pi Y_{\lambda,2}} \right] \quad (9.2)$$

$$|A_\lambda| = \frac{L^{\frac{3}{2}}}{dx^3} \sqrt{\frac{G_0^2(\lambda\xi, \chi) + F_0^2(\lambda\xi, \chi)}{2k_{\text{eff}}}} \quad (9.3)$$

ただし一様乱数 $X_{\lambda,1}, X_{\lambda,2}, Y_{\lambda,1}, Y_{\lambda,2}$ は偏光と $\boldsymbol{\kappa}$ 毎に異なる値で実装する。

一方、ゲージ場の微分は

$$\begin{aligned} A'_\lambda(\boldsymbol{\kappa}) = & \left[k_{\text{eff}} \sqrt{1 + \xi^2} \frac{G_0 G_1 + F_0 F_1}{G_0^2 + F_0^2} - k_{\text{eff}} \lambda \xi - H_{\text{ini}} \right] A_\lambda(\boldsymbol{\kappa}) \\ & - \frac{i}{\sqrt{2}} \frac{k_{\text{eff}}}{\sqrt{G_0^2 + F_0^2}} \frac{|A_\lambda|}{\sqrt{G_0^2 + F_0^2}} \left[\sqrt{-\ln X_{\lambda,1}} e^{i2\pi Y_{\lambda,1}} - \sqrt{-\ln X_{\lambda,2}} e^{i2\pi Y_{\lambda,2}} \right] \end{aligned} \quad (9.4)$$

ただし Coulomb 波動関数 G, F の引数は全て $(\lambda\xi, \chi)$ であり、一様乱数は同じ偏光と同じ

κ に対してゲージ場で用いた値と同じである。また、Coulomb 波動関数の微分漸化式

$$\frac{dg_0(\xi, x)}{dx} = \left(\frac{1}{x} + \xi \right) g_0(\xi, x) - \sqrt{1 + \xi^2} g_1(\xi, x), \quad g = F, G \quad (9.5)$$

と関係式

$$G_l(\xi, x) \frac{d}{dx} F_l(\xi, x) - F_l(\xi, x) \frac{d}{dx} G_l(\xi, x) = 1 \quad (9.6)$$

を用いた。

同様に、ゲージ場の初期条件 (9.2)(9.3)(9.4) は

$$A_{\lambda, \text{pr}}(\kappa) = A_\lambda(\kappa) = \frac{1}{\sqrt{2}} |A_\lambda|_{\text{pr}} \left[\sqrt{-\ln X_{\lambda,1}} e^{i2\pi Y_{\lambda,1}} + \sqrt{-\ln X_{\lambda,2}} e^{i2\pi Y_{\lambda,2}} \right] \quad (9.7)$$

$$|A_\lambda|_{\text{pr}} = |A_\lambda| = m \frac{L_{\text{pr}}^{\frac{3}{2}}}{dx_{\text{pr}}^3} \sqrt{\frac{G_0^2(\lambda\xi, \chi) + F_0^2(\lambda\xi, \chi)}{2k_{\text{eff,pr}}}} \quad (9.8)$$

$$\begin{aligned} \check{A}_\lambda(\kappa) = & \left[k_{\text{eff,pr}} \sqrt{1 + \xi^2} \frac{G_0 G_1 + F_0 F_1}{G_0^2 + F_0^2} - k_{\text{eff,pr}} \lambda \xi - H_{\text{ini,pr}} \right] A_{\lambda, \text{pr}}(\kappa) \\ & - \frac{i}{\sqrt{2}} \frac{k_{\text{eff,pr}}}{\sqrt{G_0^2 + F_0^2}} \frac{|A_\lambda|_{\text{pr}}}{\sqrt{G_0^2 + F_0^2}} \left[\sqrt{-\ln X_{\lambda,1}} e^{i2\pi Y_{\lambda,1}} - \sqrt{-\ln X_{\lambda,2}} e^{i2\pi Y_{\lambda,2}} \right] \end{aligned} \quad (9.9)$$

9.6.1 PBH 関係量

PBH 存在量 f_{PBH} を求めるには (3.7) より、形成される PBH の質量 M_{PBH} と形成時の存在量 β を求める必要がある。以下ではこれらを求めるのに用いた手法と、必要なインプットパラメータを記す。

PBH 質量

PBH 質量と共動的な運動量スケール k の関係 (3.4) を用いて求める。この式における k は共動的な地平線スケールの逆数 aH で評価されるが、格子シミュレーションではスケール因子の初期値を 1 と定義していて単純に $k = (aH)_{\text{pr}}$ として計算することはできない。そこで CMB スケールを参照スケールとして導入して

$$\begin{aligned} \frac{k}{k_{\text{CMB}}} &= \frac{aH}{(aH)_{\text{CMB}}} = e^{N_e(k) - N_{e, \text{CMB}}} \frac{H}{H_{\text{CMB}}} \\ &= e^{N_{e, \text{ini}} - N_{e, \text{CMB}} + \Delta N_e} \frac{H}{m} \left(\frac{H_{\text{CMB}}}{m} \right)^{-1} \end{aligned} \quad (9.10)$$

のように e-folds で表して計算する。ここで $N_{e,\text{ini}}$ は格子シミュレーション開始時の e-folds であり、 $N_{e,\text{CMB}}$ は CMB スケール k_{CMB} が地平線外に出た時の e-folds である。また ΔN_e は格子シミュレーション開始から、注目スケール k が地平線外に出るまでの e-folds である。格子上の有効運動量 k_{eff} が格子上で地平線外に出る時刻（つまり $k_{\text{eff,pr}} = (aH)_{\text{pr}}$ となる時刻）における $\Delta N_e, \frac{H}{m}$ を求めれば、 k_{eff} に対応する k が求められる。上式におけるインプットパラメータは $N_{e,\text{ini}} - N_{e,\text{CMB}}, \frac{H_{\text{CMB}}}{m}, k_{\text{CMB}}$ であり、前者2つは背景インフレーションの発展から求める。

次に格子シミュレーションにおいて、各 k_{eff} に対応する $\Delta N_e, \frac{H}{m}$ を求める手法を述べる。前述の通り $k_{\text{eff,pr}} = (aH)_{\text{pr}}$ となる時刻で $\Delta N_e, \frac{H}{m}$ を評価する。ただし格子シミュレーションでは、時間発展が離散化されているので直線近似を行う。具体的には、離散的な時間発展によって厳密に $k_{\text{eff,pr}} = (aH)_{\text{pr}}$ となる時刻を求めることはできないので、代わりに直前の時刻における $(\Delta N_{e,\text{pre}}, (aH)_{\text{pr,pre}})$ の値と直後の時刻における $(\Delta N_{e,\text{post}}, (aH)_{\text{pr,post}})$ の値を用いて、その2点間を直線で近似して求める。以下に直線近似を表す図を示す。 ΔN を求めるための式は

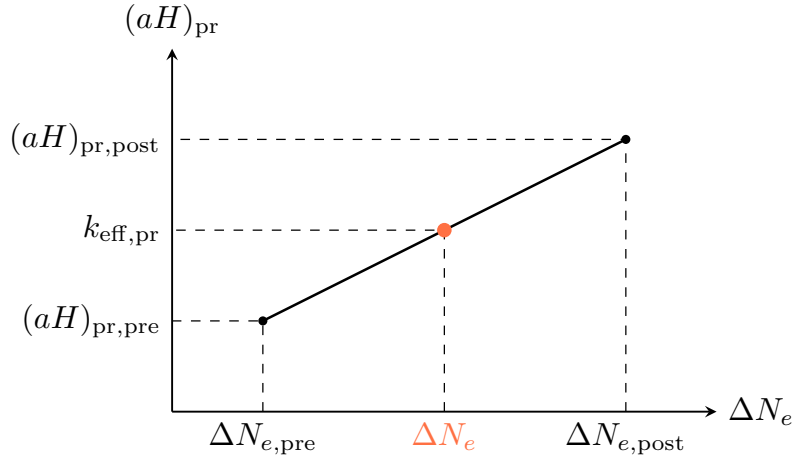


図 9.1 $k_{\text{eff,pr}} = (aH)_{\text{pr}}$ となる時刻 ΔN_e を求めるための直線近似

$$\Delta N_e = \Delta N_{e,\text{pre}} + \frac{\Delta N_{e,\text{post}} - \Delta N_{e,\text{pre}}}{(aH)_{\text{pr,post}} - (aH)_{\text{pr,pre}}} [k_{\text{eff,pr}} - (aH)_{\text{pr,pre}}] \quad (9.11)$$

である。同様に $\frac{H}{m}$ についても直線近似を用いれば、上で求めた ΔN から

$$\frac{H}{m} = \left(\frac{H}{m} \right)_{\text{pre}} + \frac{\left(\frac{H}{m} \right)_{\text{post}} - \left(\frac{H}{m} \right)_{\text{pre}}}{\Delta N_{e,\text{post}} - \Delta N_{e,\text{pre}}} [\Delta N_e - \Delta N_{e,\text{pre}}] \quad (9.12)$$

と計算できる。このようにして求めた $\Delta N_e, \frac{H}{m}$ とインプットパラメータを用いれば、 k_{eff} に対応する共動的運動量スケール k が求められ、(3.4) を用いて k_{eff} に対応する PBH 質量を求めることができる。

PBH 形成時の存在量

形成時の存在量 β は、密度ゆらぎがガウス分布に従うと仮定し (3.9) を用いて求める。ここで γ, δ_c はインプットパラメータであり、 σ は (3.14) により無次元曲率スペクトルから計算できる。ガウス型の窓関数を取ると

$$\begin{aligned}\sigma^2(k) &= \frac{16}{81} \int_0^\infty d \ln q \left(\frac{q}{k} \right)^4 \exp \left(-\frac{q^2}{k^2} \right) P_{\mathcal{R}}(q) \\ &= \frac{16}{81} \int_0^\infty d \left(\frac{q}{k} \right) \left(\frac{q}{k} \right)^3 \exp \left(-\frac{q^2}{k^2} \right) P_{\mathcal{R}}(q)\end{aligned}$$

格子上の有効運動量で表すと

$$\frac{q}{k} = \frac{q_{\text{eff,pr}}}{k_{\text{eff,pr}}} \quad (9.13)$$

であり、この関数としての無次元曲率スペクトル $P_{\mathcal{R}} \left(\frac{q}{k} \right)$ を用いれば

$$\sigma^2(k_{\text{eff}}) = \frac{16}{81} \int_0^\infty dx x^3 e^{-x^2} P_{\mathcal{R}}(x) \quad (9.14)$$

で計算できる。ただし $x = \frac{q}{k}$ に変数変換した。

格子シミュレーションでは調べることができる運動量スケールは有限で、積分範囲の上限と下限を設ける

$$\sigma^2(k_{\text{eff}}) = \frac{16}{81} \int_{x_{\min}}^{x_{\max}} dx x^3 e^{-x^2} P_{\mathcal{R}}(x) \quad (9.15)$$

ただし x_{\max} は $q_{\text{eff,max}}$ に対応し、 x_{\min} は $q_{\text{eff,min}} \approx \frac{2\pi}{L}$ に対応する。PBH 存在量 f_{PBH} が大きくなる運動量スケールは無次元曲率スペクトルのピーク付近 $k_{\text{eff}} \approx \tilde{k}_{\text{eff}}$ であるので

$$\sigma^2(k_{\text{eff}} \approx \tilde{k}_{\text{eff}}) = \frac{16}{81} \int_{x_{\min}}^{x_{\max}} dx x^3 e^{-x^2} P_{\mathcal{R}}(x) \quad (9.16)$$

が重要な計算である。この場合 $P_{\mathcal{R}}(x)$ は $x = 1$ 付近にピークを持ち、また被積分関数によって $x \sim 1$ 以外の寄与は指数関数的に減衰する。よって積分範囲が $x \sim 1$ を含んでいれば、積分範囲に上限と下限を設ける近似が良い精度で保証される。

第 10 章

ファイル構造

第 11 章

コードレビュー

第Ⅳ部

利用に関して

第 12 章

利用上の注意事項

本章では Lattice シミュレーションコード「SFInflation」「Axion-U(1)」を利用する際の注意事項を説明する。12.1 節では使用するための権限について、12.2 節ではプログラム上の時間間隔が満たすべき条件、12.3 節ではシミュレーションするために必要なメモリを説明する。適宜追加予定。

12.1 使用権限

本節ではシミュレーションコードを使用する際の権限を記しておく。私は LATTICEEASY[1] のコードを書き換えただけなので、その権限は LATTICEEASY 開発者に帰属する（[日本語あってる？](#)）。その内容は

LATTICEEASY consists of the C++ files "latticeeasy.cpp," "initialize.cpp," "evolution.cpp," "output.cpp," "latticeeasy.h," "parameters.h,". (The distribution also includes the file fteasy.cpp but this file is distributed separately and therefore not considered part of the LATTICEEASY distribution in what follows.) LATTICEEASY is free. We are not in any way, shape, or form expecting to make money off of these routines. We wrote them for the sake of doing good science and we're putting them out on the Internet in case other people might find them useful. Feel free to download them, incorporate them into your code, modify them, translate the comment lines into Swahili, or whatever else you want. What we do want is the following:

- 1) Leave this notice (i.e. this entire paragraph beginning with "LATTICEEASY consists of..." and ending with our email addresses) in with the code wherever you put it. Even if you're just using it in-house in your department, business, or wherever else we would like these credits to remain with it. This is partly so that people can...
- 2) Give us feedback. Did LATTICEEASY work great for you and help your work? Did you hate it? Did you find a way to improve it, or translate it into another programming language? Whatever the case might be, we would love to hear about it. Please let us know at the email address below.
- 3) Finally, insofar as we have the legal right to do so we forbid you to make money off of this code without our consent. In other words if you want to publish these functions in a book or bundle them into commercial software or anything like that contact us about it first. We'll probably say yes, but we would like to reserve that right.

For any comments or questions you can reach us at

gfelder@email.smith.edu

Igor.Tkachev@cern.ch

である。曰く、「LATTICEEASY のコードを自由に書き換えて良いが、上の文章をコード中に記載しておくこと。また、書き換えたコードなどをお金儲けに使いたければ必ず連絡し許可を取れ。」だそうだ。

また 2.1.3 節で説明した FFT を実行するためのコード「ffteasy.cpp」にも同様に

FFTEASY consists of the four C functions `fftc1`, `fftcn`, `fftr1`, and `fftrn`. FFTEASY is free. I am not in any way, shape, or form expecting to make money off of these routines. I wrote them because I needed them for some work I was doing and I'm putting them out on the Internet in case other people might find them useful. Feel free to download them, incorporate them into your code, modify them, translate the comment lines into Swahili, or whatever else you want. What I do want is the following:

- 1) Leave this notice (i.e. this entire paragraph beginning with "FFTEASY consists of..." and ending with my email address) in with the code wherever you put it. Even if you're just using it in-house in your department, business, or wherever else I would like these credits to remain with it. This is partly so that people can...
- 2) Give me feedback. Did FFTEASY work great for you and help your work? Did you hate it? Did you find a way to improve it, or translate it into another programming language? Whatever the case might be, I would love to hear about it. Please let me know at the email address below.
- 3) Finally, insofar as I have the legal right to do so I forbid you to make money off of this code without my consent. In other words if you want to publish these functions in a book or bundle them into commercial software or anything like that contact me about it first. I'll probably say yes, but I would like to reserve that right.

For any comments or questions you can reach me at
gfelder@email.smith.edu.

の文章をコード中に残しておく必要がある。

このように、LATTICEASY あるいは私の書いたコードに対する利用条件はおおよそ「MIT ライセンス」のようなもので、上で記載した文章をコード中に入れておけば自由に利用できる（ただし金儲けのためには直接的な許可を得る必要あり）。ちなみに MIT ライセンスとは、マサチューセッツ工科大学で作成された代表的な寛容型オープンソースライセンスであり、以下の文章を記載すれば自由にコードを利用できるという、緩い利用条件である。

Copyright (c) <YEAR> <COPYRIGHTHOLDER>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

ただし <YEAR> には著作権発生年を記入し、<COPYRIGHTHOLDER> には著作権保持者名を記入する。

12.2 コーラン条件

12.3 必要メモリ

第 13 章

利用方法

13.1 必要なアプリケーション

13.2 Lattice シミュレーションの実行

13.3 グラフの描画

付録

付録 A

計算過程の明記

A.1 第 I 部

・ (1.7)

$$\begin{aligned}
 \ddot{a} &= \frac{d}{d\tau} \dot{a} \\
 &= ma^s \frac{d}{d\tau_{\text{pr}}} (ma^s a') \\
 &= m^2 a^s (sa^{s-1} a' a' + a^s a'') \\
 &= m^2 a^{2s} \left[a'' + s \frac{(a')^2}{a} \right]
 \end{aligned}$$

・ (1.8)

$$\begin{aligned}
 \ddot{f} &= \frac{d}{d\tau} \dot{f} \\
 &= ma^s \frac{d}{d\tau_{\text{pr}}} (ma^s f') \\
 &= m^2 a^s (sa^{s-1} a' f' + a^s f'') \\
 &= m^2 a^{2s} \left[f'' + s \frac{a'}{a} f' \right]
 \end{aligned}$$

・ (2.2) と (2.3) の DFT が逆変換であること

$$\begin{aligned}
 f(\mathbf{n}) &= \frac{1}{N^3} \sum_{\mathbf{m}} \left[\sum_{\tilde{\mathbf{n}}} f(\tilde{\mathbf{n}}) e^{-\frac{2\pi i}{N} \mathbf{m} \cdot \tilde{\mathbf{n}}} \right] e^{\frac{2\pi i}{N} \mathbf{m} \cdot \mathbf{n}} \\
 &= \frac{1}{N^3} \sum_{\tilde{\mathbf{n}}} f(\tilde{\mathbf{n}}) \sum_{\mathbf{m}} \exp \left[\frac{2\pi i}{N} \mathbf{m} \cdot (\mathbf{n} - \tilde{\mathbf{n}}) \right] \\
 &= \frac{1}{N^3} \sum_{\tilde{\mathbf{n}}} f(\tilde{\mathbf{n}}) N^3 \delta_{\mathbf{n}, \tilde{\mathbf{n}}} \\
 &= f(\mathbf{n})
 \end{aligned}$$

・ (2.5)

$$\begin{aligned}
 f^*(\boldsymbol{\kappa}) &= \sum_{\mathbf{n}} f^*(\mathbf{n}) e^{\frac{2\pi i}{N} \mathbf{m} \cdot \mathbf{n}} \\
 &= \sum_{\mathbf{n}} f(\mathbf{n}) e^{\frac{2\pi i}{N} \mathbf{m} \cdot \mathbf{n}} \\
 &= \sum_{\mathbf{n}} f(\mathbf{n}) e^{-\frac{2\pi i}{N} (-\mathbf{m}) \cdot \mathbf{n}} \\
 &= f(-\boldsymbol{\kappa})
 \end{aligned}$$

ただし $\boldsymbol{\kappa} = \frac{2\pi}{L} \frac{N}{2}$ に対しては

$$f^*(\boldsymbol{\kappa}) = \sum_{\mathbf{n}} f(\mathbf{n}) e^{i\pi n_i} = \sum_{\mathbf{n}} f(\mathbf{n}) e^{-i\pi n_i} = \sum_{\mathbf{n}} f(\mathbf{n})$$

で仮想的な $m_i = -\frac{N}{2}$ が対応する。もっと言うとなりフーリエモードのうち $\boldsymbol{\kappa} = (0, 0, 0), (0, 0, \frac{N}{2}), (0, \frac{N}{2}, 0), (\frac{N}{2}, 0, 0), (0, \frac{N}{2}, \frac{N}{2}), (\frac{N}{2}, 0, \frac{N}{2}), (\frac{N}{2}, \frac{N}{2}, 0), (\frac{N}{2}, \frac{N}{2}, \frac{N}{2})$ の 8 つのみが実数で $f^*(\boldsymbol{\kappa}) = f(\boldsymbol{\kappa})$ となる。

・ (3.9) の分散が $\mathcal{P}_{\delta\phi}$ になること

$$\begin{aligned}
 \sigma^2 &= \int_0^\infty dX X^2 \left(\frac{\mathcal{P}_{\delta\phi}}{2} \right)^{-1} X \exp \left[-\frac{1}{2} \left(\frac{\mathcal{P}_{\delta\phi}}{2} \right)^{-1} X^2 \right] \\
 &= \left[-X^2 \exp \left[-\frac{1}{2} \left(\frac{\mathcal{P}_{\delta\phi}}{2} \right)^{-1} X^2 \right] \right]_0^\infty + 2 \int_0^\infty dX X \exp \left[-\frac{1}{2} \left(\frac{\mathcal{P}_{\delta\phi}}{2} \right)^{-1} X^2 \right] \\
 &= 2 \left[-\left(\frac{\mathcal{P}_{\delta\phi}}{2} \right) \exp \left[-\frac{1}{2} \left(\frac{\mathcal{P}_{\delta\phi}}{2} \right)^{-1} X^2 \right] \right]_0^\infty \\
 &= \mathcal{P}_{\delta\phi}
 \end{aligned}$$

・ (3.10) の複素ガウス分布で $|\delta\phi(k)|$ が (3.9) の Rayleigh 分布に従うこと

$$|\delta\phi(k)|^2 = (\text{Re}[\delta\phi(k)])^2 + (\text{Im}[\delta\phi(k)])^2 \quad , \quad \frac{\text{Im}[\delta\phi(k)]}{\text{Re}[\delta\phi(k)]} = \tan \theta$$

で極座標のように変換すると

$$\begin{aligned}
 &\int_{-\infty}^\infty d\text{Re}[\delta\phi(k)] \int_{-\infty}^\infty d\text{Im}[\delta\phi(k)] \frac{1}{\pi \mathcal{P}_{\delta\phi}} \exp \left[-\frac{|\delta\phi(k)|^2}{\mathcal{P}_{\delta\phi}} \right] \\
 &= \int_0^\infty d|\delta\phi(k)| \int_0^{2\pi} d\theta |\delta\phi(k)| \frac{1}{\pi \mathcal{P}_{\delta\phi}} \exp \left[-\frac{|\delta\phi(k)|^2}{\mathcal{P}_{\delta\phi}} \right] \\
 &= \int_0^\infty d|\delta\phi(k)| \left(\frac{\mathcal{P}_{\delta\phi}}{2} \right)^{-1} |\delta\phi(k)| \exp \left[-\frac{1}{2} \left(\frac{\mathcal{P}_{\delta\phi}}{2} \right)^{-1} |\delta\phi(k)|^2 \right]
 \end{aligned}$$

A.2 第 II 部

•

$$\frac{dg_\nu(x)}{dx} = g_{\nu-1}(x) - \frac{\nu}{x}g_\nu(x), \quad g = J, Y, H$$

これを用いてモード関数（両方向の波それぞれ）の微分は

$$\begin{aligned} \partial_\tau \delta\phi(k) &= \partial_\tau \left[\frac{\sqrt{-\pi\tau}}{2a} H_\nu^{(1,2)}(-k\tau) \right] \\ &= \partial_\tau \frac{\sqrt{-\pi\tau}}{2a} \sqrt{J_\nu^2(-k\tau) + Y_\nu^2(-k\tau)} \exp \left[\pm i \arctan \left(\frac{Y_\nu(-k\tau)}{J_\nu(-k\tau)} \right) \right] \\ &= \left[\frac{-1}{2(-\tau)} - \frac{\partial_\tau a}{a} \right] \frac{\sqrt{-\pi\tau}}{2a} \sqrt{J_\nu^2 + Y_\nu^2} e^{\pm i\theta} \\ &\quad + \frac{\sqrt{-\pi\tau}}{2a} (-k) \frac{\partial}{\partial x} \sqrt{J_\nu^2(x) + Y_\nu^2(x)} \exp \left[\pm i \arctan \left(\frac{Y_\nu(x)}{J_\nu(x)} \right) \right] \\ &= \left[\frac{1}{2\tau} - \mathcal{H} \right] \frac{\sqrt{-\pi\tau}}{2a} \sqrt{J_\nu^2 + Y_\nu^2} e^{\pm i\theta} \\ &\quad - k \frac{\sqrt{-\pi\tau}}{2a} \left[\frac{J_\nu \partial_x J_\nu + Y_\nu \partial_x Y_\nu}{J_\nu^2 + Y_\nu^2} \pm i \frac{J_\nu \partial_x Y_\nu - Y_\nu \partial_x J_\nu}{\left(1 + \frac{Y_\nu^2}{J_\nu^2}\right) J_\nu^2} \right] \sqrt{J_\nu^2 + Y_\nu^2} e^{\pm i\theta} \\ &= \left[\frac{1}{2\tau} - \mathcal{H} - k \frac{J_\nu \left(J_{\nu-1} - \frac{\nu}{-k\tau} J_\nu \right) + Y_\nu \left(Y_{\nu-1} - \frac{\nu}{-k\tau} Y_\nu \right)}{J_\nu^2 + Y_\nu^2} \right] \delta\phi(k) \\ &\quad \mp ik \frac{J_\nu Y_{\nu-1} - J_{\nu-1} Y_\nu}{J_\nu^2 + Y_\nu^2} \delta\phi(k) \\ &= \left[\frac{1}{2\tau} - \mathcal{H} - \frac{\nu}{\tau} - k \frac{J_\nu J_{\nu-1} + Y_\nu Y_{\nu-1}}{J_\nu^2 + Y_\nu^2} \pm ik \frac{J_{\nu-1} Y_\nu + J_\nu Y_{\nu-1}}{J_\nu^2 + Y_\nu^2} \right] \delta\phi(k) \end{aligned}$$

初期条件と Lattice 上の有効運動量を考慮すると

$$\begin{aligned} \partial_\tau \delta\phi(k) &= \left[-\frac{H_{\text{ini}}}{2} - H_{\text{ini}} + \nu H_{\text{ini}} - k_{\text{eff}} \frac{J_\nu J_{\nu-1} + Y_\nu Y_{\nu-1}}{|H_\nu|^2} \right. \\ &\quad \left. \pm ik_{\text{eff}} \frac{J_{\nu-1} Y_\nu + J_\nu Y_{\nu-1}}{|H_\nu|^2} \right] \delta\phi(k) \\ &= \left[\left(\nu - \frac{3}{2} \right) H_{\text{ini}} - k_{\text{eff}} \frac{J_\nu J_{\nu-1} + Y_\nu Y_{\nu-1}}{|H_\nu|^2} \pm ik_{\text{eff}} \frac{J_{\nu-1} Y_\nu + J_\nu Y_{\nu-1}}{|H_\nu|^2} \right] \delta\phi(k) \end{aligned}$$

リスケーリングと等方的初期条件をを考慮して最終的に

$$\begin{aligned} \delta\phi'(\boldsymbol{\kappa}) = & \left[\left(\nu - \frac{3}{2} \right) a'_{\text{ini}} - k_{\text{eff,pr}} \frac{J_\nu J_{\nu-1} + Y_\nu Y_{\nu-1}}{|H_\nu|^2} \right] \delta\phi(\boldsymbol{\kappa}) \\ & + i k_{\text{eff,pr}} \frac{J_{\nu-1} Y_\nu + J_\nu Y_{\nu-1}}{|H_\nu|^2} \frac{|\delta\phi|_{\text{pr}}}{\sqrt{2}} \left[\sqrt{-\ln X_1} e^{i2\pi Y_1} - \sqrt{-\ln X_2} e^{i2\pi Y_2} \right] \end{aligned}$$

参考文献

- [1] G. Felder, I. Tkachev, *LATTICEEASY: A Program for Lattice Simulations of Scalar Fields in an Expanding Universe*, [hep-ph/0011159]
- [2] A. Caravano, *Simulating the Inflationary Universe: from single-field to the axion- $U(1)$ model*, [2209.13616]
- [3] A. Caravano et al., *Lattice Simulations of Inflation*, [2102.06378]
- [4] A. Caravano et al., *Lattice simulations of Abelian gauge fields coupled to axions during Inflation*, [2110.10695]
- [5] A. Caravano et al., *Lattice Simulations of Axion- $U(1)$ Inflation*, [2204.12874]
- [6] D. G. Figueroa, *The art of simulating the early Universe – Part I*, [2006.15122]
- [7] R. Sharma et al., *Lattice simulations of axion- $U(1)$ inflation: gravitational waves, magnetic fields, and black holes*, [2411.04854]