MultiModeCode: An efficient numerical solver for multifield inflation

Layne C. Price, Jonathan Frazer, Jiajun Xu, Hiranya V. Peiris, and Richard Easther

1Department of Physics, University of Auckland
Private Bag 92019, Auckland, New Zealand
2Department of Theoretical Physics, University of the Basque Country, UPV/EHU
48040 Bilbao, Spain
3IKERBASQUE, Basque Foundation for Science
48011 Bilbao, Spain
4Department of Physics, University of Wisconsin-Madison
Madison, WI 53706, USA
5Department of Physics and Astronomy, University College London
London WC1E 6BT, UK
E-mail: lpri691@aucklanduni.ac.nz, j.frazer@ucl.ac.uk, jiajun.xu@gmail.com, h.peiris@ucl.ac.uk, r.easther@auckland.ac.nz

Abstract. We present MultiModeCode, a Fortran 95/2000 package for the numerical exploration of multifield inflation models. This program facilitates efficient Monte Carlo sampling of prior probabilities for inflationary model parameters and initial conditions and is the first publicly available code that can efficiently generate large sample-sets for inflation models with \(O(100)\) fields. The code numerically solves the equations of motion for the background and first-order perturbations of multi-field inflation models with canonical kinetic terms and arbitrary potentials, providing the adiabatic, isocurvature, and tensor power spectra at the end of inflation. For models with sum-separable potentials MultiModeCode also computes the slow-roll prediction via the \(\delta N\) formalism for easy model exploration and validation. We pay particular attention to the isocurvature perturbations as the system approaches the adiabatic limit, showing how to avoid numerical instabilities that affect some other approaches to this problem. We demonstrate the use of MultiModeCode by exploring a few toy models. Finally, we give a concise review of multifield perturbation theory and a user’s manual for the program.

1Available at www.modecode.org.
1 Introduction

Many simple models of inflation adeptly reproduce the observed properties of the primordial cosmological perturbations [1–4], predicting a nearly scale-invariant power spectrum and minimal amounts of primordial non-Gaussianity. In the slow-roll, single-field paradigm the predictions of a given model are easily determined as an algebraic function of the field’s potential $V$ and its derivatives in terms of a hierarchy of slow-roll parameters. The resulting observables are simple to compute and easy to interpret.

However, relaxing any of the basic assumptions of the slow-roll, single-field models complicates this simple analysis. In particular, for many inflationary scenarios (e.g., multifield inflation, gauge inflation, and non-minimal couplings), the background and mode equations are complex systems of coupled, nonlinear ODEs, making analysis difficult in all but a few cases. Furthermore, while slow-roll, single-field inflation is a simple and easily understood model, it may not necessarily be considered natural in the context of high-energy theories. For example, low energy effective theories derived from string theory generically contain hundreds of scalar fields with complicated interactions, and many theories consider non-minimal couplings to the Ricci scalar (for a recent review, see Ref. [5]). While analytical studies have been able to overcome subsets of these problems, most of the techniques that have been used are situation-specific, which limits their applicability to novel models.
While significant progress can be made in the slow-roll limit, only numerical techniques can explore the full predictions of more complex inflation models. Even in the purely homogeneous limit, numerically solving the nonlinear Klein–Gordon equation for the homogeneous background fields reveals many interesting features that do not arise in slow-roll analyses, e.g., sensitivity to initial conditions [6–9]. These complications lead naturally to the numerical exploration of inflationary models.

In this paper we present and describe MultiModeCode,\(^1\) an efficient Fortran 95/2000 package that numerically solves the equations of motion for the background fields and the first-order perturbations for multifield inflation models in which the fields have canonical kinetic terms and are minimally coupled to gravity. MultiModeCode calculates the adiabatic, tensor, and various isocurvature power spectra as a function of scale \(k\), but does not evaluate higher order correlators. If the potential is sum-separable, MultiModeCode uses the solution to the background equations of motion to evaluate the slow-roll \(\delta N\) predictions for the scalar and tensor power spectra and their derivatives near the pivot scale \(k_\star\), also giving the slow-roll results for \(n_s\), \(r\), \(f_{NL}\), etc. The code has been extensively tested with various compilers, including the open-source GNU Fortran compiler.

Several numerical codes have been developed to study single-field models [10–18]. Here, we build on ModeCode [19–21], which was developed to test single-field inflation models and interfaced with tools such as CAMB [22], CosmoMC [23], and MultiNest [24]. ModeCode was designed for the Bayesian analysis of inflation and used by the Planck collaboration [4] to obtain the posterior probabilities and marginal likelihoods for inflation models. Moving to the multifield case significantly increases the numerical demands on the solver, and puts a premium on efficiency due to the much greater computational resources required by these analyses. A few codes exist to analyze multifield models, but the publicly available codes are inadequate for models with many fields and arbitrary potentials. Notably, Pyflation [25–28] is an object-oriented Python code that uses the same method we employ here for solving the perturbation equations, but cannot easily generate large samples due to the speed constraints imposed by a dynamic programming language.

This significant extension to ModeCode can be used to study the power spectra of analytically intractable multifield inflationary potentials, and to explore the generic predictions of complex models by marginalizing over large numbers of possible parameters. Complementing currently available codes [25–28], MultiModeCode specializes in obtaining large Monte Carlo samples of initial conditions and parameter prior probabilities. To help users familiarise themselves with MultiModeCode the package includes initial conditions priors used in Refs. [8, 29, 30]. The ability of this code to efficiently generate large Monte Carlo samples has permitted studies of the generic predictions of multifield inflation models with more than 100 fields [30, 31].

In practice, the code can simulate the evolution of the mode equations for \(\mathcal{O}(10^2)\) fields,\(^2\) but will become inefficient for significantly more fields due to the increasing dimensionality of the system, which increases with the number of fields as \(\mathcal{O}(N_f^2)\). However, it can efficiently sample the evolution of the background equations of motion for at least \(\mathcal{O}(10^3)\) fields. While solving just the background equations allows the exploration of background dynamics for such a large number of fields, if the model is sum-separable, then it will also give the slow-roll predictions for the adiabatic curvature power spectrum, as well as \(f_{NL}\) and \(\tau_{NL}\), in terms of the \(\delta N\) approximation. This should be valid when the fields are much lighter than \(H\) at

\(^1\)Publicly available at www.modecode.org.

\(^2\)Estimates regarding field number are based on \(N_f\)-quadratic inflation, which is not numerically intensive.
horizon crossing and slow-roll holds throughout the duration of inflation. MultiModeCode is released with several example models already implemented and it is straightforward to add to this number.

In §5, we demonstrate the features of MultiModeCode with an $N_f$–flation potential with a sharp step, which we parametrize by a hyperbolic tangent function, following Refs. [10, 32]. We show that, in addition to oscillatory features in the adiabatic curvature power spectrum that are expected from the single-field analysis [10, 33], with more than one field there are also oscillatory features in the isocurvature spectra, which might result in non-trivial evolution of the power spectrum after inflation. We also show that the numerical computation of isocurvature modes results in an inherent numerical instability, since some definitions of isocurvature perturbations involve computing the difference between two quantities that are of the same order of magnitude. This induces a dominant numerical error when these two quantities begin to approach the adiabatic limit. We overcome this problem by implementing a modified definition of isocurvature perturbations [30], which is numerically stable to many more orders of magnitude than some alternative definitions. We also implement a geometrical optics indicator of isocurvature evolution as first presented in Ref. [34]. While this measure only relies on background quantities and also does not suffer from instabilities, as implemented here it does not provide an absolute value of isocurvature, only an indicator of its growth or decay.

Finally, in §3 we provide a concise review of multifield perturbation theory with the aim of dispelling misconceptions that exist about this topic, which the enlightened reader can skip.

2 Features of MultiModeCode

We begin by highlighting some of the useful characteristics of MultiModeCode.

**Speed:** The purpose of MultiModeCode is to provide a fast and efficient solver that is well-tested and can be applied to a wide range of possible inflationary scenarios. MultiModeCode is written in Fortran 95/2000, increasing its capabilities relative to existing codes [25–28] and making it tractable to investigate models with many fields or to obtain large Monte Carlo samples from a model’s parameter space. In particular, prototype versions of this program were used in Refs. [30, 31] to analyze large samples of 100-field $N_f$-monomial inflation.

**Generality:** The code facilitates Bayesian approaches to studying inflation, where the model’s parameters are drawn from prior probabilities from which we can compute a probability distribution for specified observable associated with the model. We consider simple situations, e.g., evolving a model given fixed model parameters and initial conditions, as sub-cases of the more general Bayesian framework. To facilitate the use of general priors we have implemented the sampling routines in modules which are simple to adapt and restructure for the user’s purposes.

**Robustness:** The program exits gracefully when encountering fatal errors of either a technical or cosmological nature, while also catching specific errors that might only affect one particular configuration of the model. We have extensively checked the program output on various Macintosh and Linux machines with both the gfortran and ifort compilers, and include both a fourth-order Runge-Kutta integrator and an implicit backward-difference formula method, which is suitable for stiff problems.
Statistics: **MultiModeCode** provides pivot-scale observables, summarized in Table 1 and can sample the adiabatic and isocurvature power spectra as a function of scale $k$. We have implemented a variety of numerically stable indicators of the amount of isocurvature present in the system.

**Slow-roll comparison:** If the potential $V$ is sum-separable, **MultiModeCode** can also calculate observables using the $\delta N$ approximation, which assumes slow-roll. Since these quantities rely only on solutions of the background equations of motion they are efficient and simple to calculate, scaling with the number of fields as $O(N_f)$. Consequently, if the model is well-described by the slow-roll approximation between horizon crossing and the end of inflation, computing observables in the $\delta N$ formalism is efficient and easy.

3 A brief review of multifield perturbation theory

We begin with a short review of first-order, non-interacting multifield perturbation theory before describing **MultiModeCode** and the dynamics of many-field inflation. There are some substantial differences between single-field and multifield inflation, which we highlight in Section 3.1. Table 1 gives a list of the pivot-scale observables that **MultiModeCode** computes.

There are a few excellent reviews of this topic [26, 35–37] and we particularly recommend Refs. [38, 39] for more information. We first present the nuts-and-bolts of the mode function approach to first-order, multifield perturbations, which is implemented in **MultiModeCode**. Then we describe the widely-used $\delta N$-formalism, which has also been implemented for ease of use and for comparison to the perturbation solutions.

3.1 The highlights

Multifield inflation differs from the single field case in the following important respects.

**Isocurvature:** Multifield inflation generally permits both adiabatic and isocurvature perturbations. Adiabatic perturbations are related by a gauge transformation to the curvature perturbation on comoving hypersurfaces $R$, while isocurvature perturbations are entropic perturbations between different matter components on flat hypersurfaces. In single-field inflation there is only one matter component, so there are only adiabatic perturbations.

**Super-horizon evolution:** Isocurvature perturbations source adiabatic perturbations, causing them to evolve even on super-horizon scales. While this can generate novel signatures such as non-Gaussianity, this can also be problematic for comparing the predictions of a model with observation: unless isocurvature modes decay into an adiabatic limit before the end of inflation, the curvature perturbation does not become conserved and is thus sensitive to post-inflationary physics.

**The two-index mode function:** With more than one field, either (a) the direct interaction between fields or (b) the gravity-mediated interaction will mix the particle creation and annihilation operators as a function of time [38]. Instead of a single index mode function, we therefore need to solve for a mode matrix $\psi_{IJ}$, where $\delta \phi_I = \psi_{IJ} a^J$, for $N_f$ annihilation operators $a^J$. 
### Power spectra (PS) Type Reference

<table>
<thead>
<tr>
<th>PS Type (k)</th>
<th>Type Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathcal{P}_R(k) )</td>
<td>Adiabatic scalar spectrum</td>
<td>Eq. (3.23)</td>
</tr>
<tr>
<td>( \mathcal{P}_S(k) )</td>
<td>Isocurvature spectrum</td>
<td>Eq. (3.25)</td>
</tr>
<tr>
<td>( \mathcal{P}_{\delta P,\text{nad}}(k) )</td>
<td>Non-adiabatic pressure spectrum</td>
<td>Eq. (3.33)</td>
</tr>
<tr>
<td>( \mathcal{P}_{\text{ent}}(k) )</td>
<td>Entropic spectrum</td>
<td>Eq. (3.36)</td>
</tr>
<tr>
<td>( \mathcal{P}_{RS}(k) )</td>
<td>Adiabatic–non-adiab. cross spectrum</td>
<td>Eq. (3.26)</td>
</tr>
<tr>
<td>( \mathcal{P}_h(k) )</td>
<td>Tensor spectrum</td>
<td>–</td>
</tr>
</tbody>
</table>

### Observable at \( k^* \)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A_s )</td>
<td>Scalar amplitude ( \mathcal{P}_R(k_s) )</td>
<td>–</td>
</tr>
<tr>
<td>( A_{\text{iso}} )</td>
<td>Isocurvature ampl. ( \mathcal{P}_S(k_s) )</td>
<td>–</td>
</tr>
<tr>
<td>( A_{\text{P, nad}} )</td>
<td>Non-adiab. pressure ampl. ( \mathcal{P}_{\delta P,\text{nad}}(k_s) )</td>
<td>–</td>
</tr>
<tr>
<td>( A_{\text{ent}} )</td>
<td>Entropy ampl. ( \mathcal{P}_{\text{ent}}(k_s) )</td>
<td>–</td>
</tr>
<tr>
<td>( A_{\text{Cross}} )</td>
<td>Cross spectra ampl. ( \mathcal{P}_{RS}(k_s) )</td>
<td>–</td>
</tr>
<tr>
<td>( n_s )</td>
<td>Scalar spectral index ( D_s \log \mathcal{P}_R + 1 )</td>
<td>–</td>
</tr>
<tr>
<td>( n_t )</td>
<td>Tensor spectral index ( D_s \log \mathcal{P}_h )</td>
<td>–</td>
</tr>
<tr>
<td>( n_{\text{iso}} )</td>
<td>Isocurvature spectral index ( D_s \log \mathcal{P}_S )</td>
<td>–</td>
</tr>
<tr>
<td>( n_{\text{ent}} )</td>
<td>Entropy spectral index ( D_s \log \mathcal{P}_{\text{ent}} )</td>
<td>–</td>
</tr>
<tr>
<td>( n_{\text{P, nad}} )</td>
<td>Non-adiab. pressure spectral index ( D_s \log \mathcal{P}_{\delta P,\text{nad}} )</td>
<td>–</td>
</tr>
<tr>
<td>( \alpha_s )</td>
<td>Scalar running ( D_s^2 \log \mathcal{P}_R )</td>
<td>–</td>
</tr>
<tr>
<td>( r )</td>
<td>Tensor-to-scalar ampl. ( \mathcal{P}_h(k_s)/\mathcal{P}_R(k_s) )</td>
<td>–</td>
</tr>
<tr>
<td>( \Theta )</td>
<td>Bundle width ( \mathcal{P}_h(k_s)/\mathcal{P}_R(k_s) )</td>
<td>Eq. (3.58)</td>
</tr>
<tr>
<td>( \cos \Delta )</td>
<td>( \omega )-s correlation angle ( \mathcal{P}_h(k) )</td>
<td>Eq. (3.27)</td>
</tr>
</tbody>
</table>

**Table 1.** Typical observables at the pivot scale \( k_* \). The derivative \( D_s \equiv d/d \log k \) is evaluated at \( k = k_* \). **MULTI_MODE_CODE** can also generate the full power spectra as a function of scale \( \mathcal{P}(k) \).

**Initial conditions dependence:** Multifield inflation models have an infinite number of possible inflationary solutions each of which can, in principle yield a different perturbation spectrum. Consequently, the observable spectra for multifield models can depend on their initial conditions in ways that have no direct analogue in slow-roll, single-field models, which have only one possible trajectory in field-space.

**Inherently stochastic predictions:** Even if the potential \( V \) is completely fixed, multifield models will give an inherent spread of predictions due to the allowed variance in the fields’ initial conditions. In general, multifield models will predict a variety of spectra, unless the stochasticity in the initial conditions can be controlled \textit{a priori}.

3.2 Classical background

Consider $N_f$ scalar fields $\phi_I$ with the matter sector of the action given by

$$S = \int d^4x \sqrt{-g} \left[ -\frac{1}{2} \partial_\mu \phi_I \partial^\mu \phi_I^J - V(\phi_I) \right], \quad (3.1)$$

where we use the Einstein summation convention over repeated indices. Greek indices describe spacetime, going from 0, $\ldots$, 3, upper-case Latin indices describe the number of fields, going from 1, $\ldots$, $N_f$, and lower-case Latin indices describe space, going from 1, $\ldots$, 3. The field space indices are raised using the Kronecker delta $\delta^{IJ}$. The determinant of the spatial metric $g_{\mu\nu}$ is $g$. In this paper we only consider inflation models with minimal coupling to Einstein gravity and a matter sector described by scalar fields. The current incarnation of MultiModeCode only solves models with canonical kinetic terms, but we give the equations of motion for models with a non-trivial field-space metric in Appendix A. Implementing these general field-space metrics is straightforward since MultiModeCode has been written modularly, but is left for future work.

First-order, non-interacting perturbation theory separates the homogeneous, classical background from the spatially-dependent modes as $\phi_I(t, \vec{x}) \to \phi_I(t) + \delta \phi_I(t, \vec{x})$, where we assume that these two components can be treated independently. The homogeneous background fields obey the Klein–Gordon equations

$$\ddot{\phi}_I + 3H \dot{\phi}_I + \frac{\partial V}{\partial \phi_I} = 0, \quad (3.2)$$

where an overdot indicates a derivative with respect to cosmic time $t$ and we use $M_{01}^2 = (8\pi G)^{-1} = 1$ throughout this paper. The 0-0 Einstein field equation gives the Friedmann equation

$$3H^2 = \frac{1}{2} \phi_I \phi_I^J + V(\phi_I), \quad (3.3)$$

which can be differentiated with respect to $t$ to yield

$$2\dot{H} = -\dot{\phi}_0^2. \quad (3.4)$$

In Eq. (3.4) we have used the inflaton trajectory velocity, $\dot{\phi}_0^2 \equiv \dot{\phi}_I \dot{\phi}_I^J$. We can regard the composite field $\phi_0$ as the clock of multifield inflation. It is the classical field defined along the inflaton trajectory, and represents the length of the classical field-space path.

In practice, if the dynamics are inflationary, it is numerically convenient to evolve the equation with the number of e-folds $N_e \equiv \ln a(t)$ as the independent variable, giving

$$\frac{d^2 \phi_I}{dN_e^2} + (3 - \epsilon) \frac{d \phi_I}{dN_e} + \frac{1}{H^2} \frac{\partial V}{\partial \phi_I} = 0, \quad (3.5)$$

where we have defined the first slow-roll parameter as

$$\epsilon \equiv -\frac{\dot{H}}{H^2} = \frac{1}{2} \frac{d \phi_I}{dN_e} \frac{d \phi_I}{dN_e}. \quad (3.6)$$

The Friedmann equation (3.3) can then also be expressed as

$$H^2 = \frac{V}{3 - \epsilon}. \quad (3.7)$$
If $V \approx 0$, Eq. (3.7) requires $\epsilon \approx 3$, which will result in numerical instability whenever we try to set initial conditions that are dominated by their kinetic energy. We side-step this issue by using the cosmic time Eq. (3.2) and $H$ as defined in Eq. (3.3).

Solving Eq. (3.5) therefore only requires the initial conditions $\phi_I$ and $d\phi_I/dN_e$, because the dependence on the scale factor $a$ is explicitly removed by the 0-0 Einstein equation (3.7) as a result of assuming a flat FLRW spacetime. As mentioned in §3.1, the perturbation spectrum depends on these initial conditions, which are specified as a prior probability distribution $P(\phi_I, \phi_I')$.

### 3.3 Mode equations

To obtain the first-order equation of motion for the perturbations $\delta \phi_I$, we need to expand the action (3.1) to second-order and include the first-order scalar perturbations to the flat FLRW metric, given by

$$\text{d}s^2 = -(1 + 2 \Phi) \text{d}t^2 - 2 a^2 H \text{d}t \text{d}x^i + a^2 \left[(1 - 2 \Psi) \delta_{ij} - 2 \partial_i \partial_j E \right] \text{d}x^i \text{d}x^j,$$

where

$$\partial_i \partial_j E \equiv \partial_i \partial_j E - \frac{1}{3} \delta_{ij} \nabla^2 E$$

is trace-free. We choose the spatially-flat gauge, so that $\Psi = E = 0$, and vary the expanded action $\delta S_{\phi}$ with respect to the perturbations $\delta \phi_I(t, \vec{x})$ to get the first-order equation of motion for the free-field perturbations. After Fourier-transforming the scalar perturbations to $\delta \phi_I(k)$, the mode equations in this gauge are

$$\frac{d^2 \delta \phi_I}{dN_e^2} + (3 - \epsilon) \frac{d \delta \phi_I}{dN_e} + \frac{k^2}{a^2 H^2} \delta \phi_I + C_{IJ} \delta \phi_J = 0,$$

where

$$C_{IJ} \equiv \frac{\partial_I \partial_J V}{H^2} + \frac{1}{H^2} \left( \frac{\partial_I \partial_J V}{dN_e} \right) + (3 - \epsilon) \frac{d \phi_I}{dN_e} \frac{d \phi_J}{dN_e}$$

and $\partial_I \equiv \partial_i \partial \phi_I$. The equation of motion for the tensor metric perturbations can be derived similarly; since the non-gauge degrees of freedom are massless and only minimally coupled to the matter sector, the resulting equations of motion are identical to the case of single-field inflation.

To solve the perturbation equations, it is usually convenient to work with the Mukhanov–Sasaki variable $u_I \equiv a \delta \phi_I$. The mode equation for $u_I$ is

$$\frac{d^2 u_I}{dN_e^2} + (1 - \epsilon) \frac{d u_I}{dN_e} + \left( \frac{k^2}{a^2 H^2} - 2 + \epsilon \right) u_I + C_{IJ} u_J = 0$$

with $C_{IJ}$ as in Eq. (3.11). Since the mass matrix, defined as $m^2_{IJ} \equiv \partial_I \partial_J V$, is not necessarily diagonal, the perturbation equations (3.12) mix the annihilation operators for all of the fields [38]. We therefore need to expand each perturbation mode $u_I(k)$ and $u^*_I(k)$ using $N_f$ harmonic oscillators $a_I(k)$:

$$u_I(k, N_e) = \psi_I^{J*}(k, N_e) a_J(k) \quad \text{and} \quad u^*_I(k, N_e) = \psi_I^{J*}(k, N_e) a^*_J(k),$$

where

$$\psi_I^{J*}(k, N_e) \equiv \frac{1}{N_f^{1/2}} \int \text{d}k' \psi_I^{J*}(k', N_e) a_J(k')$$
where (†) and (∗) represent Hermitian and complex conjugation, respectively. We can then define canonical commutation relations \([a_I(k), a_J^†(k')] = (2\pi)^3 \delta_{IJ} \delta^{(3)}(k - k')\). The mode matrix \(\psi_{IJ}\) evolves according to
\[
\frac{d^2 \psi_{IJ}}{dN_e^2} + (1 - \epsilon) \frac{d \psi_{IJ}}{dN_e} + \left(\frac{k^2}{a^2 H^2} - 2 + \epsilon\right) \psi_{IJ} + C_{IL} \psi_{LJ} = 0.
\] (3.14)

Finding the perturbation spectrum requires setting initial conditions in Eq. (3.14) and using the background equations (3.5) to find the time \(N_e\), when the mode \(k\) leaves the horizon, which also depends on the moment at which the pivot scale \(k_*\) leaves the horizon, \(N_*\) e-folds before the end of inflation.

The usual initial condition is the Bunch-Davies state [43], which assumes the field basis has been chosen such that the \(\psi_{IJ}\) are originally diagonal and sets the initial condition for Eq. (3.14) as if the mode matrix were freely oscillating in Minkowski space. This is well-motivated, since for modes deep in the horizon \(k \gg aH\), the mode matrix \(\psi_{IJ}\) obeys the free wave equation in conformal time
\[
\frac{d^2 \psi_{IJ}}{d\tau^2} + k^2 \psi_{IJ} = 0,
\] (3.15)

where \(d\tau \equiv a dt\). If we assume that the mode matrix is initially diagonal at \(\tau = -\infty\), then Eq. (3.15) yields two solutions
\[
\psi_{IJ} = \frac{1}{\sqrt{2k}} \left( C_1 e^{ikt} + C_2 e^{-ikt} \right) \delta_{IJ}.
\] (3.16)

Translating to e-fold time, the initial conditions can be set by
\[
\psi_{IJ}\bigg|_{N_e=0} = \frac{1}{\sqrt{2k}} \left( C_1 + C_2 \right) \delta_{IJ} \quad \text{and} \quad \frac{d\psi_{IJ}}{dN_e}\bigg|_{N_e=0} = \frac{i}{aH} \sqrt{\frac{k}{2}} \left( C_1 - C_2 \right) \delta_{IJ}.
\] (3.17)

The Bunch-Davies initial condition is equivalent to choosing \(C_1 = 0\) and \(C_2 = 1\). While only the Bunch-Davies initial condition is implemented in MultiModeCode, non–Bunch-Davies modes could be easily accommodated.\(^4\)

Although the \(u_I's\) are convenient for short wavelength modes, they grow exponentially after the modes exit the horizon. So once the mode is outside the horizon, MultiModeCode switches from \(u_I\) to \(\delta\phi_I\) by matching boundary conditions at a time \(N_e^*\) just after horizon crossing with
\[
u_I\bigg|_{N_e^*} = e^{N_e^* \delta\phi_I}\bigg|_{N_e^*} \quad \text{and} \quad \frac{d\nu_I}{dN_e}\bigg|_{N_e^*} = e^{N_e^*} \left( \delta\phi_I + \frac{d\delta\phi_I}{dN_e} \right)\bigg|_{N_e^*}.
\] (3.18)

### 3.4 Power spectra

Unlike single-field inflation, the multifield power spectrum involves contractions of the mode matrix. Using the canonical commutation relations above, the two-point VEV of the field perturbations yields the power spectrum
\[
P_{\delta\phi}^{IJ}(k) = \frac{k^3}{2\pi^2} \left[ \frac{1}{a^2} \right] \psi_{LJ}^I \psi_{LJ}^{*I}.
\] (3.19)

\(^3\)An alternative approach is to simply bypass this issue by solving for the field correlation functions directly rather than the individual modes, as in the transport method [34, 40–42].

\(^4\)One would do this by changing the modes’ initial conditions in the set_background_and_mode_ic() subroutine in modpk.f90.
When the field trajectories are not turning, on super-horizon scales the fields $\phi_I$ and their
momenta $\pi_I$ commute, indicating that they have transitioned to a regime where Eq. (3.19)
can be interpreted as an expectation value over realizations of classical, random fields.

To relate this field-space power spectrum to gauge-invariant perturbation variables [44–
46], we first define the curvature perturbation on comoving hypersurfaces $R$ by

$$R \equiv \Psi + \frac{1}{3} \nabla^2 E + aH (B + v),$$  \hspace{1cm} (3.20)

where $v$ is given in terms of the momentum density of the stress-energy tensor $T_{\mu \nu}$ as

$$T^i_0 \equiv (\bar{\rho} + \bar{P}) \delta^{ij} \frac{\partial v}{\partial x^j},$$  \hspace{1cm} (3.21)

where $\bar{\rho}$ and $\bar{P}$ are the background energy and pressure densities, respectively. If we evaluate
Eq. (3.20) on spatially-flat hypersurfaces during inflation, $R$ reduces to

$$R = -\frac{H}{\phi_0} \omega_I \delta \phi^I,$$  \hspace{1cm} (3.22)

where $\omega_I \equiv \dot{\phi}_I / \dot{\phi}_0$ is a basis vector that projects $\delta \phi_I$ along the direction of the classical
background trajectory, given by the solutions to Eq. (3.5). The vector $\vec{\omega}$ and a complementary
set of $(N_f - 1)$ mutually orthonormal basis vectors $\vec{s}_K$ form the kinematic basis [47, 48], where the separation between the adiabatic perturbations in Eq. (3.22) and transverse, isocurvature
perturbations is made explicit. Since $\vec{\omega}$ depends on the nonlinear background evolution, in
MULTIMODECODE we find the $\vec{s}_K$ numerically by Gram–Schmidt orthogonalization.

The adiabatic curvature power spectrum $P_R$ is then the projection of $P_{\delta \phi}^{IJ}$ along the field
vector $\omega_I$, scaled by the pre-factor in Eq. (3.22), leaving

$$P_R(k) = \frac{1}{2\epsilon} \omega_I \omega_J P_{\delta \phi}^{IJ}(k).$$  \hspace{1cm} (3.23)

The gauge-invariant scalar density spectrum in Eq. (3.23) is the final result for the adiabatic
two-point function to first-order in perturbation theory.

Since Eqs. (3.22) and (3.23) are projected along $\vec{\omega}$, a simple definition for the isocurvature
perturbations $S_K$ is the orthogonal projection along the $\vec{s}_K$ directions

$$S_K \equiv -\frac{H}{\phi_0} s_K^I \delta \phi^J.$$  \hspace{1cm} (3.24)

By projecting $P_{\delta \phi}^{IJ}$ onto all the directions $s_K$ that are orthogonal to $\omega_I$ and scaling the result
as in Eq. (3.23), leads to the isocurvature power spectrum:

$$P_S(k) = \frac{1}{2\epsilon} \sum_{KL} \sum_{IJ} s_K^I s_J^L P_{\delta \phi}^{IJ}(k),$$  \hspace{1cm} (3.25)

where we have left the summations explicit to indicate that the isocurvature basis vectors are
$(N_f - 1)$–dimensional. We include this definition of isocurvature because it is numerically
stable, as we discuss in Sect. 5.1.
Similarly, we define the adiabatic-isocurvature cross-spectra $P_{RS}$, which is the cross-correlation between the comoving curvature perturbation and the total isocurvature perturbation, given by the contraction of $P_{I}^{I}$ with both $\omega$ and the isocurvature basis vectors $s_{K}$

$$P_{RS}(k) = \frac{1}{2\epsilon} \sum_{N_{f}}^{N_{f}-1} \sum_{IJ} \omega_{I} s_{K}^{I} (P_{\delta\phi}^{I} + P_{\delta\phi}^{IJ}).$$  

(3.26)

Cross-correlations are generically expected if the background trajectory is curved as modes of interest leave the horizon. By parametrizing Eq. (3.26) with the scalar value

$$\cos \Delta \equiv \frac{P_{RS} \sqrt{P_{R} P_{S}}}{},$$  

(3.27)

it was shown in Ref. [49] that, for the case of $N_{f} = 2$, the value of $r$ is suppressed relative to the single-field, slow-roll expectation by $r \approx 16 \epsilon \sin^{2} \Delta$, to first-order in slow-roll. In principle, $\Delta$ may be detectable from CMB observations [50, 51].

However, by differentiating Eq. (3.20) with respect to time $t$, the comoving curvature perturbation will not necessarily be constant even for $k \ll aH$. Instead,

$$\dot{R} = -\frac{H \dot{\phi}}{\phi_{0}} \delta P_{\text{nad}},$$  

(3.28)

where $\delta P_{\text{nad}}$ is the non-adiabatic pressure perturbation [52–54]. This quantity is the difference between the total pressure perturbation

$$\delta P = \dot{\phi}_{I} \delta \phi^{I} - \dot{\phi}_{I} \dot{\phi}^{I} \Phi - V_{,I} \delta \phi^{I},$$  

(3.29)

and the adiabatic pressure perturbation

$$\delta P_{\text{ad}} = c_{s}^{2} \delta \rho,$$

where the speed of sound is $c_{s}^{2} = \frac{\dot{P}}{\rho}$ and the lapse function is

$$\Phi = \frac{1}{2H} \dot{\phi}_{I} \delta \phi^{I},$$  

(3.30)

defined in the spatially-flat gauge [55]. Given the total density perturbation

$$\delta \rho = \dot{\phi}_{I} \delta \phi^{I} - \dot{\phi}_{I} \dot{\phi}^{I} \Phi + V_{,I} \delta \phi^{I},$$  

(3.31)

the non-adiabatic pressure power spectrum $P_{\delta P_{\text{nad}}}$ reduces to

$$P_{\delta P_{\text{nad}}}(k) = \frac{k^{3}}{2\pi^{2} a^{2}} \left[ A_{I} A_{J} \psi_{L}^{I} \psi_{L}^{J} + A_{I} B_{J} \psi_{L}^{I} \psi_{L}^{J} + B_{I} A_{J} \psi_{L}^{J} \psi_{L}^{I} + B_{I} B_{J} \psi_{L}^{I} \psi_{L}^{J} \right],$$  

(3.32)

where $(l)$ indicates a derivative with respect to $e$-foldings $N_{e}$ and we have defined the vectors

$$A_{I} = \frac{1}{3aH^{2} \epsilon^{I} \phi_{L}^{I} \left[-3H^{2} \phi_{L}^{I} - \partial_{L}V\right] \partial_{I}V + H^{2} \partial_{M}V \phi_{L}^{I} \left(\delta_{LI} + \frac{1}{2} \phi_{L}^{I} \phi_{L}^{I}\right) \right]$$  

and

$$B_{I} = \left(1 - c_{s}^{2}\right) H^{2} \phi_{L}^{I}.$$

(3.33)

(3.34)

By analogy to Eq. (3.24), we can build an entropy perturbation from the non-adiabatic pressure perturbation [26, 47, 56], with

$$\delta S = \frac{H}{P} \delta P_{\text{nad}}.$$  

(3.35)
From this we obtain our final definition of isocurvature, the *comoving entropy spectrum*, given by

\[ P_{\text{ent}}(k) = \left( \frac{H}{F} \right)^2 P_{\delta P, \text{nad}}. \]  

### 3.5 \( \delta N \) formalism

The separate-universe assumption [53, 57–61], often referred to as \( \delta N \), states that when smoothed on some physical scale much larger than the horizon, the evolution of each smoothed patch can be computed using only background quantities. By identifying that \( \zeta = \delta N \), where \( \zeta \) is the curvature perturbation on constant density hypersurfaces and \( \delta N \) measures the variation in the number of e-folds between an initial flat hypersurface and a subsequent constant density hypersurface, Lyth and Rodriguez demonstrated that this assumption can be taken advantage of when computing correlation functions by performing a Taylor expansion in terms of the initial conditions [62].

\[ \zeta = N_I \delta \phi^I + \frac{1}{2} N_{IJ} \delta \phi^I \delta \phi^J + \ldots. \]  

(3.37)

The main difficulty in this approach lies in computing the derivatives of the number of e-folds (\( N_I = \partial N_e / \partial \phi^I \), \( N_{IJ} \) etc.). However for sum-separable models these expressions can be computed analytically [63, 64]. For models with fields much lighter than \( H \) at horizon crossing, the numerically intensive calculation of solving for the modes may therefore be unnecessary. **MultiModeCode** implements this \( \delta N \) slow-roll formalism where we assume that \( t^* \) is the moment when the pivot-scale \( k^* \) leaves the horizon and that the field perturbations at this time are uncorrelated, with a power spectrum

\[ P_{\delta \phi}^{IJ} = \left( \frac{H}{2\pi} \right)^2 \delta^{IJ}. \]  

(3.38)

We also assume that the tensor modes, which are massless and uncoupled to the matter sector, have a power spectrum \( P_h = 8 (H/2\pi)^2 \). At least to first order, on super-horizon scales \( \zeta = \mathcal{R} \) [65], which allows us to compare the predicted power spectrum for \( \zeta \) using the \( \delta N \) formalism to the adiabatic power spectrum in Eq. (3.23).

If the potential \( V \) is sum-separable so that

\[ V = \sum_I V_I(\phi_I), \]  

(3.39)

then we can use the Klein–Gordon equations (3.5) for the scalar fields to obtain a sum-separable expression for the amount of expansion between the two surfaces

\[ N_e = -\sum_I \int_{t^*}^c \frac{V_I}{V_I'} d\phi_I, \]  

(3.40)

where \( V_I' \equiv dV_I / d\phi_I \). If \( V \) were not sum-separable, the derivatives of \( N_e \) would in general have to be obtained numerically by evolving the background equations of motion (3.5) on a stencil in field-space and taking the finite difference. We have not implemented this feature in **MultiModeCode** as it is more computationally intensive than solving the mode equations.
When the potential is sum-separable, the derivatives of $N_e$ can be simplified into the expressions \cite{63, 64}

$$N_{I} = \frac{1}{\sqrt{2\epsilon_I}} \frac{V_I^* + Z_I^j}{V^*},$$  

and

$$N_{IJ} = \delta_{IJ} \left[ 1 - \frac{\eta_I}{2\epsilon_I} \left( \frac{V_I^* + Z_I^j}{V^*} \right) \right] + \frac{1}{\sqrt{2\epsilon_I} V^*} \frac{\partial Z_I^j}{\partial \phi_I^*},$$

where

$$Z_I^j = V^c_I \epsilon_I^c - V_I^c,$$

$$Z_{IJ}^j = -\frac{V^2}{V^*} \sqrt{\frac{2}{\epsilon_j}} \sum_{K=1}^{N_I} \epsilon_K \left( \frac{\epsilon_I}{\epsilon} - \delta_{IK} \right) \frac{\left( \epsilon_j - \delta_{JK} \right) \left( 1 - \frac{\eta_K}{\epsilon} \right)}{\epsilon},$$

and the slow-roll parameters are

$$\epsilon \equiv \sum_I \epsilon_I = \frac{1}{2} \sum_I \frac{V_I^2}{V^2},$$

and

$$\eta \equiv \sum_I \eta_I = \sum_I \frac{V''_I}{V'}.$$

The contribution from the EOI surface is therefore completely encoded in the functions $Z_I$ and its derivatives.

The relationship \eqref{3.37} and the expansion equation \eqref{3.40} allow us to define pivot-scale observables for the scalar perturbations $\zeta$. We will focus on the observables obtainable only through the first and second derivatives of $N_e$, and express our results only to the lowest order in slow-roll. We start with the $\zeta$ power spectrum

$$\mathcal{P}_\zeta = N_{I} N_{I}^* \left( \frac{H}{2\pi} \right)^2,$$

and the tensor-to-scalar ratio

$$r = \frac{8}{N_{I} N_{I}^*},$$

which have simple expressions only in terms of $N_{I}$. The adiabatic and tensor spectral indices $n_s$ and $n_t$ also have easily evaluated expressions

$$n_s - 1 = -2\epsilon_s - \frac{2}{N_{I} N_{I}^*} + \left( \frac{2}{V} \right) \frac{V_{IJ} N_{I}^* N_{J}^*}{N_{K} N_{K}^*}$$

and

$$n_t = \frac{-2\epsilon_s}{1 - \epsilon_s}.$$

The expression for the scalar running $\alpha_s$ is more complicated, but straightforward to compute \cite[e.g., Eq. 6.14 in Ref. \cite{66}].
To obtain the amplitude of the predicted non-Gaussianity we further assume that the field perturbations at horizon crossing are purely Gaussian, since the non-Gaussianity generated by sub-horizon evolution of the modes is typically slow-roll suppressed \cite{63, 67}, assuming that slow-roll is not violated. Following Refs. \cite{63, 68}, we use the non-linearity parameter

\[ \frac{-6}{5} f_{\text{NL}} \equiv \left( \prod_i k_i^3 \right) \frac{B_\zeta}{4 \pi^4 P_\zeta^2} \approx \frac{N_{I,J,K} N_{I,J} N_{K}}{(N_{I,K} N_{J,I})^2}, \]  

(3.51)

where \( B_\zeta \) is the bispectrum. Given Gaussian field perturbations at horizon crossing, the trispectrum amplitude is then parametrized by the non-linearity parameters \cite{69, 70}

\[ \tau_{\text{NL}} = \frac{N_{I,J,K} N_{J,I} N_{K}}{(N_{I,J,K} N_{I,J,K})^3} \]  

(3.52)

and

\[ g_{\text{NL}} = \left( \frac{25}{54} \right) \frac{N_{I,J,K} N_{J,I} N_{J,K}}{(N_{I,J,K} N_{I,J,K})^3}. \]  

(3.53)

Since \( g_{\text{NL}} \sim N_{I,J,K} \) we do not compute it here, although it could be implemented by taking the third derivative of \( N_e \) as in Ref. \cite{64}.

### 3.6 Bundle width

An alternative method of monitoring isocurvature is to acknowledge that under slow-roll, the separate universe assumption is precisely analogous to geometrical optics in field space \cite{34}. The smoothed spatial patches described in Sect. 3.5 each correspond to a distinct non-interacting trajectory in field space with perturbed initial conditions with respect to some arbitrary fiducial trajectory. These perturbed trajectories can then be thought of as forming a bundle moving through a medium with refractive index \( \sqrt{2} \epsilon \). One can therefore track isocurvature evolution using only background quantities, by associating isocurvature growth and decay with dilation and contraction of the bundle. While the precise analogy with geometrical optics does not remain when slow-roll is violated, one still has a useful set of geometrical quantities for understanding the evolution of field perturbations.

Under slow-roll, the Klein–Gordon equation reduces to

\[ \frac{d\phi_I}{dN_e} = -\partial_I \log V, \]  

(3.54)

which is Huygen’s equation and an infinitesimal vector propagated along the beam is called a Jacobi field. If we take \( \delta \phi_I \) to be such a field, we can obtain from Eq. (3.54) how it will propagate:

\[ \frac{d\delta \phi_I}{dN_e} = -[\partial_I \partial_J \log V] \delta \phi_J, \]  

(3.55)

which is the slow-roll analogue of Eq. (3.12) \cite{34, 42}. Indeed, we could have recast the whole of Sect. 3.3 in this language \cite{34}. The term in square brackets is usually referred to as the expansion tensor and it encodes all information required for tracking field perturbations.\footnote{This point is heavily emphasised in the context of the transport method of computing inflationary correlation functions \cite{34, 40–42, 71}.}
under slow-roll it has a particularly simple geometric interpretation. We can decompose the
expansion tensor as
\[ \partial_I \partial_J \log V = \frac{\theta}{N} + \sigma_{IJ} + \omega_{IJ}, \] (3.56)
where \( \sigma_{IJ} \) is the symmetric shear, \( \omega_{IJ} \) is the antisymmetric twist, and the key quantity for
our purposes is the dilation, given by the trace
\[ \theta = \text{Tr} \partial_I \partial_J \log V. \] (3.57)
If \( \theta > 0 \), then isocurvature is growing and if \( \theta < 0 \), then isocurvature is decaying. We can then
find a measure \( \Theta \) of the bundle width by integrating this along the inflationary trajectory
\[ \Theta \equiv \exp \left[ \int_{N_0}^{N} \theta(N') dN' \right], \] (3.58)
which normalizes the bundle width so that \( \Theta(N_0) \equiv 1 \). In situations where we only want to
solve the background equations of motion, the bundle width is informative for understanding
whether or not \( \zeta \) becomes conserved on superhorizon scales, which is a crucial requirement
when comparing the predictions of a model with observation. For two fields \( \Theta \to 0 \) is a
necessary and sufficient condition for the approach to an adiabatic limit. However when
there are more fields the situation is more complicated, e.g., the bundle may contract to a
sheet rather than a caustic. We refer the reader to Ref. [34, 66, 71] for more details.

4 The method

We outline the procedure used to obtain the power spectrum predictions, with the algo-
rithmic structure of MULTIMODECODE in Algorithm 1. While this largely follows previous
implementations, such as PYFLATION [25–28], we give the method the sake of clarity and
reproducibility.

We start by defining the functional form of the potential \( V \) and prior probability distri-
bution functions (PDFs) for the parameters that define \( V \), which we call Lagrangian param-
ters or model parameters, and the background initial conditions \( \phi_{I,0} \) and \( \phi'_{I,0} \). We treat the
simple situation of exactly specifying a set of Lagrangian parameters and initial conditions
as a special case, where the prior probability for these parameters is trivial. Given these
priors, the program will build a numerical sample by iteration until a pre-defined number of
samples is reached.

MULTIMODECODE first solves the background equations of motion (3.5) until the end-
of-inflation. While we have included the natural condition of \( \epsilon = 1 \) as the default ending
criterion for inflation, there is complete functionality to end inflation by another method, in
particular a waterfall transition via the hybrid mechanism [72, 73] at some reference phase-
space point.

Given a value for the number of \( e \)-folds \( N_\star \) between when the pivot scale \( k_\star \) leaves the
horizon and the end-of-inflation, which is either fixed by the user or set in each iteration
of the code through the sampling of a prior probability \( P(N_\star) \), we obtain the value of \( H \)
at horizon crossing by interpolating the numerical background solution. The pivot scale \( k_\star \)
must be pre-defined by the user and defaults to 0.002 Mpc\(^{-1}\). From this, we normalize the
size of the universe so that \( k_\star = aH_\star \) at \( N_\epsilon = N_{\text{tot}} - N_\star \).
Algorithm 1 MultiModeCode method

```plaintext
define sample size, \( V \), \( k^* \)
for all elements in sample do
  procedure Background Solver:
    get Lagrangian parameters for \( V \) and ICs for Eq. (3.5) from prior PDF
    with the end-of-inflation (EOI) criterion set by user, solve Eq. (3.5) until EOI
    check inflation \((\ddot{a} > 0)\) started and ended
  procedure Scale-factor Normalizer:
    get \( N_* \) from user or by prior PDF
    check total inflationary e-folds \( N_{\text{tot}} \geq N_* \)
    define \( a \) such that \( k^* = a\H_0 \) at \( N_e = N_{\text{tot}} - N_* \) before inflation ends
  procedure \( \delta N \) Calculator:
    if \( V \) is sum-separable, then calculate \( \delta N \) observables near \( k^* \)
  for all modes \( k \) do
    procedure Mode Initializer:
      define initial time \( N_{e,0} \) with \( k \gg a_0H_0 \)
      while the corrections to Eq. (3.15) are above some tolerance:
        set earlier \( N_{e,0} \) and check \( N_{e,0} > 0 \)
        set Bunch-Davies IC for mode matrix \( \psi_{IJ}(k) \) at \( N_{e,0} \)
    procedure Mode Solver:
      solve Eq. (3.12) until \( k \approx aH \)
      change variable as in Eq. (3.18) and solve until EOI
      calculate power spectra for \( k \)
  procedure \( k^* \)-observable Calculator:
    calculate amplitudes, spectral indices, etc. at \( k^* \) by finite difference in \( k \)-space

For each scale of interest \( k \), we set the modes’ initial conditions at a time \( N_{e,0} \) when that mode is significantly sub-horizon, \( k \gg a_0H_0 \). For the Bunch-Davies initial state, this point is chosen iteratively by making sure that the relative corrections to Eq. (3.15) that are sub-dominant for \( k \gg aH \) are smaller than a pre-defined tolerance. This tolerance is set to \( 1 \times 10^{-5} \); from observing the sub-horizon evolution of the modes, using a tolerance at least this tight gives no change to the value of the modes at horizon crossing.

We then solve the mode equations (3.12) for the variable \( \psi_{IJ} \) for the period of time when the modes are smaller than the causal horizon, \( k \gtrsim aH \), and then switch to a two-index matrix built from the \( u_I \) in Eq. (3.18) for super-horizon evolution. The power spectra are calculated for each \( k \) and various pivot-scale statistics are evaluated by finite-difference between a few scales \( k_i \) near \( k^* \). If the potential \( V \) is sum-separable, the program also calculates the \( \delta N \) values for the observables described in Section 3.5.

Numerous checks are performed on the background and mode equation evolution so that MultiModeCode will either fail gracefully if a fatal exception is raised or declare a particular initial parameter set invalid and iteratively generate a new set of parameters in order to explore cosmologically relevant parameter sets. We have extensively tested the numerical stability of the code and have included a number of easily controllable options allowing the user to control the numerical accuracy, as well as the type of ODE solver.
```
5 Numerical results

5.1 Isocurvature stability

Fig. 1 illustrates a problem that arises when computing the isocurvature spectra $P_{\delta P,\text{nad}}$ and $P_{\text{ent}}$. We have plotted the super-horizon evolution of the power spectra for the adiabatic and non-adiabatic pressure perturbations, as well as the adiabatic curvature, entropic, and isocurvature spectra, with $N_e = 55$, for a two-field inflation model with the potential

$$V = \frac{1}{2}m_1^2\phi_1^2 + \frac{1}{2}m_2^2\phi_2^2.$$  

(5.1)

To match the analysis performed in Refs. [26, 28, 74, 75] we choose $m_1^2 = 10^{-11.7}$, $m_2^2 = 10^{-10.0}$, and initial conditions $\phi_{1,0} = \phi_{2,0} = 12.0M_P$. In particular, Fig. 1 can be compared directly to Figs 1 and 3 in Ref. [26]. With this choice of parameters, the background trajectory evolves primarily along the direction of the heavier field $\phi_2$ for $N_e \lesssim 25$, then turns sharply toward the $\phi_1$ direction for the remainder of inflation. The effect of this turn on the super-horizon perturbations can be seen clearly in the power spectra in Fig. 1.

In general, the calculation of $P_{\delta P,\text{nad}}$ and $P_{\text{ent}}$ becomes dominated by numerical error as the isocurvature perturbations decay. From Fig. 1, regardless of the amplitude of the isocurvature modes, the adiabatic pressure perturbations $\delta P_{\text{ad}} = c_s^2\delta\rho$ do not exponentially decay between horizon exit and the end of inflation. For the example model (5.1), the power spectrum for $\delta P_{\text{ad}}$ is approximately constant after the turn at $N_e \approx 25$. However, the total pressure perturbation $\delta P$ is approximately equal to $\delta P_{\text{ad}}$ during this time and the difference between the two reduces exponentially as the isocurvature modes decay.
Since \( \delta P_{\text{nad}} \equiv \delta P - \delta P_{\text{ad}} \) and \( \delta P_{\text{ad}} \to \delta P \), the numerical accuracy for \( \delta P_{\text{nad}} \) is limited by the real precision of the computer, which results in a finite difference error in the numerical calculation of \( \delta P_{\text{nad}} \) and a loss of significance. Using standard double precision accuracy, the expected error in \( \delta P_{\text{nad}} \) should then be

\[
\Delta_{\text{err}} \delta P_{\text{nad}} \sim \mathcal{O}(10^{-15}) \quad \mathcal{P}_{\delta P} \sim \mathcal{O}(10^{-15}) \quad \mathcal{P}_{\delta P,\text{ad}},
\]

which is confirmed in Fig. 1. Without correcting for this dominant error term, the value of \( \mathcal{P}_{\delta P,\text{nad}} \) will oscillate arbitrarily between zero and the limit in Eq. (5.2), which is an upper bound on the amplitude of the non-adiabatic pressure perturbations. Since entropic perturbations are usually defined as

\[
S_{IJ} \equiv \zeta_I - \zeta_J,
\]

where \( \zeta_I \) is the curvature perturbation resulting from the \( I \)th fluid, this problem will arise naturally for all calculations of \( \mathcal{P}_{\text{ent}} \).

In contrast, the calculation of \( \mathcal{P}_S \) in Eq. (3.25) is directly proportional to the value of the decaying isocurvature modes in the kinematic basis. Using this isocurvature spectrum largely alleviates the numerical problems with \( \delta P_{\text{nad}} \), yielding a more faithful measure with a higher degree of accuracy. Figure 1 shows the exponential decay of \( \mathcal{P}_S \) after the super-horizon turn at \( N_e \sim 25 \). We compare this to \( \mathcal{P}_{\text{ent}} \), which becomes numerically unstable at \( N_e \approx 30 \), showing that the two measures \( \mathcal{P}_S \) and \( \mathcal{P}_{\text{ent}} \) are separated by 27 orders of magnitude at the end of inflation, despite being of the same magnitude at horizon crossing.\(^6\)

### 5.2 A case study: \( N_f \)-flation with a step

We have shown in Refs. [30, 31] that MultiModeCode is able to produce large volume Monte Carlo samples for \( N_f \)–monomial inflation with the potential

\[
V = \frac{1}{p} \sum_I \lambda_I |\phi_I|^p,
\]

for real exponents \( p \) [76–85]. In Ref. [30] we focused on the \( N_f \)–quadratic case with \( p = 2 \) and demonstrated that the predictions for the power spectrum do not sensitively depend on the prior probability chosen for the initial conditions of the fields. In Ref. [31] we further demonstrated this for the general case in Eq. (5.4), while focusing on the gravitational wave consistency relation. We were able to straightforwardly compare the analytical \( \delta N \) results to the numerics, greatly simplifying the procedure for comparing analytical results to the full numerical calculation. We include all of the IC priors used in these papers in MultiModeCode.

Since we have already demonstrated the power of MultiModeCode in Monte Carlo sampling, in this paper we will instead focus on a few case studies that are interesting

\(^6\)As the adiabatic limit is approached, \( \mathcal{P}_S \) can also receive a dominant contribution from roundoff error in the Gram-Schmidt orthogonalization procedure. If some components of the isocurvature vectors \( s_K \) are much smaller than others, this can result in a spurious projection of \( \mathcal{P}_S \) onto the isocurvature directions. In MultiModeCode we have included an optional subroutine \texttt{renormalize_remove_smallest in modpk\_potential.f90}, where the components of \( s_K \) are set to zero if they do not affect the normalization of \( s_K \), i.e., if the value of \( s_K \) is indistinguishable from roundoff error. In practice, we have never seen this problem arise, so this option needs to be uncommented in the source code before compilation.
due to their analytic intractability. We present results for a multifield generalization of the inflationary step-potential first used in Ref. [10]. This potential has the form

$$V = \frac{1}{2} \sum_I m_I^2 \phi_I^2 \left[ 1 + c_I \tanh \left( \frac{\phi_I - \bar{\Phi}_I}{d_I} \right) \right]$$

with masses $m_I$ and real constants $d_I$, $c_I$, and $\bar{\Phi}_I$ specifying the slope, amplitude, and position, respectively, for a step feature in the $I^{th}$ field. Phase transitions in sectors coupled only gravitationally to the inflaton sector may generate these hyperbolic-tangent features in $V$ and leave an observable imprint in the primordial density spectra if these symmetry breaking transitions occur during the last $O(60)$ e-folds of inflation [10, 32]. In the sharp-step limit, these features introduce oscillations as a function of $k$ into the adiabatic curvature power spectrum and a scale-dependent, oscillatory bispectrum [10, 33, 86, 87]. To keep $V > 0$ we require $c_I < 1$ and to satisfy the latest constraints on oscillations in the scalar power spectrum amplitude requires $c_I \lesssim 10^{-3}$, assuming that the step occurs as the scales relevant for the CMB leave the horizon [88–90].

With $c_I \to 0$, Eq. (5.5) is an uncoupled assisted inflation model [76, 91], first proposed in Ref. [81]. Models with a step feature are additionally interesting, because they can fit a wider range of data and have been well-studied in the single-field case. In particular, Ref. [33] contains an elegant analytical calculation for the single-field case of Eq. (5.5). However, replicating the same calculation for the general potential would be difficult — if not impossible — with the same techniques, since the possible existence of isocurvature perturbations significantly complicates the analysis. Consequently, a numerical exploration of this model is well-motivated.

Fixing the number of fields to $N_f = 10$, we set the initial conditions to $\phi_{I,0} = 10$, with the initial velocities set in slow-roll. The size and slope of the step are set to $c_I = 10^{-3}$ and $d_I = 10^{-2}$ respectively, and the masses $m_I$ relative to the fiducial mass to $\bar{m}^2 = 4.30 \times 10^{-11}$, which in the single-field limit yields $A_s$ at the best-fit value from the Planck TT data. Following Ref. [80], we choose the masses $m_I$ according to the Marčenko-Pastur distribution

$$P(m_I^2) = \frac{1}{2\pi m_I^2 \bar{m}^2 \beta} \sqrt{(\beta_+ - m_I^2)(m_I^2 - \beta_-)},$$

where

$$\beta_\pm = \bar{m}^2 \left( 1 \pm \sqrt{\beta} \right)^2$$

with $\beta = 1/2$. This distribution of masses is derived in Ref. [80], and has also been used in Refs. [30, 84, 92, 93].

We set the step positions $\bar{\Phi}_I$ for each field at the field-space point where the pivot scale $k_* = 0.002 \text{ Mpc}^{-1}$ leaves the horizon at $N_* = 55$ e-folds before the end of inflation in the no-step limit, $c_I \to 0$. Since the fields have identical initial conditions, the $\bar{\Phi}_I$ are functions only of the masses, so we plot the step positions versus the $m_I$ in Fig. 2. We also present the field-space trajectories according to Eq. (3.5) for the last 75 e-folds of inflation with these parameters. The heavier fields relax more quickly toward their minimum at $\phi_I = 0$ and the lighter fields have a larger value at horizon crossing. Since $c_I = 10^{-3}$, the step is not obviously visible at the level of the background trajectory without zooming in significantly.

However, Fig. 3 shows the substantial effect on the power spectra due to the steps. We see oscillatory behavior in the adiabatic, isocurvature, and entropic power spectra, but
almost no change in the tensor spectrum. Furthermore, we can see clearly that $P_S$ and $P_{\text{ent}}$ exhibit a nearly identical feature, simply scaled by a factor of roughly 65. These features in the isocurvature spectrum may lead to interesting effects during reheating or the subsequent evolution of the post-inflation universe.

6 Conclusion

We present the Fortran 95/2000 code **MultiModeCode**, designed to maximize computational efficiency when numerically exploring a broad range of multifield inflation models. The code also provides Monte Carlo sampling of prior probabilities for inflationary model parameters and initial conditions, enabling automated model exploration and the computation of probability distributions for observables. The mode equation method has a broad range of applicability, but the computational cost scales with the number of fields as $O(N_f^2)$. For models with sum-separable potentials, we have also implemented a slow-roll $\delta N$ calculation, which only requires solving the background equations of motion once in order to obtain the full power spectrum as well as higher order statistics. This drastically improves computation time, since the background equations of motion are only $O(N_f)$.

This code was used to explore the predictions of models with $O(100)$ fields in Refs. [30, 31]; here, we demonstrated its use with an $N_f$-flation model with a step. We find that a feature in the inflationary potential not only results in a feature in both the adiabatic power spectrum as a function of scale, $P_R(k)$, as well as the isocurvature spectra $P_S$, $P_{\text{ent}}$, and $P_{\delta P}$, with possible implications for the dynamics of many-field preheating scenarios. Further, we see numerical evidence that the isocurvature spectrum $P_S$ is a simple rescaling of the entropic spectrum $P_{\text{ent}}$, indicating that the projection of the mode power spectrum.
Figure 3. Features in the power spectra due to the step (5.5), which is positioned so that it affects the power spectra around the pivot scale $k_* = 0.002 \, \text{Mpc}^{-1}$ (gray). We compare (dashed, blue) the no-step case with $c_I = 0$, to (solid, green) the case with $c_I = 10^{-3}$. While there are oscillations in the adiabatic $P_R$, isocurvature $P_S$, and entropic $P_{\text{ent}}$ spectra, there is little variation in the tensor spectrum $P_h$.  

onto the isocurvature directions is related to a quantity that sources a change in $R$ on super-horizon scales.

MultiModeCode complements codes that currently exist to numerically compute the inflationary power spectra [10, 12, 19–21, 25–28, 71, 94]. We provide a basic usage manual for MultiModeCode in Appendix B to help users to adapt this program to their own problems. The theoretical basis of the method is outlined in Section 4. The ability of MultiModeCode to solve numerically challenging problems, such as the step-potential in §5.2, and to provide large samples of many-field inflationary models adds significantly to the early universe cosmologist’s toolkit for exploring and understanding realistic inflation models.

Acknowledgments

We thank Grigor Aslanyan, Adam Christopherson, Mafalda Dias, Ian Huston, Karim Malik, David Mulryne, David Seery, and Jonathan White for many helpful discussions throughout the duration of this work. We acknowledge the use of the dvode.f90.m.f90 numerical integration code,\(^7\) developed by G. Byrne, S. Thompson, and LLNL, which we redistribute here under the terms of their BSD-like license, and the CSV writing capabilities of the FLIBS

\[^7\]www.radford.edu/~thompson/vodef90web/
open-source library, developed by Arjen Markus. JF is supported by IKERBASQUE, the Basque Foundation for Science. HVP is supported by STFC and the European Research Council under the European Community’s Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement no 306478-CosmicDawn. The authors acknowledge the contribution of the NeSI high-performance computing facilities and the staff at the Centre for eResearch at the University of Auckland. New Zealand’s national facilities are provided by the New Zealand eScience Infrastructure (NeSI) and funded jointly by NeSI’s collaborator institutions and through the Ministry of Business, Innovation & Employment’s Research Infrastructure programme [http://www.nesi.org.nz]. This work has been facilitated by the Royal Society under their International Exchanges Scheme. This work was supported in part by National Science Foundation Grant No. PHYS-1066293 and the hospitality of the Aspen Center for Physics.

A Non-canonical kinetic terms

This appendix largely follows Ref. [59] and describes the background and first-order mode equations for inflation models with multiple scalar fields, but a general field space metric $G_{IJ}(\phi_K)$. These equations are not coded into MultiModeCode, but are an important reference, since multifield models with non-canonical kinetic energy terms fit the Planck data extremely well [95]. These have also been implemented into the single-field version of ModeCode by Ref. [96]. We recommend Ref. [59] for a thorough derivation of these quantities.

We start with the action for $N_f$ scalar fields $\phi^I$, given by

$$S = \int d^4x \sqrt{-g} \left[ -\frac{1}{2} G_{IJ} \partial^\mu \phi^I \partial_\mu \phi^J - V(\phi^I) \right]. \quad (A.1)$$

Again, we assume that Greek indices $\alpha \in [0, \ldots, 3]$ are for spacetime, upper-case Latin letters index the number of fields, $I \in [1, \ldots, N_f]$, and lower-case Latin letters index three-space, $i \in [1, \ldots, 3]$. The action (3.1) is a special case of Eq. (A.1), with $G_{IJ} = \delta_{IJ}$. To change the equations in §3 to reflect the general field-space metric, we follow the typical procedure of replacing partial derivatives with respect to the fields with covariant derivatives.

Varying Eq. (A.1) with respect to the fields gives the background equation of motion

$$\frac{D\phi^I}{dt} + 3 H \dot{\phi}^I + G^{IJ} V_{,J} = 0, \quad (A.2)$$

where we have assumed that the background fields are homogeneous, $\partial_i \phi_J = 0$. In Eq. (A.2) we have used the covariant differential [59]

$$D\phi^I = d\phi^I + \Gamma^I_{JK} \phi^J d\phi^K, \quad (A.3)$$

with the field-space Christoffel symbols

$$\Gamma^I_{JK} = \frac{1}{2} G^{IL} (G_{LJ,K} + G_{LK,J} - G_{JK,L}). \quad (A.4)$$

---

8http://flibs.sourceforge.net/
9We will update MultiModeCode to give this functionality in the near future and a simple implementation in Mathematica using the Transport method will be available soon [71].
which gives the covariant derivative \( V^I_I \equiv \partial V / \partial \phi^I \), because \( V \) is a scalar.

Following the treatment in Section 3, it is convenient to express the equations of motion for the field perturbations \( \delta \phi^k \) in spatially-flat gauge in terms of the generalized trajectory velocity \( \dot{\phi}_0 \), which can be extended from Eq. (3.4) to

\[
\dot{\phi}_0^2 \equiv G_{IJ} \dot{\phi}^I \dot{\phi}^J. \tag{A.5}
\]

From \( \dot{\phi}_0 \), we define the adiabatic unit vector in the kinematic basis \([47, 48]\) as \( \omega^I \equiv \dot{\phi}^I / \dot{\phi}_0 \), which projects vectors along the adiabatic direction in the generalized field-space. Finally, the mode equations in spatially-flat gauge read \([59]\)

\[
\frac{D^2}{dt^2} \delta \phi^k + 3H \frac{D}{dt} \delta \phi^k + \frac{k^2}{a^2} \delta \phi^k + C^I_{J} \delta \phi^I_{J} = 0, \tag{A.6}
\]

which is a simple generalization of Eq. (3.10), with the mass matrix

\[
C^I_{J} = G^{IK} V^K_{J L} - \dot{\phi}_0^2 R^I_{JKL} \omega^K \omega^L + 2e \frac{H}{\dot{\phi}_0} (\omega^I V^J_{,I} + \omega_J V^I) + 2(3 - e)H^2 \omega^I \omega_J, \tag{A.7}
\]

where \( R^I_{JKL} \) is the field-space Riemann tensor, built from \( G_{IJ} \). Again, we note that index contraction implies summation with respect to the field-space metric, \( X_I Y_J = G_{IJ} X^I Y^J \).

Quantizing the modes using the mode-matrix \( \delta \phi^I = \psi^I_{J} a_J \) proceeds as in Section 3.3 and the adiabatic power spectrum \( P_R \) is identical to Eq. (3.23), except contracting with respect to \( G_{IJ} \). The isocurvature directions should again be found by Gram-Schmidt orthogonalization, except now implemented in the curved field-space, and the spectra \( P_{\delta P, \text{nad}} \) and \( P_{\text{ent}} \) can be found as in Eqs. (3.33) and (3.36). The projected isocurvature spectrum \( P_S \) in Eq. (3.25) can instead be built by replacing the summation over the isocurvature directions by contraction with respect to the isocurvature directions of the field-space metric, \( \hat{G}_{IJ} \), after transforming to the kinematic basis.

### B MultiModeCode usage

**MultiModeCode** is publicly available at [www.modecode.org](http://www.modecode.org) and is released with a Modified BSD License. **MultiModeCode** has been implemented in a mix of Fortran 95/2000 and has been thoroughly checked on Mac OS and Linux systems with the freely-available GNU Fortran compiler (version 4.6.3+) and with Intel Fortran (version 14.0.2), with the Intel compiler yielding significant improvements in speed. There are no external dependencies necessary to use **MultiModeCode**.

We include a driver file `multimodecode_driver.f90` that contains the basic structure needed to explore the predictions of a model. The driver has many important routines for calculating the power spectrum and outputting the results. The file `parameters_multimodecode.txt` contains runtime parameters that are often changed between subsequent runs. The parameters are listed in Fortran namelists, so a change here does not require the user to recompile the whole code. There are a mix of basic and advanced parameters available and we describe them here.

The `&init` namelist holds important parameters related to initializing the program, the choice of inflationary potential \( V \), and program output:
\begin{verbatim}
&init
    num_inflaton = 10
    potential_choice = 1
    vparam_rows = 4
    slowroll_infl_end = .true.
    instreheat = .false.
/

The number of fields is set with the variable num_inflaton and an indentifying number is
chosen for the potential with potential_choice. The currently available potentials are listed
in the routine pot(phi) in modpk.potential.f90, including the multifield \( N_f \)-quadratic [76, 81, 97], \( N_f \)-monomial inflation with \( V \sim \lambda I | \phi_I | ^n \) [31], two-field hybrid inflation [72, 73],
a product of exponentials [76], and the multifield generalization of the hyperbolic-tangent
step potential in Ref. [10], which was used in Section 5.2. Adding a new potential is as easy
as providing the potential \( V \) and its derivatives in modpk.potential.f90 with a new value
for potential_choice. The array vparams contains information passed to the potential
function (e.g., masses and couplings) and has dimensions \((vparam_rows) \times (num_inflaton)\).
The values for vparams are set in parameters_multimode.txt in the namelist &params;
variables related to the pivot scale \( N_{pivot} \) and \( k_{pivot} \) are also set here.

The user can change the conditions for inflation to end by varying slowroll_infl_end,
which when set to .true. evolves the background fields until \( \epsilon = 1 \). If you do not want this
to be the ending criterion, then set slowroll_infl_end=.false. and adapt the subroutine
alternate_infl_end in modpk.odeint.f90 to change the ending condition. Furthermore,
with instreheat=.true. \( N^* \) becomes a derived parameter by requiring inflation to thermalize immediately after the end of inflation with \( w = 1/3 \), as in Ref. [20].

We use the &analytical namelist to set the options for calculating the power spectrum,
either using the \( \delta N \) calculations of §3.5 or evaluating the full mode equations of §3.3 or both.
\end{verbatim}

\begin{verbatim}
&analytical
    use_deltaN_SR = .true.
    use_horiz_cross_approx = .false.
    evaluate_modes = .true.
    get_runningofrunning = .false.
/

With use_deltaN_SR=.true. MULTI MODE CODE will calculate the \( \delta N \) observables at the
pivot scale (as given in namelist &params) and if use_horiz_cross_approx=.true., it will
ignore the contribution to \( N_{IJ} \) and \( N_{IJJ} \) from the end-of-inflation surface via the horizon
crossing approximation (HCA) [63, 83]. Setting evaluate_modes=.false. will make the
program only solve the background equations of motion, relying on the \( \delta N \) calculations to
obtain the spectra. If get_runningofrunning=.true., then the derivative of \( \alpha_s \) with respect
to \( \ln k \) will be calculated by a five-point stencil finite difference method, which requires two
additional calls to the code that solves the mode equations, significantly slowing down the
speed of the program.

The way by which the initial conditions are chosen for a given simulation depends on
the namelist &ic_sampling.nml namelist.
\end{verbatim}

\begin{verbatim}
&ic_sampling_nml
    ic_sampling = 1
\end{verbatim}
The variable `ic_sampling` controls the main behavior of the initial conditions’ prior probability and is set to an identifying number defined in the file `modpk_icsampling.f90` with the `ic_samp_flags` type. The currently available values are below:

```fortran
type :: ic_samp_flags
    integer :: reg_samp = 1
    integer :: eqen_samp = 2
    integer :: slowroll_samp = 3
    integer :: iso_N = 6
end type
```

Invoke each case by setting `ic_sampling` equal to the desired number. The functionality of each of these cases is:

- **reg_samp** (regular sampling): the simple case of setting a multifield initial condition and calculating the power spectrum. The initial conditions for the fields are set as the variable `phi_init0` in the `&params` namelist and the fields’ velocities are assumed to be initially in slow-roll.

- **eqen_samp** (equal-energy sampling): quasi–equal-area sampling of a phase-space surface with same initial energy, as in Refs [8, 9, 30]. Set the initial energy with the `energy_scale` variable in units of $M_{Pl}^2 = (8\pi G)^{-1}$ = 1. To also record and output the background field values as the background reaches $N_e = N_{iso \_ref}$ set `save_iso_N=.true.`. The prior ranges for the fields and velocities are set in the namelist `&priors`.

- **slowroll_samp** (slow-roll sampling): choose initial conditions in field space and set velocities by the slow-roll condition. The prior ranges for the fields are again chosen in `&priors`.

- **iso_N** (sampling e-fold surface): uniformly samples the surface $N_e = \sum \phi_i^2/2p$ for $N_f$-monomial inflation, as in Refs [29, 30]. Set the value for $N_e$ with $N_{iso \_ref}$ in the `&ic_sampling` namelist.

To implement a different initial conditions measure or sampling behavior, add a new identifier for the `ic_samp_flags` type. The initial conditions are set in the routine `get_ic` in `modpk_icsampling.f90` and you will need to implement your sampling technique here, following the examples in the code.

Similarly, the prior probabilities on the `vparams` array that defines the potential are set through the `&param_sampling_nml` namelist.

```fortran
&param_sampling_nml
  param_sampling = 1
  use_first_priorval = .true.
  vp_prior_min(1,:) = -14
  vp_prior_max(1,:) = -12
```
As with the initial conditions sampler, different behaviors are chosen by setting the variable `param_sampling` to a unique integer as specified in the `param_samp_flags` type in `modpk_icsampling.f90`.

```fortran
  type :: param_samp_flags
    integer :: reg_constant = 1
    integer :: unif_prior = 2
    integer :: log_prior = 3
  end type
```

Again, invoke each case by setting `param_sampling` to the desired number. To vary the number of e-folds after the pivot scale leaves the horizon, set `varying_N_pivot=.true.` and set the limits on the prior on $N_*$ in the `&priors` namelist, where we have assumed a uniform prior. Note that this is overridden if `instreheat=.true.`.

To use a different prior on the $vparams$ array, add a new integer to the `param_samp_flags` type and change the routine `get_vparams` in `modpk_icsampling.f90`. The default behavior is:

1. **reg_constant** (regular, constant parameters): the $vparams$ array is kept constant, as specified in the `$params$ namelist.

2. **unif_prior** (uniform prior probability): each column in $vparams$ is chosen with a uniform prior between $vp_{prior\_min}$ and $vp_{prior\_max}$. If `use_first_priorval=.true.`, then the first entry in the priors are used for all the columns.

3. **log_prior** (logarithmic prior probability): the columns of a dummy array $\alpha_{IJ}$ are chosen with a uniform prior according to the priors in the namelist. The columns in the $vparams$ are then set by $vparams = 10^{\alpha_{IJ}}$.

The `$params$ namelist is used to set the $vparams$ array, the fields’ initial conditions $\phi_{init0}$, the pivot scale $k_{pivot}$, and the number of e-folds between horizon exit and the end of inflation for the pivot scale by $N_{pivot}$.

```fortran
&params
  N_pivot = 55.0
  k_pivot = 0.002
  dlnk = 0.4
  phi_init0 = 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0
  vparams(1,:) = -11.4 -11.0 -10.8 -10.6 -10.5 -10.4 -10.3 -10.2 -10.1 -9.9
  vparams(2,:) = 8.8 7.3 6.0 4.8 3.6 2.6 1.7 1.0 5.6 0.16
  vparams(3,:) = 1e-3 1e-3 1e-3 1e-3 1e-3 1e-3 1e-3 1e-3 1e-3 1e-3
  vparams(4,:) = 1e-2 1e-2 1e-2 1e-2 1e-2 1e-2 1e-2 1e-2 1e-2 1e-2
/
```

These are all default values and may be overridden with different choices of sampling, as mentioned above. The variable $dlnk$ is the difference in $k$-space that is used when we evaluate the pivot-scale observables from the mode equations via finite difference. All scales are in units of $\text{Mpc}^{-1}$ and fields are in $M_{Pl}$. 

---

...
The prior probability ranges for the sampling of the fields’ initial values and $N_*$ are in the &priors namelist, which is relatively self-explanatory.

```plaintext
&priors
  phi0_priors_min = 2.0 2.0
  phi0_priors_max = 30.0 30.0
  dphi0_priors_min = -1.262e0 -1.262e0
  dphi0_priors_max = 1.262e0 1.262e0
  N_pivot_prior_min = 30
  N_pivot_prior_max = 70
/
```

MULTIMODECODE defaults to obtaining only the pivot-scale observables by taking the numerical derivative of the power spectra at $k_*$. However, the full power spectra can also be solved for and provide the full description of the model’s predictions over the scales of interest, as in §5.2. We control this through the &full_pk namelist.

```plaintext
&full_pk
  calc_full_pk = .false.
  steps = 300
  kmin = 1.0e-4
  kmax = 1.0e0
/
```

The variable calc_full_pk options the calculation of the full $P(k)$. The program will interpolate between a number of points, given by the variable steps, between the scales kmin and kmax.

MULTIMODECODE is able to save and output a significant amount of data for later analysis. However, since this has an obvious affect on the speed of the code, the amount and verbosity of this output can be specified by the attributes of the out_opt instance of the print_options type in the print_out namelist.

```plaintext
&print_out
  out_opt%modpkoutput = .true.
  out_opt%output_reduced = .true.
  out_opt%output_badic = .false.
  out_opt%save_traj = .true.
  out_opt%fields_horiz = .false.
  out_opt%fields_end_infl = .false.
  out_opt%spectra = .false.
  out_opt%modes = .false.
/
```

If you want nothing to write to stdout (the terminal, usually), then set modpkoutput=.false.; if less output is requested, then set output_reduced=.true.. If output_badic=.false., then any set of parameters that do not lead to a successfully inflating universe are discarded and ignored; otherwise, they will be saved and output into the data files with dummy values for their spectra. The remaining attributes of out_opt controls what cosmological quantities are printed to file in addition to the pivot scale observables: save_traj records the background trajectory as a function of $N_*$; fields_horiz saves the field values as the pivot scale crossing the horizon; fields_end_infl saves the field values at the end of inflation; spectra will
record the superhorizon power spectra as a function of $N_e$; and modes prints all the mode functions during the entire evolution. Setting modes=.true. consequently results in a lot of output.

The output of MultiModeCode is saved in comma-delimited CSV files, with the first row corresponding to a header that names each column. Subsequent rows correspond to different samples of the same model with different parameters. To change the output, simply find the point where the header is written in the code, add a new column(s), and print out the desired quantity in the correct order.

Finally, some technical options are controllable via the &technical namelist, by changing the attributes of the tech_opt instance of the tech_options type. In particular, the choice of numerical integration scheme can be changed by the tech_opt%use_dvode_integrator flag. Setting this to .true. will invoke a backwards-difference formula method, which is suitable for stiff problems, while .false. will use a fourth-order Runge-Kutta integrator. Various accuracy settings for the integrators can also be controlled in this namelist, with the global behavior set by the variable accuracy_setting = -1, 1, 2. Using 1 sets a minimal amount of accuracy, which we find is suitable for calculating the adiabatic power spectrum for simple models. Increasing this to 2, increases the accuracy and, in particular, increases the accuracy as the evolution moves out of slow-roll, which we find is necessary to obtain resolved isocurvature spectra, as in Fig. 3. Lastly, you can override our choices for the absolute and relative error tolerances by setting accuracy_setting=-1 and manually changing the remaining attributes of tech_opt in the namelist, which is self-explanatory.

References


D. H. Lyth and A. R. Liddle, *The primordial density perturbation: Cosmology, inflation and the origin of structure*,.


