The Chaotic Inflationary Universe

Adam Furlong, Ciaran MacNamara, Cillian Grall

May 2023

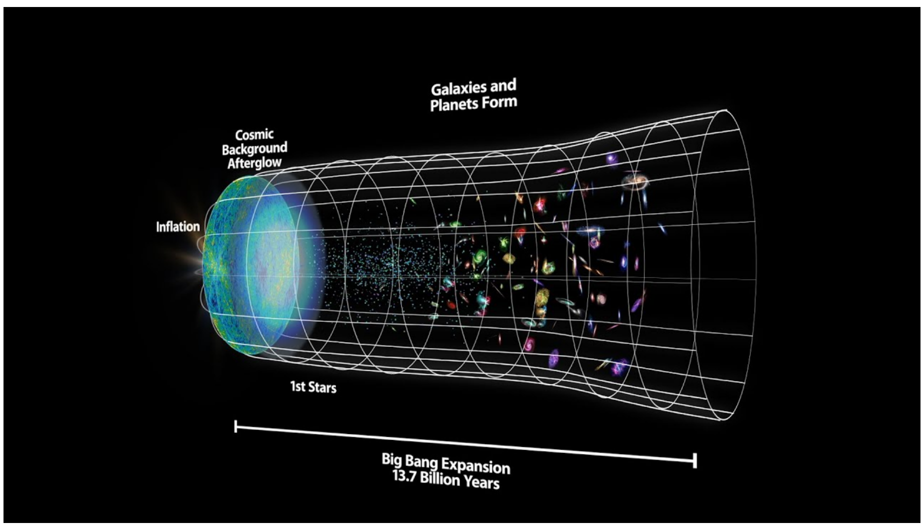

Figure 1: Expansion of the Universe

Abstract

The aim of this project is to investigate the expansion of the early universe in the case of an absence of spatial curvature. We primarily focus on cases where inflation occurs during expansion using the Klein-Gordon equation and the Friedman Equation. We does this by solving these equations using numerical methods.

1 Background

The pursuit of an accurate model of our universe has been a constant point of interest throughout the past century and still is to this day. There are many different models that have been constructed in attempts to describe our universe and its origin. Most of these models arose after Einstein published his paper on his gravitational field equations in 1915 and from his paper on General Relativity published in 1916. Einstein himself introduced a cosmological constant that allowed for a static solution (a solution in which the universe is neither expanding or contracting). Following this many other scientists attempted to create other cosmological constants for different solutions. One such scientist was Alexander Friedman who in 1922 mathematically predicted the expansion of the universe [3]. The main source of evidence that supported the claim of an expanding universe would be Edwin Hubble's paper released in 1929 which investigated the relationship between Redshift and distance [2]. It was not until the 1980s with theoretical developments which led to the creation of accurate models describing cosmic inflation. During a period of inflation the metric describing the system would change exponentially as the system expands at an exponential rate.

2 Introduction

The chaotic inflationary model is one of the simplest models of the inflationary universe as it does not take into account spatial curvature. The equations describing the evolution of the early Universe dominated by a scalar field in the absence of spatial curvature can be written in the form.

$$H = \frac{\dot{a}}{a} = [\frac{8\pi}{3M_p^2}(\frac{\dot{\phi}}{2} + V(\phi))]^{\frac{1}{2}}$$

This is the Friedman equation and is one of the two equations we will primarily work with. The other equation we primarily use is the Klein Gordon Equation.

$$\ddot{\phi} + 3H\dot{\phi} + \frac{dV}{d\phi} = 0$$

. These equations are written using natural units where M_p is the Planck Mass and T_p is the Planck time. Our choice of the scalar field is as follows:

$$V(\phi) = \frac{1}{2}m^2\phi^2$$

where m is the mass associated with the scalar field. The primary aim is to obtain the cosmic scale factor $a(t)$ which describes the changing distance between two points as the Universe expands. In particular we have exponential expansion when $log(a)$ is progressing linearly. In this report we will investigate the effects of different initial conditions for the Klein-Gordon equation and the Friedman equation to obtain scenarios in which the universe undergoes exponential expansion.

3 Basic Equations and Model

Recall the Friedman equation:

$$H = \frac{\dot{a}}{a} = (\frac{8\pi}{3M_p^2}[\frac{\dot{\phi}}{2} + V(\phi)])^{\frac{1}{2}}$$

And the Klein-Gordon equation:

$$\ddot{\phi} + 3H\dot{\phi} + \frac{dv}{d\phi} = 0$$

From eq.1 we get the following

$$H = \frac{\dot{a}}{a} = \frac{dlog(a)}{dt}$$

from here we can rewrite the two ODEs in the form

$$\frac{dlog(a)}{dt} = ((\frac{8\pi}{3M_p^2}(\frac{\dot{\phi}}{2} + V(\phi)))^{\frac{1}{2}})$$

$$\ddot{\phi} = -(3H\dot{\phi} + \frac{dV}{d\phi})$$

$$\ddot{\phi} = \frac{d\dot{\phi}}{dt}$$

Interestingly we will see for a large number of initial conditions and a scalar potential $V(\phi) = \frac{1}{2}m^2\phi^2$

$$H = \frac{d\log a}{d\tau} \cong constant$$

for $t \gg t_p$ from here we can approximate the Klein-Gordon equations as a damped harmonic oscillator:

$$m\ddot{x} + b\dot{x} + kx \longrightarrow \ddot{\phi} + 3H\dot{\phi} + m^2\phi = 0$$

for a solution : $\phi(t) = Ae^{-\lambda t}\cos(\omega t)$ with $\omega = \sqrt{m^2 - \frac{3H^2}{4}}$ and $\lambda = \frac{3H}{2}$ at $t = tp$ $A = \phi_0$ with a frequency $\omega = 2\pi f \rightarrow \frac{1}{2\pi}\sqrt{m^2 - \frac{9H^2}{4}}$ if

$$9H^2 \gg 4m^2 \longrightarrow overdamped$$

$$9H^2 < 4m^2 \longrightarrow underdamped$$

$$9H^2 = 4m^2 \longrightarrow critallydamped$$

$$\phi(t) = \phi_0 e^{-\frac{3H}{2}t}\cos(\sqrt{m^2 - \frac{3H^2}{4}}t)$$

4 Exploration of initial conditions

Next we shall consider what types of initial conditions are particularly interesting or illuminating.

4.1 $\dot{\phi}_0 = 0$ and the Effect of ϕ on Early Expansion

Let us consider the a field with a massive scalar potential $V(\phi) = \frac{1}{2}m^2\phi^2$. This yields the following equations.

$$\ddot{\phi} + 3H\dot{\phi} + m^2\phi = 0$$

$$H = (\frac{4\pi}{3M_p^2}[\dot{\phi}^2 + m^2\phi^2])^{\frac{1}{2}}$$

from here we can fix $\dot{\phi}_0 = 0$ which gives $H_0 = (\frac{4\pi m^2\phi_0^2}{3M_p^2})^{\frac{1}{2}} \rightarrow H_0 \propto \phi_0$ is always positive, however $\ddot{\phi}_0 = -M^2\phi_0$ depends on the sign of ϕ_0 . We define $\phi_0 \in I = \{-i, i\}$ a specific interval which after $t_p \times 10^6$ generates the following graph (assuming $V(\phi) = \frac{1}{2}m^2\phi^2$ and $m = M_p \times 10^-5$)

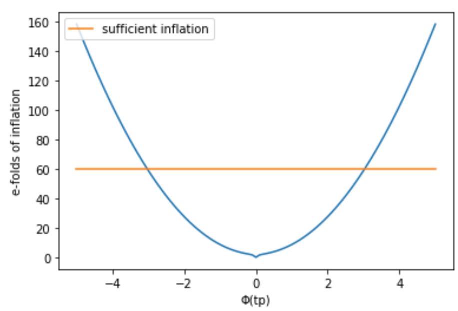

Figure 2: e-folds for $V(\phi)$. $m = 0.05M_p$

Notice that provided $|\phi_0| > 3$, we get sufficient inflation. Anything under this and we enter the oscillating phase too soon. So we can easily conclude that even restricting ourselves to $\dot{\phi} = 0$ we can still achieve sufficient inflation.

4.2 Fixed $H_0 = M_p$

Next we assume $H_0 = M_p$ which for a given ϕ_0 fixes $\dot{\phi}_0$ up to a sign. In particular let $\dot{\phi}_0 = \alpha \cos(k)$ and $m\phi_0 = \alpha \sin(k)$. combing this with eq gives:

$$H_0 = M_p = (\frac{4\pi}{3M_p^2}(\alpha^2 \cos^2(k) + \alpha^2 \sin^2(k)))^{\frac{1}{2}} = \sqrt{\frac{4\pi}{3M_p^2}\alpha^2}$$

$$\alpha = M_p^2 \sqrt{(\frac{3}{4\pi})}$$

Now we can consider the circle $k \in (-\pi, \pi]$ with initial conditions $X_o = [0, \frac{\alpha \sin(k)}{m}, \alpha \cos(k)] = [log(a), \dot{\phi}_0, m\phi_0]$. Graphed below is a system which expands in polar coordinates with z-axis depicting spatial expansion, radial expansion corresponds to passage of time and the angle determines the initial conditions, beside it is the same system but "unravelled" so that the appropriate angle is described on the x-axis.

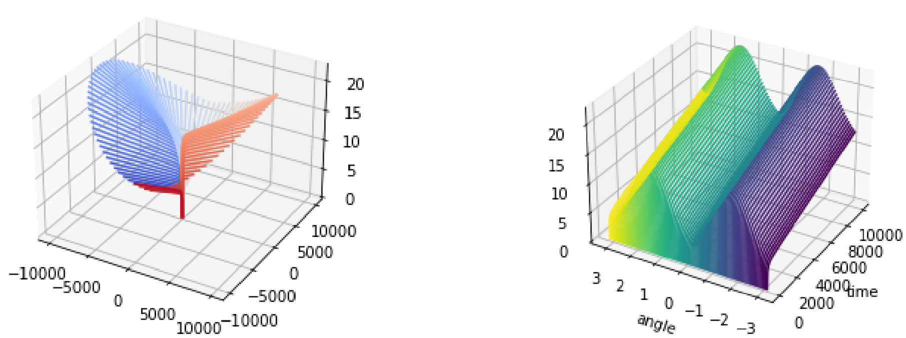

Figure 3: effect of angle k with H_0

There is a clear positive relation between the size of $\phi = \alpha \sin(k)$ and a rapid spurt of early expansion. However the time interval in figure 3 is too short to see if this would have an significant impact over longer timescale, instead we should look at the graph 4.23.

We can go even further and consider the disc $H_0 \leq M_p$ in the input space $m\phi_0$, $\dot{\phi}$ after some specific time T. If we graph the e-folds of inflation against this disc corresponding to inputs in the range $H_0 \leq M_p$, with our axis in terms of $m\phi_0$, $\frac{d\phi}{dt}$ and expansion after T.

Traversing along the $\dot{\phi}$ axis, we retrieve the graph in index 4.21. Traversing along the $m\phi$ axis, we should retrieve the graph in index 4.22, though that isn't clear from the above image. Traversing along the circumference of this disc, we recover the graph 4.23. Overall, for a fixed value of H_0, the amount of inflation underwent by the system seems to be far more sensitive to a change in ϕ_0 than a change in $\dot{\phi}$, we postulate that this is due to the damping term in the Klein-Gordon equation $3H\dot{\phi}$ which for large values of $\dot{\phi}$ exerts a powerful viscous effect on the initial expansion. Furthermore we can say that for $H_0 \leq M_p$, achieving sufficient inflation is very likely for values of $k \in (-\pi, \pi]$.

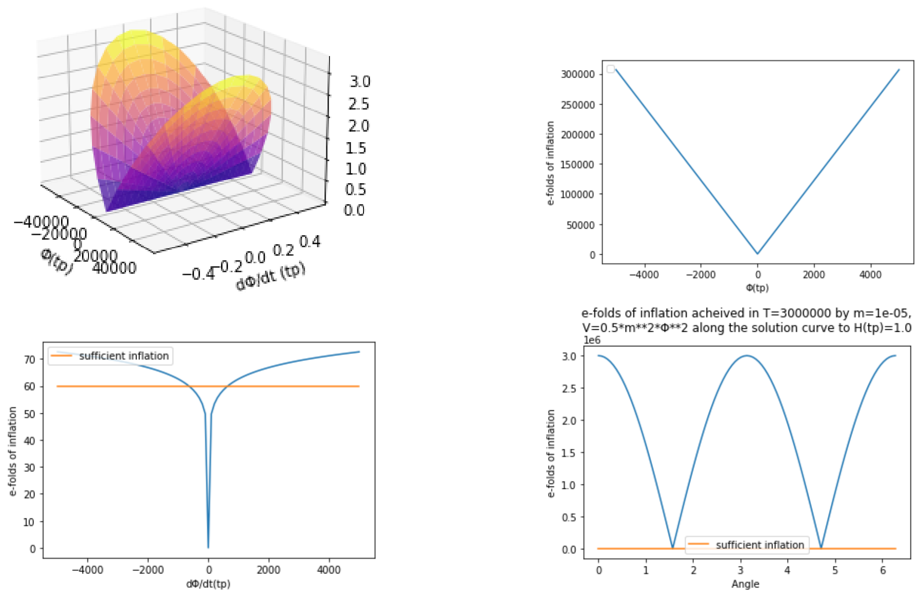

Figure 4: Figures 4.20-4.23 arranged clockwise from top left

4.3 General H_0

If we now consider how varying the initial value H_0 , we introduce two new degrees for freedom specifically in terms of ϕ and $\dot{\phi}$. Below in the graph we see how this effects inflation. Interestingly, for very small values of ϕ, $\dot{\phi}$, we see that inflation is far more dependent on the initial velocity than the initial position. Perhaps this is because at such small values, H is very small, and so the damping the system feels is very small, allowing the system to move higher up the potential, before inflating on its way down. Yet for the scales at which $\dot{\phi}_p$ is more important, we see that the system does not experience sufficient inflation.

As we zoom out, we notice a diagonal valley appear, along inputs whose sign differ. This valley represents the initial conditions that send the system deep into the potential very early, so that the oscillating phase is entered very early and little inflation is felt.

As we zoom out more, we notice that ϕ becomes the stronger of the two initial conditions. It is around this scale that some initial conditions are meeting sufficient inflation in the given time-frame.

Zooming out to the scale that fits the curve $H_p = M_p$, we notice that ϕ is completely dominating. However, we also notice that if we move along the $\dot{\phi}$ axis, as ϕ increases, and so H increases, the inflation increases linearly. This is a symptom of the fact that these systems are still in their inflation stages, and $log(a)$ is increasing linearly with time with almost the initial value of H. Eg. Look at the point maximum ϕ, on the $\dot{\phi}$ axis. Here $H_p = M_p$ while ϕ is very large. As ϕ is very large, the system remains high in the potential for a very long time. On its slow climb down the potential however, H remains very close to M_p, and so $log(a)$ increases linearly, corresponding to exponential inflation for an exaggerated period of time. If we were to run these systems for much longer we should expect to see a different shape in our final graph. We expect that specific example to reach 1.5×10^{10} e-folds of inflation.

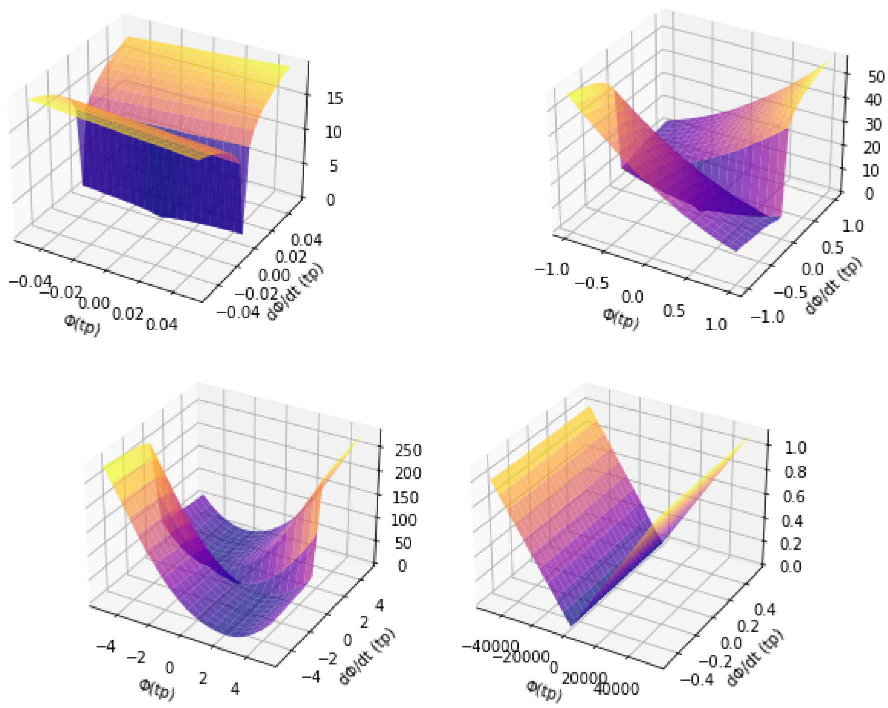

Figure 5: effect of angle k on ϕ (left image) and Hubble constant (right image)

4.4 Impact of mass m

Another variable with which we can tinker with is the mass m of the scalar field potential. In fact we can consider how a range of masses would effect the evolution of our system. i.e $m \in [0, 50 \times M_p] = I$ which for some initial conditions would produce a graph in the form of Figure 5. In general the presence of higher mass accelerates the evolution of the system inducing a rapid oscillation of the Klein-Gordon terms. For greater values of m the system experiences prolonged acceleration. Compared to initial conditions the mass of the scalar field continues to exert it's influence long after the initial burst of inflation. These influences include the frequency of the Klein-Gordon terms and Hubble constant as well as determining the rate of latter expansion . We examine the effect of mass over $10000t_p$ with the following initial conditions over the range $(0, 50M_p)$. Observe that near $m = 0$ a small change in mass causes a large discrepancy in the final states of the system. We can also examine a much smaller mass range between $(0, 1e-7)$ over a far longer period of time ,$1 \times 10^8 t_p$: We also noticed that in general larger values of m are more computationally intensive

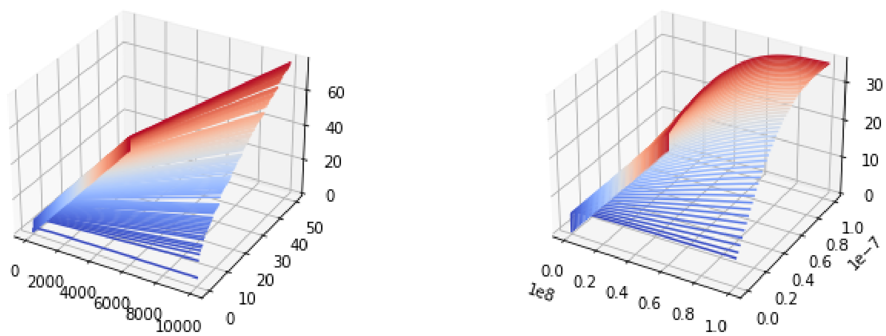

Figure 6: effect of mass on expansion

5 The Chaotic interplay of the Klein-Gordon terms

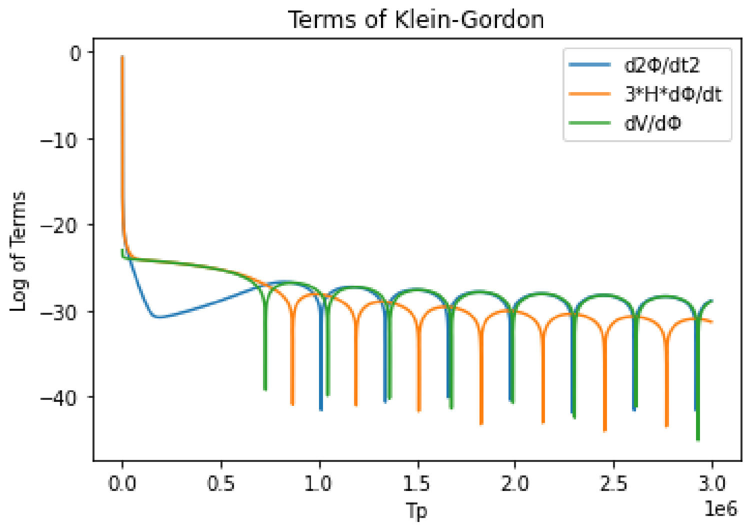

Figure 7: initial conditions $X_0 = [H_0 = 1, \phi_0 = 1, \dot{\phi_0} = \phi_-]$

$\dot{\phi}_- = \sqrt{H_0^2 \frac{3M_p^2}{4\pi} - 2V(\phi)}$

An evaluation of the interaction of the respective terms of the Klein-Gordon equation : $\ddot{\phi}$, $3H\dot{\phi}$, $m^2\phi$.If we consider the system described by

$$\ddot{\phi} + 3H\dot{\phi} + m^2\phi = 0$$

initially our system undergoes a substantial damping effect induced by the viscosity of the $3H\dot{\phi}$ term which in turn slows down the rate of expansion of the universe inducing a rapid erosion of $H = \frac{d\log(a)}{dt}$ to a stable but small constant. Indeed in the above illustrated case with m = $M_pe - 5$ this occurs after 1e6 plank times. Since H is roughly constant the frequency of the $m^2\phi$ terms is :

$$f = \frac{1}{2\pi}\sqrt{m^2 - \frac{3H^2}{4}} \approx \frac{1}{2\pi}\sqrt{m^2} \propto m$$

for a sufficiently small H. It also holds for sufficiently negligible H that:

$$\phi(t) = \phi_0 e^{-\frac{3H}{2}t} \cos(\sqrt{m^2 - \frac{3H^2}{4}}t) \approx \phi_0 \cos(m)$$

and therefore

$$\ddot{\phi}(t) \propto \cos(m)$$

and

$$\dot{\phi} \propto \sin(m)$$

and thus have roughly the same frequency as ϕ for small values of H. The affect of mass on the frequency of the Klein-Gordon terms, specifically $\ddot{\phi}$, is illustrated in the diagram below which varies in mass from $0 \longrightarrow \frac{M_p}{2}$ over $t_p \times 100$ with darking of blue en-coding the respective increase in mass in the graph below. From the graph we can see that the frequency of the oscillation is indeed proportional to the mass of the field.

We can conclude from this that the undulation of the Klein-Gordon terms depends acutely on the mass of the potential $V(\phi)$. The larger value of m not only corresponds to a greater frequency in the latter oscillation of the Klein-Gordon terms but also hastens the decay of H, quickening the arrival of this under-damped harmonic phase. We can exploit this knowledge to save computationally resources by approximating how the Klein-Gordon terms act for large timescales and small masses by examining shorter time frames with larger masses in the potential. We can investigate how how initial conditions effect their evolution by considering how the Klein-Gordon terms act with the following initial conditions: $X_o = [0, \frac{\alpha \sin(k)}{m}, \alpha \cos(k)]$, mass $= M_p$ over $10 \times t_p$.

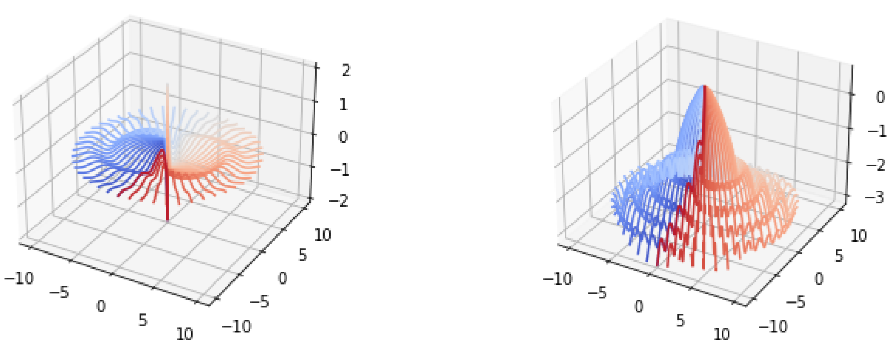

Figure 8: effect of angle k on ϕ (left image) and Hubble constant (right image)

(changing colour : cool$\longrightarrow$warm encapsulates changing angle $-\pi \longrightarrow \pi$)

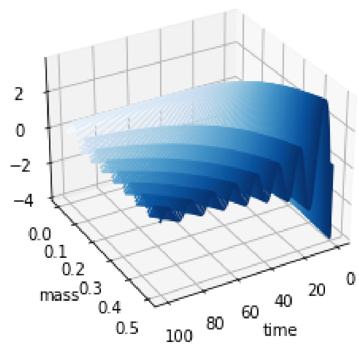

Figure 9: effect of mass on $\ddot{\phi}$

6 Reproducibility and comparison of model to prior papers

In order to probe the validity of our model we investigated prior papers on chaotic inflation that existed in the literature. Primarily we consulted a 1988 paper, [1], titled "Chaotic Inflation" by Mark S.Madsen and Peter Coles with the specific intention of replicating their graphs for initial conditions $\phi_0 = 1$ and $H_0 = M_p$. To do this we must evaluate $\dot{\phi}_0$ given ϕ_0. Substituting ϕ_0 into the Friedman equations produces:

$$H_p = \sqrt{\frac{8\pi}{3M_p^2}}\sqrt{\frac{1}{2}\dot{\phi}_0^2 + V(\phi_0)}$$

Rearranging gives :

$$\dot{\phi}_0^2 = (H_p^2\frac{3M_p^2}{4\pi}) - 2V(\phi_0).$$

which yields to us a positive and negative value for $\dot{\phi}_0$. As in accordance with the 1988 paper we assume $V(m,\phi) = \frac{1}{2}m^2\phi^2$ and $m = \frac{1}{10^5}M_p$ and graph these choices for $3 \cdot 10^6 t_p$.

The positive choice produces:

The negative choice produces:

These are indeed remarkably similar to the 1988 paper. However there were problems with these results as originally in the project brief we were provided with the Klein-Gordon equation as

$$H = \frac{8\pi}{3M_p^2}(\frac{1}{2}\ddot{\phi}^2 + \frac{1}{2}m^2\phi^2)^{\frac{1}{2}}$$

which gave us that the time axis appear to have contracted by approximately a factor of 3. Upon delving into the relevant literature we discovered that the

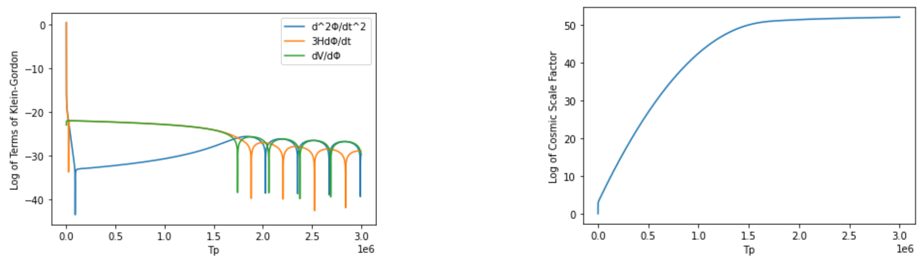

Figure 10: e-fold expansion and Klein-Gordon For positive ϕ_0

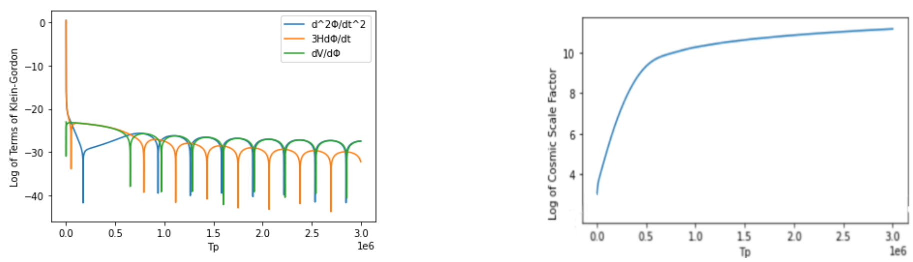

Figure 11: e-fold expansion and Klein-Gordon For negative ϕ_0

Klein-Gordon equation provided was incorrect. The correct equation is given as follows:

$$H = [\frac{8\pi}{3M_p^2}(\frac{1}{2}\ddot{\phi}^2 + V(\phi))]^{\frac{1}{2}}$$

This then gave us results that coincided with 1988 paper.

In relation to the dynamics we see in both graphs we see an initial period, the first three spikes in the Klein-Gordon graphs, that are as a result of the effect the initial value of $\dot{\phi}$. The strong viscous damping term $3H\phi$ quickly annihilates this initial $\dot{\phi}$. Then we have a period of inflation as ϕ decays, approximately as an over-damped oscillator. It is in this period that $log(a)$ has sustained linear growth, corresponding to exponential growth of the Cosmic Scale Factor a. The third phase such that as ϕ reaches the bottom of the potential, the damping term becomes weaker, allowing ϕ to oscillate about the bottom of the potential well, approximately as an under-damped oscillator. This is evidenced by the series of spikes in the Klein-Gordon terms $\phi, \dot{\phi}, \ddot{\phi}$ corresponding to each of repeatedly crossing zero. The majority of inflation in both scenarios occurred during. Neither scenario achieved the condition of "sufficient inflation" as neither achieved 60 e-folds of inflation.

7 Exploration Of Different Potentials

Now we shall consider the effect of different field potentials on the evolution of the Universe. Firstly we will note that

$$\cos(x) = 1 + \frac{1}{2}x^2 - \frac{1}{24}x^4 + ...$$

therefore

$$m^2(\cos(\phi) - 1) = m^2(\frac{1}{2}\phi^2 - \frac{1}{24}\phi^4 + ...) \approx \frac{1}{2}m^2\phi^2$$

Hence we would expect that for small values of ϕ we would expect that $V_c = m^2(\cos(\phi) - 1) \approx V_m = \frac{1}{2}m^2\phi^2$ and that as we approach inflection at $\phi - = \frac{\pi}{2}$ the potential would gradually diverge, becoming completely different by first turning point at $\phi_{=}\pi$. Due to the undulating nature of V_c we expect to produce periodic crests and valleys when we plot the varying expansion rate of different initial ϕ_0. Indeed setting $\dot{\phi}_0 = 0$ generates the following after $3 \times 10^6 t_p$ for V_c as compared to what V_m produces.

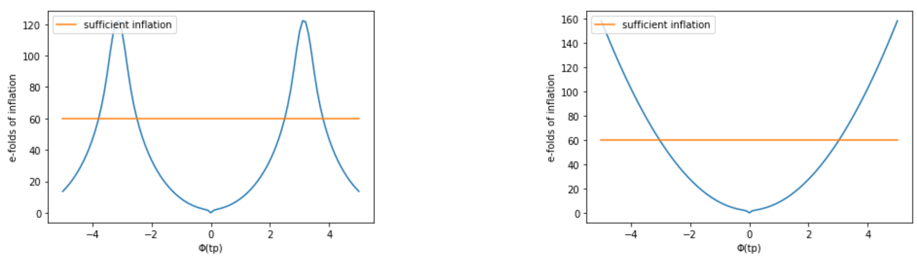

Figure 12: V_c (Left) and V_m (Right)

Note that phi equals $\pm\pi$ should experience indefinite constant exponential inflation
We can compare the potentials V_c and V_m over the input space $\dot{\phi}$ and ϕ after time $t = t_p \times 10^7$.

This potential allows us to explore how the system behaves with a local maximum in a potential. Placing ϕ in the region of a local maximum, with a small $p\dot{h}i$, doing so causes the Hubble Parameter to have a relatively large value. This will act against $\dot{\phi}$ increase due to the slope of the potential, which will be small anyway. This means that ϕ will remain near the maximum for a sustained period of time, causing H to be near that original value for a sustained period of time. Thus the system experiences a sustained period of inflation. The amount of inflation depends on the size of the local maxima.

We expect something similar to happen in a false minimum, or with a non-zero ground state. Take the potential

$$V_\epsilon = \frac{1}{2}m^2\phi^2 + \epsilon$$

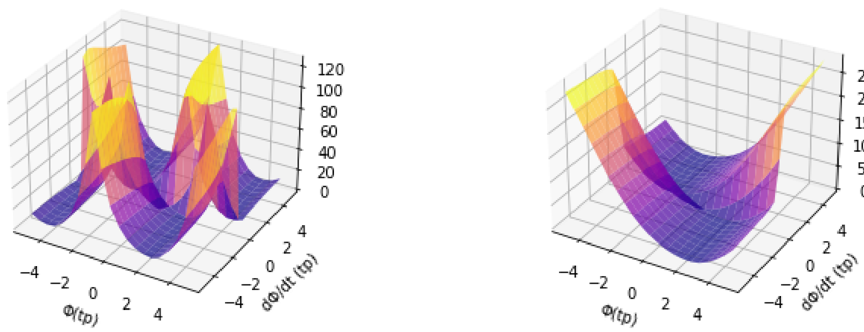

Figure 13: 3D graphs of potentials V_c (Left) and V_m (Right) over the input space $\phi, \dot{\phi}$

which is a modified version of the standard potential.We should expect that even when the system has ϕ at a potential minimum and $\dot{\phi}$ is zero.

$$H = \sqrt{\frac{8\pi}{3M_p^2}\epsilon} = \frac{d}{d\tau}\log(a)$$

and thus

$$a(t) = exp(\sqrt{\frac{8\pi}{3M_p^2}\epsilon t})$$

Therefore we would expect inflation even from initial conditions where $\phi = \dot{\phi} = 0$. If we consider the graph of inflation generated after $T = 3 \times 10^6 t_p$ for some $\epsilon = 0.25m^2$ we see that sufficient inflation is achieved in all but $\phi \in (-2, 2)$ even with $\dot{\phi}_0 = 0$. By comparing to the graph of the standard potential, we see that this graph has been raised by just over 40 e-folds of inflation.

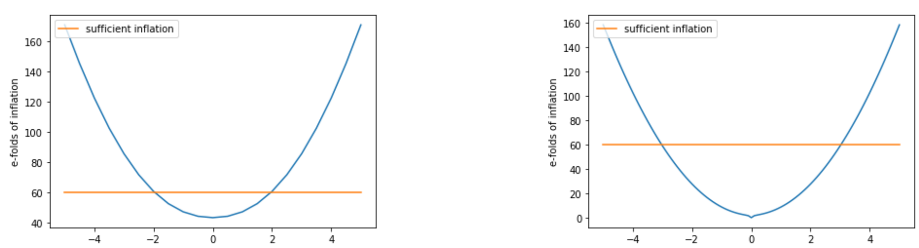

Figure 14: Graph of $V_\epsilon(\phi)$ (Left) and $V_m(\phi)$ (Right)

8 Explanation of Code

Our programme is designed to compute the evolution of a universe with no spatial curvature as described by the Friedman and Klein-Gordon equations. Importantly for our computation we rewrite them as in figure 3 :

$$\frac{d\log\dot{a}}{d\tau} = (\frac{8\pi}{3M_p^2}[\frac{\dot{\phi}}{2} + V(\phi)])^{\frac{1}{2}}$$

$$\ddot{\phi} = -(3H\dot{\phi} + \frac{dV}{d\dot{\phi}})$$

$$\ddot{\phi} = (\frac{d\dot{\phi}}{d\tau})$$

If we consider the scalar potential to be

$$V(\phi) = \frac{1}{2}m^2\phi^2$$

there are a number of different ways we can go about solving the equations numerically. From **SciPy.integrate** we imported odeint which we used to solve the relevant ODEs. Our primary function is "chaos" which computes the evolution of the universe for a given set of initial conditions and scalar field potential.

Taken together these 4 sections of code below provide a robust foundation for which to compute the various plots and graphs which illustrate this report. It is however pertinent to address the limitations of this programme before concluding;
firstly in the pursuit of a general function capable of dealing with a variety of potentials we had to sacrifice the efficiency that simpler programme would have provided.
Secondly since the derivative of V(ϕ) is calculated symbolically from **sympy** it requires an input potential to be *sympy-defined* which reduces slightly the generality of the programme.

9 Further Research

Areas of further research include expanding our model to accommodate spatial curvature and quantum mechanic effects. If you have potential with which has a false minimum, the system will inflate for a period but then may the system will decay to period oscillation and

10 Conclusion

The chaotic inflationary model of the expansion of the universe is interesting from a theoretical point of view, even using simple models help gives us a grasp of the dynamics of the expansion and how various terms in the equations affect the dynamics. Throughout the course of this project we developed a set of computational methods that would help solve these systems. While our main focus was on chaotic inflation using a standard scalar potential $V(\phi) = \frac{1}{2}m^2\phi^2$ in the absence of spatial curvature, the methods we developed, as we have shown, can be used to describe system with more complex potentials and in the future, systems where spatial curvature is present. We have shown that there many ways to achieve suitable inflation, such as in the case of ϕ_0. Some of the initial conditions which give rise to inflation such as cases with sufficiently large ϕ, sufficiently large ϕ, potentials with a non-zero minimum, system with a large mass associated with the scalar potential.

11 Acknowledgements

We would like to thank the Theoretical Physics department for our supporting our endeavour to complete this project. We thank Professor Peter Coles for giving us the opportunity to study this topic, a topic he himself studied in far greater depth.

12 References

[1] Coles, P., Madsen, (1988) M. *Chaotic Inflation*, Astronomy Centre, University of Sussex, Brighton BN1 9QH.
[2] A. Bahcall, N. (2015) Hubble's Law and the expanding universe, Proceedings of the National Academy of Sciences Vol. 112, No. 11
[3]Siegried, T. (2022) A century ago, Alexander Friedmann envisioned the universe's expansion. Science News

(Note Latex doesnt recognise the symbol representing phi, so in the latex code it is present, but in the printed python section it is not)

```

"""
Defining the 'chaos' function and its pre-requisites
Throughout {x, y, z, La, fp} denote {  , d  /dt, d^  /dt^2, log(a), f(tp)}
Vs is a sympy-defined version of a potential V(m,   )
"""
import numpy as np
import sympy as sp
from scipy.integrate import odeint

#rename a common constant c=sqrt(8  /3Mp^2)
c  = np.sqrt(np.pi*8/3)
#standard mass
m0 = 10**-5

def chaos(m, Vs, xp, yp, T, dt):
    """
    ----------
    m  : constant associated with V
    Vs : V(m,   ),   a non-negative sympy function
    xp :   (tp),     the initial value of the scalar field
    yp : d  /dt(tp), the intial value of the derivative of the scalar field
    T  : >1  Simulation end time, in tp
    dt : >0  Timestep increment,  in tp
    ----------
    Uses odeint to solve the ODE IVP
    given by xp, yp, and the Friedmann and Klein-Gordon equations
    and returns an array of [  , d  /dt, log(a)] over time.
    The log( a(T) ) is referred to as the e-fold number,
    the number of times the Cosmic Scale Factor
    increased by a factor of e.
    It is the entry [-1][2] of the chaos output.
    ----------
    """
    def deriv(u, t):
        return DERIV(Vs, m, u, t)

    t   = np.arange(1, T+dt, dt)
    #time axis over which to evaluate
    Lap = 0.0
    #ap=1, so Lap=log(ap)=0
    up  = [xp, yp, Lap]
    #initial vector u = [  , d  /dt, log(a)]

    U   = odeint(deriv, up, t)
    #solve the IVP
    return U





def DERIV(Vs, m, u, t):
```

```
    """
    ----------
    m  : constant associated with V
    Vs : V(m,   ),   a non-negative sympy function
    u  : [  , d  /dt, log(a)]
    t  : time axis over which to evaluate
    ----------
    Returns [d  /dt, d^2  /dt^2, d(log(a))/dt]
    Calculated according to the Friedman and Klein-Gordon equations
    ----------
    """
    def V(m, x):
        return float( Vs(m, x) )
    #returns a float answer of Vs
    #behaves better inside other functions

    h   = Hubble(m, V, u[0], u[1])
    #Friedmann
    y   = u[1]
    #simply the derivative
    z   = -3.0*h*(u[1]) - dVd (Vs, m, u[0])
    #Klein-Gordon
    return [y, z, h]



def Hubble(m, V, x, y):
    """
    ----------
    m  : m          constant associated with V
    V  : V(m,   )    a non-negative function, the potential
    x  :             the value of the scalar field
    y  : d  /dt      the value of the derivative of the scalar
    field
    ----------
    Returns the Hubble parameter
    Defined as sqrt( (8  /3Mp^2)*(1/2 *(d  /dt)^2  + V(m,   )))
    """
    h = c*( ( 0.5*(y**2) + V(m, x) )**0.5 )
    #from Friedmann
    return h



def dVd (Vs, m, x):
    """
    ----------
    m  : m          constant associated with V
    Vs : V(m,   )    a non-negative sympy function
    x  :             the value of the scalar field
    ----------
    Returns the derivative of V(m,   ) with respect to
    Calculated symbolically using sympy
    This is the reason that we request a sympy-defined V
    ----------
    """
    pot  = Vs(m, sp.symbols('  '))
```

```
    #creates a symbolic version of the functionV(m,   )
    dpot = sp.diff(pot, sp.symbols('  '))
    #differenciate this symbolic function wrt
    f    = sp.lambdify(sp.symbols('  '), dpot)
    #turn the derivative into a numeric function
    #now return evaluated at x
    return f(x)
```

```
"""
Defining functions based on 'chaos',
to plot e-folds of inflation against initial conditions
Throughout {x, y, z, La, fp} denote {  , d  /dt, d^  /dt^2, log(a),
     f(tp)} respectively
Vs is a sympy-defined version of a potential V(m,   )
"""
import matplotlib.pyplot as plt
import numpy as np
import sympy as sp
from scipy.integrate   import odeint
from Project_chaos     import chaos, Hubble, dVd
from Project_helper_functions import phidotp, phip, Vm

#rename a common constant c=sqrt(8  /3Mp^2)
c  = np.sqrt(np.pi*8/3)
#standard mass
m0 = 10**-5
#the positive value of d  /dtp such that H( p =1, d  /dtp)=Mp
#used for testing against the 1988 graphs
y0 = phidotp(m=m0, Vs=Vm, xp=1.0, hp=1.0)
#R1 is the radius of the 'circle' of solutions { p , d  /dtp}
#such that Hp=Mp
R1 = ( (3/4) * (1/np.pi) )**0.5


def BP1(m, Vs, yp, xmin, xmax, dx, T, dt):
    """
    ----------
    m    : constant associated with V
    Vs   : V(m,   ),   a non-negative sympy-defined function
    yp   : d  /dt(tp), the intial value of the derivative of the
    scalar field
    xmin : minimum value of   (tp) to be tested
    xmax : maximum value of   (tp) to be tested
    dx   : increment between test values of   (tp)
    T    : >0 Simulation end time, in tp
    dt   : >0 Timestep increment,  in tp
    ----------
    Runs 'chaos' for each   (tp) in arange[xmin, xmax+dx, dx]
    Plots the e-fold number as a function of   (tp)
    ----------
    """
    #define the choices of   (tp) to test
    X = np.arange(xmin, xmax+dx, dx)

    #for each initial condition, solve the ODE with 'chaos'
    Z = np.zeros(len(X))
    for i in range(len(X)):
        Z[i] = chaos(m, Vs, X[i], yp, T, dt)[-1][2]
```

```

    #graph e-folds, over time
    plt.figure()
    plt.xlabel('  (tp)')
    plt.ylabel('e-folds of inflation')
    plt.plot(X, Z)

    #define the sufficient inflation to graph
    si= np.zeros(len(X))
    si= si+60
    plt.plot(X, si, label=str('sufficient inflation'))
    plt.legend(loc=2)
    return


def AntiBP1(m, Vs, xp, ymin, ymax, dy, T, dt):
    """
    ----------
    m    : constant associated with V
    Vs   : V(m,   ),   a non-negative sympy function
    xp   :  d  /dt(tp), the intial value of the derivative of the scalar field
    ymin : minimum value of  d  /dt(tp) to be tested
    ymax : maximum value of  d  /dt(tp) to be tested
    dy   : increment between test values of  d  /dt(tp)
    T    : >0 Simulation end time, in tp
    dt   : >0 Timestep increment,  in tp
    ----------
    Runs 'chaos' for each  d  /dt(tp) in arange[ymin, ymax, dy]
    Plots the inflation as a function of  d  /dt(tp)
    ----------
    """
    #define the choices of  d  /dt (tp) to test
    Y = np.arange(ymin, ymax+dy, dy)

    #for each initial condition, solve the ODE with 'chaos'
    Z = np.zeros(len(Y))
    for i in range(len(Y)):
        Z[i] = chaos(m, Vs, xp, Y[i], T, dt)[-1][2]

    #graph e-folds, over time
    plt.figure()
    plt.xlabel('d  /dt(tp)')
    plt.ylabel('e-folds of inflation')
    plt.plot(Y, Z)

    #define the sufficient inflation to graph
    si= np.zeros(len(Y))
    si= si+60
    plt.plot(Y, si, label=str('sufficient inflation'))
    plt.legend(loc=2)
    return


def circumference(m, hp, dtheta, T, dt):
    """
    ----------
```

```
    m      : constant associated with V
    hp     : H(tp), >0  the initial value of the Hubble parameter
    dtheta : >0 increment of Theta along the circle
    T      : >1 Simulation end time, in tp
    dt     : >0 Timestep increment,  in tp
    ----------
    Assumes the standard massive potential Vm(m,   )= 0.5 m^2    ^2
    Runs chaos on pairs of inputs (   (tp) ,d  /dt(tp))
    that lie on the circular solution curve to:
        hp = H(xp, yp) = sqrt(8  /3Mp^2)*sqrt(0.5*y^2 + V(m, xp))
    which is the circle:
        (  d  /dt(tp) )^2 + ( m*   (tp) )^2 = ( hp/ sqrt(2)*c )^2
    Graphs the e-folds underwent in T,
    over the circumference of this circle
    """
    #define the choices of Theta
    Theta  = np.arange(0.0, 2*np.pi + dtheta, dtheta)

    #radius of this 'circle'
    R      = (2**0.5)*hp/c

    #for each initial condition, solve the ODE with 'chaos'
    E     = np.zeros(len(Theta))
    for i in range(len(Theta)):
        x = R*np.cos( Theta[i] )/m
        y = R*np.sin( Theta[i] )
        E[i] = chaos(m, Vm, x, y, T, dt)[-1][2]

    #graph e-folds, over time
    plt.figure()
    plt.xlabel('Angle ')
    plt.ylabel('e-folds of inflation')
    plt.plot(Theta, E)

    #define the sufficient inflation to graph
    si = np.zeros(len(Theta))
    si = si + 60
    plt.plot(Theta, si, label=str('sufficient inflation'))
    plt.legend(loc=8)
    return



def BP1_M(M, Vs, yp, xmin, xmax, dx, T, dt):
    """
    ----------
    M    : array of constants associated with V
    Vs   : V(m,   ),   a non-negative sympy-defined function
    yp   :  d  /dt(tp), the intial value of the derivative of the scalar field
    xmin : minimum value of    (tp) to be tested
    xmax : maximum value of    (tp) to be tested
    dx   : increment between test values of    (tp)
    T    : >1 Simulation end time, in tp
    dt   : >0 Timestep increment,  in tp
    ----------
    Runs 'chaos' for each    (tp) in arange[xmin, xmax+dx, dx]
```

```
    and for each m in M, witu d  /dt(tp) fixed
    Plots the inflation as a function of   (tp)
    ----------
    """
    #define the choices of   (tp) to test
    X = np.arange(xmin, xmax+dx, dx)

    #set up graph
    plt.figure()
    plt.xlabel('  (tp)')
    plt.ylabel('e-folds of inflation')

    #for each choice of m,
    #for each choice of   (tp),
    #evaluate the number of e-folds acheived
    #then plot against the choice of   (tp),
    #labelled with the choice of m
    Z = np.zeros(len(X))
    for i in range(len(M)):
        for j in range(len(X)):
            Z[j] = chaos(M[i], Vs, X[j], yp, T, dt)[-1][2]
        plt.plot(X, Z, label=str('m='+str(M[i])))

    #now define the sufficient inflation to graph
    si = np.zeros(len(X))
    si = si + 60
    plt.plot(X, si, label=str('sufficient inflation'))
    plt.legend(loc=1)
    #loc=best is slow with large data volume
    return


def AntiBP1_M(M, Vs, xp, ymin, ymax, dy, T, dt):
    """
    ----------
    M    : array of constants associated with V
    Vs   : V(m,   ),   a non-negative sympy-defined function
    xp   :   (tp), the intial value of the scalar field
    ymin : minimum value of d  /dt(tp) to be tested
    ymax : maximum value of d  /dt(tp) to be tested
    dy   : increment between test values of d  /dt(tp)
    T    : >1 Simulation end time, in tp
    dt   : >0 Timestep increment,  in tp
    ----------
    Runs 'chaos' for each d  /dt(tp) in arange[ymin, ymax+dy, dy]
    and for each m in M, with   (tp) fixed
    Plots the inflation as a function of d  /dt(tp)
    ----------
    """
    #define the choices of d  /dt(tp) to test
    Y = np.arange(ymin, ymax+dy, dy)

    #set up graph
    plt.figure()
    plt.xlabel('  (tp)')
    plt.ylabel('e-folds of inflation')

```

```
    #for each choice of m,
    #for each choice of d  /dt(tp),
    #evaluate the number of e-folds acheived
    #then plot against the choice of d  /dt(tp),
    #labelled with the choice of m
    Z = np.zeros(len(Y))
    for i in range(len(M)):
        for j in range(len(Y)):
            Z[j] = chaos(M[i], Vs, xp, Y[j], T, dt)[-1][2]
        plt.plot(Y, Z, label=str('m='+str(M[i])))

    #now define the sufficient inflation to graph
    si = np.zeros(len(Y))
    si = si + 60
    plt.plot(Y, si, label=str('sufficient inflation'))
    plt.legend(loc=1)
    #loc=best is slow with large data volume
    return


def circumference_M(M, hp, dtheta, T, dt):
    """
    ----------
    M      : an array of constants associated with V
    hp     : H(tp), >0  the initial value of the Hubble parameter
    dtheta : the increment of Theta along the circle
    T      : >1 Simulation end time, in tp
    dt     : >0 Timestep increment,  in tp
    ----------
    Assumes the standard massive potential Vm(m,   )= 0.5 m^2   ^2
    For each m in M
    Runs chaos on pairs of inputs (  (tp) ,d  /dt(tp))
    that lie on the circular solution curve to:
        hp = H(xp, yp) = sqrt(8  /3Mp^2)*sqrt(0.5*y^2 + V(m, xp))
    which is the circle:
        ( d  /dt(tp) )^2 + ( m*  (tp) )^2 = ( hp/ sqrt(2)*c )^2
    For each m in M, graphs the inflation underwent in T,
    over the circumference of this circle
    """
    #define the choices of Theta
    Theta  = np.arange(0.0, 2*np.pi + dtheta, dtheta)

    #radius of this 'circle'
    R  = (2**0.5)*hp/c

    #set up graph
    plt.figure()
    plt.xlabel('Angle')
    plt.ylabel('e-folds of inflation')

    #for each choice of m,
    #for each choice of Theta,
    #evaluate the number of e-folds acheived
    #then plot against the choice of Theta,
    #labelled with the choice of m
    E  = np.zeros(len(Theta))
    for j in range(len(M)):
```

```
        for i in range(len(Theta)):
            x = R*np.cos( Theta[i] )/M[j]
            y = R*np.sin( Theta[i] )
            E[i] = chaos(M[j], Vm, x, y, T, dt)[-1][2]
        plt.plot(Theta, E, label=str('m=')+str(M[j]))

    #define the sufficient inflation to graph
    si = np.zeros(len(Theta))
    si = si + 60
    plt.plot(Theta, si, label=str('sufficient inflation'))
    plt.legend(loc=8)
    return
```

```



(3D graphs)

"""
Defining functions based on 'chaos',
to plot the e-folds of inflation
against initial conditions a 2d input space
Throughout {x, y, z, La, fp} denote {  , d  /dt, d^  /dt^2, log(a),
     f(tp)} respectively
Vs is a sympy-defined version of a potential V(m,   )
"""
import matplotlib.pyplot as plt
import numpy as np
import sympy as sp
from scipy.integrate   import odeint
from Project_chaos     import chaos, Hubble, dVd
from Project_helper_functions import phidotp, phip, Vm

#rename a common constant c=sqrt(8  /3Mp^2)
c  = np.sqrt(np.pi*8/3)
#standard mass
m0 = 10**-5
#the positive value of d  /dtp such that H( p =1, d  /dtp)=Mp
#used for testing against the 1988 graphs
y0 = phidotp(m=m0, Vs=Vm, xp=1.0, hp=1.0)
#R1 is the radius of the 'circle' of solutions { p , d  /dtp}
#such that Hp=Mp
R1 = ( (3/4) * (1/np.pi) )**0.5


def graph_3d_Vm(m, hp, T, dt, dr, dtheta):
    """
    ----------
    m      : constant associated with V
    hp     : >0, initial value of the Hubble Parameter
    T      : >1 Simulation end time, in tp
    dt     : >0 Timestep increment,  in tp
    dr     : >0 increment of r
    dtheta : >0 increment of theta
    ----------
    Assumes V(m,   ) = Vm(m,   ) = 1/2 m^2   ^2
    Returns a 3-d graph of the e-fold number
    over the initial conditons [ p , d  /dtp]
```

```
    such that H(tp) <= hp,
    a circle in the input space [m p , d /dtp]
    ----------
    """
    fig = plt.figure()
    ax = fig.add_subplot(projection='3d')

    # Create the mesh in polar coordinates and compute corresponding Z.
    r = np.arange(0, hp*(3/(4*np.pi))**0.5 + dr, dr)
    p = np.arange(0, 2*np.pi+ dtheta, dtheta)
    R, P = np.meshgrid(r, p)
    # Express the mesh in the cartesian system.
    X, Y = R*np.cos(P)/m, R*np.sin(P)

    z = np.zeros( len(r)*len(p))
    Z = z.reshape( len(p), len(r) )

    for i in range(len(r)):
        for j in range(len(p)):
            x = r[i]*np.cos(p[j])/m
            y = r[i]*np.sin(p[j])
            Z[j][i] = chaos(m, Vm, x, y, T, dt)[-1][2]

    # Plot the surface.
    ax.plot_surface(X, Y, Z, cmap="plasma", linewidth=0, alpha=0.7)

    ax.contourf(X, Y, Z, zdir='z', offset=0*Z.max(), cmap='coolwarm')
    ax.contourf(X, Y, Z, zdir='x', offset=-1.2*(hp*(3/(4*np.pi))**0.5)/m, cmap='coolwarm')
    ax.contourf(X, Y, Z, zdir='y', offset= 1.2*(hp*(3/(4*np.pi))**0.5),   cmap='coolwarm')

    ax.set_xlabel(' p ')
    ax.set_ylabel('d  /dtp')
    ax.set_zlabel('e-folds of inflation')
    ax.view_init(elev=20., azim=-35, roll=0)
    plt.show()
    return


def Graph_3d(m, Vs, xmin, xmax, dx, ymin, ymax, dy, T, dt):
    """
    ----------
    m    : constant associated with potential V
    Vs   : non-negative, sympy-defined function
    xmin :         minimum value of  p  to be tested
    xmax : >xmin   maximum value of  p  to be tested
    dx   : >0      increment between    values
    ymin :         minimum value of d  /dtp to be tested
    ymax : >ymin   maximum value of d  /dtp to be tested
    dy   : >0      increment between d  /dtp values
    T    : >1 Simulation end time, in tp
    dt   : >0 Timestep increment,  in tp
    ----------
    Returns a 3-d plot of the e-fold number
```

```
    over the initial a rectangular region
    in the input space [ p , d  /dtp]
    ----------
    """
    x = np.arange(xmin, xmax+dx, dx)
    y = np.arange(ymin, ymax+dy, dy)

    X, Y = np.meshgrid(x, y)

    z = np.zeros( len(x)*len(y) )
    Z = z.reshape( (len(x), len(y)) )


    for i in range(len(x)):
        for j in range(len(y)):
            Z[j][i] = chaos(m, Vs, x[i], y[j], T, dt)[-1][2]

    fig = plt.figure()
    ax = plt.axes(projection='3d')
    ax.plot_surface(X, Y, Z, cmap="plasma", linewidth=0, alpha=0.7)
    ax.set_xlabel('  (tp)')
    ax.set_ylabel('d  /dtp')
    ax.set_zlabel('e-folds of inflation')

    ax.contourf(X, Y, Z, zdir='z', offset=1*Z.min(), cmap='coolwarm')
    ax.contourf(X, Y, Z, zdir='x', offset=-1.2*xmax, cmap='coolwarm')
    ax.contourf(X, Y, Z, zdir='y', offset= 1.2*ymax, cmap='coolwarm')
    plt.show()
    return
```