# Generación de campos gaussianos en teorías inflacionarias

TFG: FS20-29-FSC

16 de noviembre de 2022

Universidad de Córdoba

Rafael Jurado Ariza

Abstract

1. Introducción

2. Objetivos

3. Materiales y métodos

4. Resultados

5. Conclusions

# Abstract

1. Introducción

2. Objetivos

3. Materiales y métodos

4. Resultados

5. Conclusions

# Abstract

In order to explain certain peculiarities that appear in the Big Bang theory, we resort to the concept of inflation. By treating this idea in a quantum way, a mechanism that produced the primordial density fluctuations that gave rise to the complex structures of the universe such as galaxies, stars and life itself appears.

This paper gives an introduction to the theoretical framework of the Big Bang and its problems and inflation, as well as the fluctuations that occurred in the primordial universe and the statistics needed to understand them. The initial distribution of these fluctuations can be shaped by means of a random density field with Gaussian statistics (which is fully characterized by its primordial power spectrum $\mathrm{P}_0(k) \propto k^{n_s}$), as predicted by inflationary theories. For the purpose to delve, beyond intuition, into the meaning of such a Gaussian statistics, an algorithm has been programmed in the Python language that is able to numerically simulate these fluctuations in the primordial universe and in the recombination epoch by incorporating the transfer function $T(k)$ into the power spectrum $\mathrm{P}(k) \propto k^{n_s} T^2(k)$. This is done by using tools such as the fast Fourier transform or random numbers, of which a theoretical introduction is also provided.

Finally, the numerical simulations for different $n_s$ values are represented by means of heat maps, which are the peak results of the work.

Abstract

**1. Introducción**

2. Objetivos

3. Materiales y métodos

4. Resultados

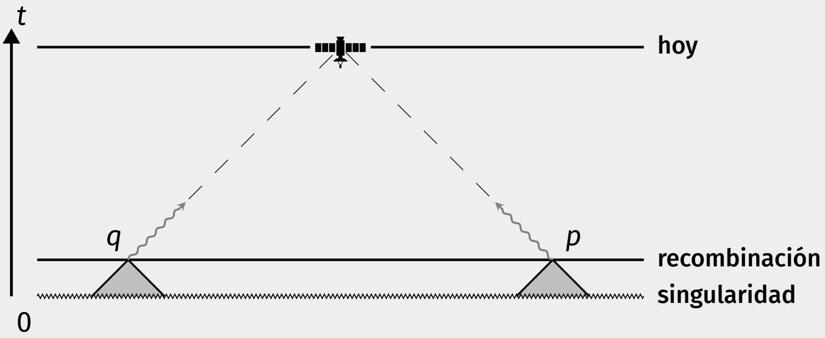
5. Conclusions

# Introducción

## Inflación

El Universo no es como las ecuaciones del *Hot Big Bang* orquestan. Aparecen una serie de problemas teóricos, entre ellos: **el problema del horizonte**

# Introducción



# Introducción

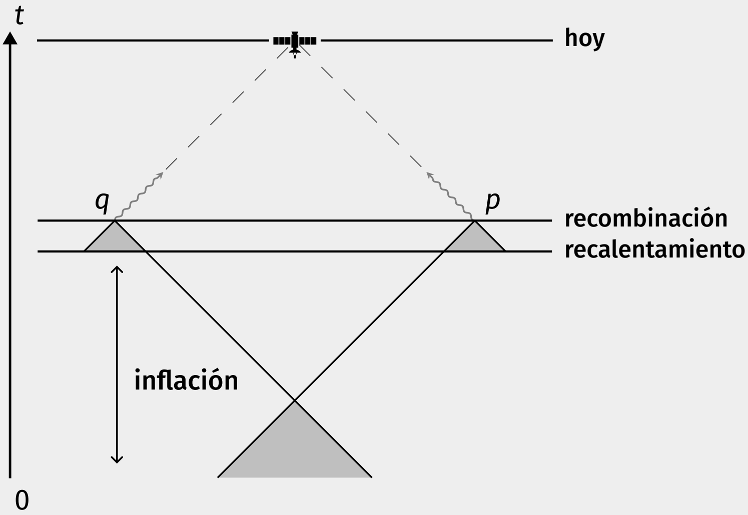
## Inflación

El Universo no es como las ecuaciones del *Hot Big Bang* orquestan. Aparecen una serie de problemas teóricos, entre ellos: **el problema del horizonte** y el **problema de la planitud**:

$$H^2 = \frac{8\pi G}{3}\rho - \frac{kc^2}{a^2}, \qquad \rho_c \equiv \frac{3H^2}{8\pi G} = \rho.$$

Inflación es capaz de resolver estos dos problemas y además provee de manera natural un mecanismo generador de las fluctuaciones primordiales.

# Introducción

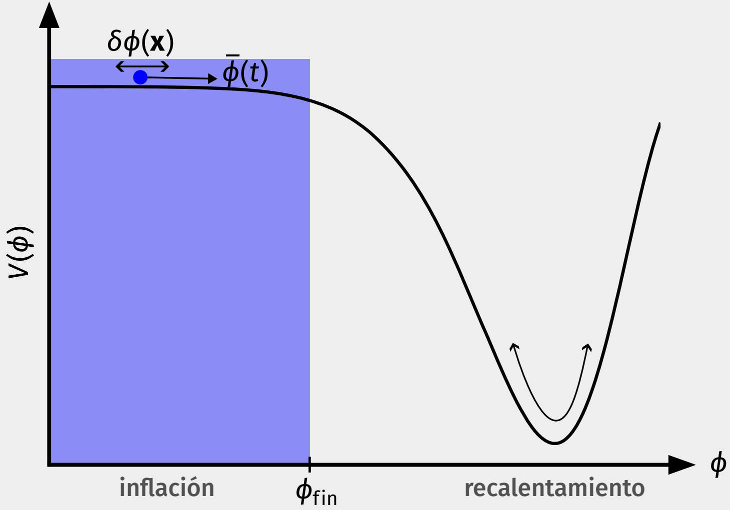


# Introducción

## Inflación

- Marco teórico $\longrightarrow$ Campos escalares
- El campo escalar es el **inflatón** $\phi(t, \mathbf{x})$
- Su ec. de estado $w = \mathcal{P}_\phi / \rho_\phi < -1/3$
- Tiene una densidad de e. potencial $V(\phi)$ y de e. cinética $\dot{\phi}^2/2$
- Su ec. de movimiento $\ddot{\phi} + 3H\dot{\phi} + V'(\phi) = 0$
- Escenario *slow-roll*

# Introducción



# Introducción

## Espectro de potencias y función de transferencia

Fluctuaciones cuánticas $\delta\phi(\mathbf{x}) \longrightarrow$ Fluctuaciones clásicas densidad de energía $\delta\rho(t, \mathbf{x})$

Estas fluctuaciones son descritas en el **régimen lineal**:

$$\delta(t, \mathbf{x}) = \frac{\delta\rho(t, \mathbf{x})}{\bar{\rho}(t)} \ll 1.$$

El espacio de Fourier entra en juego para facilitar el estudio de las fluctuaciones ("modos"). En cosmología nos importa el comportamiento estadístico de las cantidades estudiadas. El campo de fluctuaciones $\delta$ se modela con un **campo aleatorio**.

# Introducción

## Espectro de potencias y función de transferencia

Estamos interesados en la correlación espacial de las fluctuaciones:

$$\xi(\mathbf{x},\mathbf{x}',t) \equiv \langle \delta(\mathbf{x},t), \delta(\mathbf{x}',t) \rangle = \int \mathcal{D}\delta \, \mathbb{P}[\delta] \; \delta(\mathbf{x},t)\delta(\mathbf{x}',t).$$

$$\text{Homogeneidad e isotropía} \implies \xi(\mathbf{x},\mathbf{x}',t) = \xi(r).$$

En el espacio de Fourier:

$$\langle \delta(\mathbf{k})\delta^*(\mathbf{k}') \rangle \equiv (2\pi)^3 \, \delta_{\mathrm{D}}(\mathbf{k}-\mathbf{k}') \, \mathrm{P}(k),$$

donde $\mathrm{P}(k)$ es el espectro de potencias.

En el caso que veremos adelante, $\mathrm{P}(k)$ lleva toda la información del campo aleatorio. La evolución gravitatoria de las fluctuaciones es descrita con la función de transferencia $T(k)$.

# Introducción

## Campos aleatorios gaussianos

En este tipo de campos aleatorios, la FDP es una gaussiana en cada punto del espacio. Hay motivos de peso para tomar este campo como modelo del campo de densidad primordial.

El espectro de potencias define completamente un campo aleatorio gaussiano.

Para los diferentes vectores de onda **k**:

$$\delta(\mathbf{k}) = A_{\mathbf{k}} + \mathrm{i}B_{\mathbf{k}} = |\delta(\mathbf{k})|\ \mathrm{e}^{\mathrm{i}2\pi\phi_{\mathbf{k}}}, \quad |\delta(\mathbf{k})| \sim \mathcal{N}(0, \mathrm{P}^2(k)) \ \wedge\ \phi_{\mathbf{k}} \sim U(0,1).$$

Abstract

1. Introducción

2. Objetivos

3. Materiales y métodos

4. Resultados

5. Conclusions

# Objetivos

- 👉 Algoritmo Python
- 👉 FFT y números aleatorios como herramientas principales en este algoritmo
- 👉 Modelo para espectro de potencias primordial $\mathrm{P}_0(k)$
- 👉 Modelo para espectro de potencias en época de recombinación $\mathrm{P}(k) = \mathrm{P}_0(k)T^2(k)$
- 👉 Visualizar los campos aleatorios

Abstract

1. Introducción

2. Objetivos

**3. Materiales y métodos**

4. Resultados

5. Conclusions

# Materiales y métodos

## FFT y números aleatorios gaussianos

Al simular por ordenador la transformada de Fourier se convierte en una DFT.

- ✗ DFT (definición directa) $\longrightarrow O(N^2)$
- ✓ FFT (algoritmo ingenioso) $\longrightarrow O(N \log_2 N)$

# Materiales y métodos

## FFT y números aleatorios gaussianos

Al simular por ordenador la transformada de Fourier se convierte en una DFT.

✗ DFT (definición directa) $\longrightarrow O(N^2)$

✓ FFT (algoritmo ingenioso) $\longrightarrow O(N \log_2 N)$

El campo aleatorio gaussiano es computado mediante el uso de número aleatorios bajo una estadística normal (o gaussiana) con media nula y desviación típica $\mathrm{P}(k)$.

# Materiales y métodos

## Método (librerías)

```
"""
Módulo Python del TFG Generación de campos gaussianos en teorías inflacionarias.
"""
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams.update({
        'axes.autolimit_mode': 'data',
        'axes.xmargin': 0,
        'axes.ymargin': 0,
        'xtick.direction': 'in',
        'xtick.top': True,
        'ytick.direction' : 'in',
        'ytick.right': True,
        'text.usetex': True})
from matplotlib.colors import ListedColormap, CenteredNorm
from powerbox.tools import _magnitude_grid as _mgrid
from powerbox.powerbox import _make_hermitian as _hermitian
from powerbox.dft import fftfreq as _fftfreq, ifft as _ifftn
```

# Materiales y métodos

## Método (inicialización de la clase)

```python
class FS2029FSC():
    """
    Calcula campos gaussianos en espacio real y de fourier generados con un espectro de potencia dado.
    """
    def __init__(self, amplitude, power, size, dimensions, show, method,
                boxlength=1.0, normalise=True, a=1., b=1.):

        self._amplitude = amplitude
        self._boxlength = boxlength
        self._normalise = normalise
        self._coef_a = a
        self._coef_b = b
        self._power = int(power)
        self._size = int(np.power(2, size))
        self._dimensions = dimensions
        self._show = show
        self._method = method
        self._volume = self._boxlength ** self._dimensions
        self._dx = float(boxlength) / self._size
        self._n = self._size + 1
```

# Materiales y métodos

## Método (generador)

```python
def generator(self):
    """
    Genera el campo de fluctuaciones de densidad de energía δ(x).
    """
    rng = np.random.default_rng(seed=42)
    seed_gaussiano_mag = rng.normal(0.0,1.0, size=[self._n] * self._dimensions)
    seed_gaussiano_phase = 2 * np.pi * rng.uniform(size=[self._n] * self._dimensions)
    seed_gaussiano_hermitian = _hermitian(seed_gaussiano_mag, seed_gaussiano_phase)
    cutindex = (slice(None, -1),) * self._dimensions
    seed_gaussiano_hermitian = seed_gaussiano_hermitian[cutindex]

    self._vectork = _fftfreq(self._size, d=self._dx, b=self._coef_b)
    normks = _mgrid(self._vectork, self._dimensions)
    mask = normks != 0
    self._normks = normks
    self._normks[mask], self._name = self._method(self._amplitude, self._normks[mask], self._power)

    if self._normalise:
        p_spectro = self._normks / self._volume
    else:
        p_spectro = self._normks
```

# Materiales y métodos

## Método (generador)

```
p_spectro_root = np.sqrt(p_spectro)
self._k_realize = seed_gaussiano_hermitian * p_spectro_root
self._x_realize = np.empty((self._size,) * self._dimensions,
dtype='complex128')
self._x_realize[...] = self._k_realize
self._x_realize[...] = self._volume * _ifftn(self._x_realize,
L=self._boxlength, a=self._coef_a, b=self._coef_b)[0]
self._x_realize = np.real(self._x_realize)
np.clip(self._x_realize, -1, np.inf, self._x_realize)

return self._x_realize
```

# Materiales y métodos

## Método (espectro primordial)

```python
def k_pol(amplitude, modulus, power):
    """
    Espectro de potencias primordial.
    """
    name = "pdek"
    return amplitude * (modulus/k0) ** power, name
```

# Materiales y métodos

## Método (espectro en época de recombinación)

```
def transferfunction_k(amplitude, modulus, power):
    """
    espectro de potencias en época de recombinación.
    """
    q = modulus/0.142
    L0 = np.log(2 * np.exp(1) + 1.8 * q)
    C0 = 14.2 + 731 * np.power(1 + 62.5 * q, -1)
    T0 = L0 * np.power(L0 + C0 * np.power(q, 2), -1)
    name = "transfer"
    return amplitude * np.power(q, 1 * power) * np.power(T0, 2), name
```

Abstract

1. Introducción

2. Objetivos

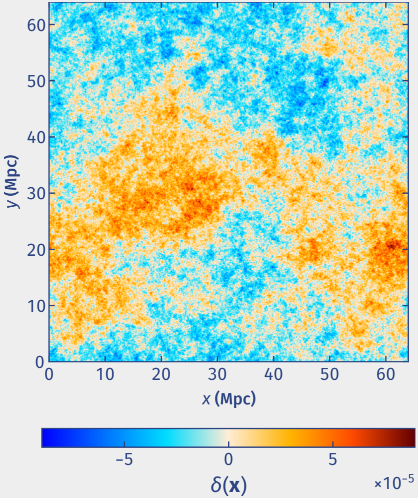
3. Materiales y métodos

**4. Resultados**

5. Conclusions

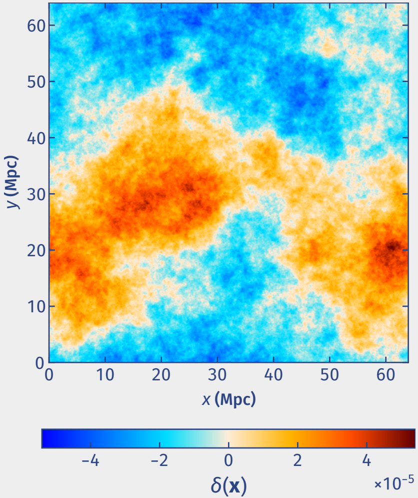
# Resultados

## Campo primordial



# Resultados

## Campo en época de recombinación



# Resultados

## Campo tridimensional

Abstract

1. Introducción

2. Objetivos

3. Materiales y métodos

4. Resultados

**5. Conclusions**

# Conclusions

→ Python 💙 Computational cosmology

→ FFT and random numbers are essential in this simulation approach

→ For $P_0(k)$:

- Large fluctuations for $n_s < 0$
- Small fluctuations for $n_s > 0$
- Harrison-Zel'dovich ($n_s = 1$) homogeneous fluctuations at large scales with noticeable fluctuations at smaller scales

→ With $T(k)$ we reproduce the spatial distribution of fluctuations in the CMB

→ 2D and 3D realizations are useful for the comprehension

# Generación de campos gaussianos en teorías inflacionarias

TFG: FS20-29-FSC

16 de noviembre de 2022

Universidad de Córdoba

Rafael Jurado Ariza