



# Expansión acelerada de un universo vacío: aportes para el estudio de la relatividad general

Jaream Camilo Arias Amaya

Licenciatura en Física  
**Universidad Pedagógica Nacional**

Línea de Profundización:  
La Enseñanza de la Física y la Relación Física Matemática

Trabajo de Grado

Asesor:  
Néstor Fernando Méndez Hincapié

Abril. 2026  
Bogotá D.C., Colombia

# Índice general

|                                                                              |           |
|------------------------------------------------------------------------------|-----------|
| <b>1. Problema de Investigación</b>                                          | <b>7</b>  |
| 1.1. Objetivos . . . . .                                                     | 9         |
| 1.1.1. Objetivo General . . . . .                                            | 9         |
| 1.1.2. Objetivos específicos . . . . .                                       | 9         |
| 1.2. Antecedentes . . . . .                                                  | 9         |
| 1.3. Metodología . . . . .                                                   | 11        |
| <b>2. Marco Teórico</b>                                                      | <b>12</b> |
| 2.1. Didáctica del lenguaje Python . . . . .                                 | 12        |
| 2.2. Universo de de Sitter . . . . .                                         | 13        |
| 2.3. Parámetros físicos . . . . .                                            | 14        |
| 2.3.1. Constante y parámetro de Hubble . . . . .                             | 14        |
| 2.3.2. Factor de escala . . . . .                                            | 16        |
| 2.4. Cálculo tensorial . . . . .                                             | 26        |
| 2.4.1. Tensor métrico . . . . .                                              | 26        |
| 2.4.2. Métrica de Friedmann-Lemaître-Robertson-Walker . . . . .              | 27        |
| 2.4.3. Métrica en de Sitter . . . . .                                        | 28        |
| 2.4.4. Símbolos de Christoffel . . . . .                                     | 32        |
| 2.4.5. Conexión Levi-Civita . . . . .                                        | 36        |
| 2.4.6. Tensor de Riemann . . . . .                                           | 37        |
| 2.4.7. Tensor de Ricci . . . . .                                             | 37        |
| 2.4.8. Tensor Energía-Momento . . . . .                                      | 39        |
| 2.5. Métrica de de Sitter . . . . .                                          | 40        |
| <b>3. Implicaciones físicas: Expansión acelerada y constante cosmológica</b> | <b>42</b> |
| 3.1. Homogeneidad . . . . .                                                  | 42        |
| 3.2. Isotropía . . . . .                                                     | 47        |
| 3.3. Estructura causal en de Sitter . . . . .                                | 47        |
| 3.3.1. Horizonte cosmológico . . . . .                                       | 49        |
| 3.4. Constante cosmológica . . . . .                                         | 52        |
| 3.5. Distancia entre partículas . . . . .                                    | 59        |
| 3.5.1. Distancia física . . . . .                                            | 59        |
| 3.5.2. Distancia comóvil . . . . .                                           | 59        |
| 3.6. Simulación constante de Hubble . . . . .                                | 65        |
| <b>Conclusiones</b>                                                          | <b>66</b> |

|                                               |    |
|-----------------------------------------------|----|
| Apéndices                                     | 70 |
| A. Código constante de Hubble                 | 71 |
| B. Solución ecuación diferencial 2.11         | 73 |
| C. Solución de los factores para obtener 2.17 | 74 |
| D. Solución de los factores para obtener 2.23 | 75 |
| E. Código distancia física                    | 76 |
| F. Código de geometrías del espacio $k$       | 78 |
| G. Escalar de Ricci                           | 81 |

*A mi mamá, por ser el pilar fundamental de mi vida y el ejemplo constante de esfuerzo, amor y perseverancia. Su apoyo incondicional ha sido la fuerza que me ha permitido superar cada desafío y llegar hasta este momento tan importante. A ella le dedico este trabajo, con profundo agradecimiento y amor.*

# Agradecimientos

Le quiero agradecer a mi mamá, por ser el corazón de todo lo que soy. Su amor incondicional, su entrega y su fe en mí han sido la luz que ha guiado cada uno de mis pasos. Este logro es, en gran parte, reflejo de todo lo que me ha dado.

A mi profesor asesor, por creer en mí, por su paciencia y dedicación, y por acompañarme con generosidad en este camino académico. Su guía ha sido fundamental para llegar hasta aquí.

Le agradezco a mi pareja, por ser refugio y fuerza en los momentos de duda, por su comprensión, su compañía y su amor constante. Gracias por caminar a mi lado incluso en los días más difíciles.

# Introducción

En el desarrollo de este trabajo se tendrá en cuenta la importancia de los aspectos avanzados del formalismo matemático y su relación con los fenómenos físicos. Seguidamente se hará el desarrollo matemático y geométrico de la métrica de de Sitter paso a paso, señalando el significado físico-matemático del proceso, respaldado complementariamente, con representaciones gráficas y geométricas ya que, en palabras de Creswell (2013a), citado en (Hernández et al., 2014) Para entender un problema de las ciencias se utilizan datos numéricos, textuales, gráficos y simbólicos. Posteriormente se hará la traducción del proceso en lenguaje Python con la intención de señalar las ventajas de las representaciones computacionales en el estudio de dicha solución.

# Capítulo 1

## Problema de Investigación

La enseñanza de la relatividad general y la cosmología representa uno de los mayores desafíos pedagógicos en la formación de estudiantes de física y áreas afín. Teniendo en cuenta que estos temas requieren un dominio riguroso del lenguaje matemático y una capacidad de abstracción elevada para interpretar modelos físicos, cuya evidencia empírica no es directa o intuitiva, esto termina en una falta de comprensión, tanto del concepto físico, como del modelo matemático, de tal forma que se dificulta la elaboración de una visión del mundo físico (Ayala, 2007).

En este contexto, conceptos como la métrica del espacio-tiempo, la expansión acelerada del universo, la constante cosmológica y las soluciones exactas de las ecuaciones de Einstein (como la solución de de Sitter) pueden ser una base para muchos estudiantes interesados en esta área, ya que, según Pérez y Solbes (2003), estudiar este tipo de contenidos otorga una visión más amplia en el desarrollo de la ciencia porque muestra los límites de la física clásica en los nuevos fenómenos que aparecieron con la física moderna.

La solución de de Sitter, que describe un universo vacío con constante cosmológica positiva, es fundamental para entender el modelo inflacionario y la aceleración cósmica observada actualmente. No obstante, el tratamiento matemático para las métricas que aparecen en la solución es extenso y, por lo tanto, propenso al error humano como afirman Castillo et al. (2025). Es así que el tratamiento de esta temática en un curso de física podría tener un acompañamiento visual o práctico, que ayudan a reforzar la conexión entre las expresiones matemáticas y su interpretación física.

Frente a esta situación, se plantea la posibilidad de incorporar herramientas computacionales como Python en la enseñanza de estos temas. La implementación de la solución de de Sitter mediante programación permite simular y visualizar la evolución de la métrica, la dinámica del espacio-tiempo

y otros aspectos relevantes del modelo. Esta aproximación no solo favorece la comprensión conceptual a través de la experimentación virtual, sino que también promueve el desarrollo de competencias digitales y el pensamiento computacional, elementos cada vez más valorados en la formación científica contemporánea.

Se busca resolver la ecuación de campo de Albert Einstein en el modelo propuesto por W. de Sitter en la cual el tensor energía masa es igual a cero dando como resultado la siguiente ecuación.

$$R_{\mu\nu} - \frac{1}{2}g_{\mu\nu}R + \Lambda g_{\mu\nu} = 0 \quad (1.1)$$

La solución de Willem de Sitter constituye un modelo geométrico dominado por una constante cosmológica positiva. En la actualidad, esta constante se interpreta como una manifestación de la energía oscura, componente que representa cerca del 70 % del contenido energético del cosmos y que es responsable de la expansión acelerada observada (Peebles & Ratra, 2003). Asimismo, la solución de Sitter ofrece un marco de referencia para comparar modelos alternativos, como las teorías de la gravedad modificada.

Además de ser una solución exacta de la relatividad general, teoría que según Kersting et al. (2020) de por sí es abstracta y conceptualmente densa, la solución de de Sitter involucra temáticas avanzadas que exceden el marco conceptual de la física que se enseña en la Educación Media o inicial de pregrado. Por ejemplo, se rompe con la idea de un espacio euclidiano, no se tienen referentes sobre la energía del vacío, ni la expansión del universo, mucho menos de la expansión de un universo vacío, y esto, según Henriksen et al. (2014), resulta contraintuitivo para los estudiantes. Por consiguiente, se precisa de una traducción de los conceptos geométricos hacia representaciones que no distorsionen el significado científico, lo que suscita en el requerimiento de una mediación didáctica apoyada por las herramientas computacionales y visuales.

### **pregunta problema**

¿Cómo puede la solución de de Sitter, derivada de las ecuaciones de campo de Einstein con constante cosmológica positiva, ser representada y analizada mediante simulaciones computacionales que permitan visualizar la relación entre la constante cosmológica y la expansión del universo con fines didácticos?

## **1.1. Objetivos**

### **1.1.1. Objetivo General**

Analizar la solución de de Sitter como modelo cosmológico dentro del marco de la relatividad general, a partir de su formulación matemática, sus implicaciones físicas y su representación mediante simulaciones computacionales, con el fin de reconocer su relevancia en la descripción de la expansión acelerada del universo.

### **1.1.2. Objetivos específicos**

1. Describir la métrica y las propiedades fundamentales de la solución de de Sitter dentro del marco teórico de la relatividad general.
2. Examinar las implicaciones físicas de la solución de de Sitter en el contexto de la expansión acelerada del universo y la constante cosmológica.
3. Desarrollar varias simulaciones computacionales que modelen la solución de de Sitter que muestren el compromiso entre la constante cosmológica y la expansión del universo.
4. Analizar las ventajas y limitaciones del empleo de herramientas computacionales en el estudio y análisis de soluciones exactas de las ecuaciones de Einstein.

## **1.2. Antecedentes**

La problemática de enseñar relatividad general en secundaria superior ha sido abordada por Kersting et al. (2018) Quienes implementaron un entorno educativo en línea a estudiantes de secundaria superior. El material utilizado se basó en la perspectiva sociocultural del aprendizaje con enfoque histórico-filosófico. En esta investigación los autores concluyeron que los estudiantes de estos niveles pueden alcanzar una comprensión adecuada de la relatividad general si se les proporcionan las herramientas pedagógicas apropiadas que fomenten la comprensión cualitativa.

También se destacan los trabajos de sector models, que es una propuesta de modelos por sectores para introducir las nociones de curvatura, geodésica de la luz, espacio-tiempo, corrimiento al rojo y caída libre, con la matemática

básica (aproximación de una superficie curva mediante triángulos planos). Estos trabajos han implementado talleres con estudiantes de secundaria y universitarios (Kraus & Zahn, 2018; Zahn & Kraus, 2014).

Los anteriores trabajos muestran el compromiso con los conceptos cualitativos, geométricos y visuales que se tendrán en cuenta en esta propuesta para desarrollar las simulaciones computacionales.

Desde el campo disciplinar, el trabajo de Campo (2022) hace una breve introducción a las propiedades del álgebra lineal y de la topología necesaria para llegar a las ecuaciones de campo de Albert Einstein. A la vez que introduce las condiciones propuestas por W. de Sitter para dar solución a estas ecuaciones muestra las estructuras matemáticas que se pueden obtener con base en la Teoría Especial de la Relatividad (TER). Se desarrolla la solución de las ecuaciones de Killing para el universo de de Sitter; todo ello con procesos matemáticos extensos propios del área del cálculo tensorial que además están implícitos en los resultados obtenidos.

En el trabajo de Balaguera (2006) se hace un estudio de la estructura cosmológica y astrofísica de la constante cosmológica con un valor diferente de cero. En este trabajo se tienen en cuenta aspectos físicos como la energía oscura, la homogeneidad y la temperatura, entre otras. Además, hace el análisis de las consecuencias, el desarrollo y consideraciones necesarias para llegar dichos conceptos. Su desarrollo matemático y su implementación en la Teoría General de la Relatividad (TGR) hace énfasis en la implicación geométrica y cómo ello influye en el comportamiento y valor de la constante cosmológica, más específicamente en la geometría esférica para finalmente observar el movimiento de una partícula hipotética en la métrica de Schwarzschild y el modelo de W. de Sitter. Se destaca en este trabajo el uso de las representaciones gráficas que ilustran y dan sentido a las ecuaciones que presentan allí.

El trabajo de Sánchez (1985) hace un recorrido histórico sobre las concepciones de la física moderna y su respectivo análisis ontológico; teniendo un apartado en donde se abarcan las soluciones cosmológicas a las ecuaciones de campo de Albert Einstein, sobre todo para dar contexto de la solución de W. de Sitter, su importancia y las consecuencias que trajo consigo en la comunidad científica del momento.

El trabajo de Niels y Saavedra (2020) se realiza un código en lenguaje Python 3 como herramienta complementaria para estudiar la solución Friedmann-Lemaître-Robertson-Walker (FLRW) de las ecuaciones de cam-

po, empezando con la descripción general de los aspectos que considera la solución, de los cuales hay algunos compatibles con la solución de W. de Sitter y por consiguiente útiles en el trabajo, como lo son la suposición de un universo homogéneo e isótropo, además la solución de Sitter puede escribirse en términos de la métrica FRLW. En el trabajo hay una guía con los códigos que describen cada elemento necesario para describir dicha métrica.

### 1.3. Metodología

Se propone una metodología de enfoque mixto entre análisis cualitativo, modelación matemática y simulación computacional, dado que el estudio busca integrar el análisis conceptual y formal de la solución de W. de Sitter con representaciones computacionales que faciliten su análisis. Esta metodología, en palabras de Hernández et al. (2014), representa un conjunto de procesos de recolección y análisis de datos cuantitativos y cualitativos, así como su integración y discusión conjunta, para lograr un mayor entendimiento del fenómeno bajo estudio.

Este trabajo busca recopilar información de las limitaciones y alcances que trajo consigo la solución; desde el enfoque cualitativo, se analizarán textos históricos y científicos relacionados con la solución de de Sitter, mientras que el enfoque teórico-computacional de carácter exploratorio y explicativo se centrará en el desarrollo matemático de la métrica, así como en la representación computacional y visualización gráfica a través de Python.

Se dividirá en trabajo en cinco fases, siendo la primera la fase introductoria abordar la solución de W. de Sitter, en donde se hará búsqueda del contexto histórico. En la segunda fase se hará la formulación matemática y desarrollo de la solución partiendo de las ecuaciones de campo de Albert Einstein y señalando las consideraciones de Willem de Sitter para llegar a su métrica. En la tercera fase se analiza la relación entre la constante cosmológica y la expansión acelerada del universo. La cuarta fase se centrará en la implementación computacional con Python del desarrollo matemático, respaldado con las representaciones gráficas de las consideraciones y de la solución misma. Para última fase se hace la valoración de las ventajas y limitaciones de la implementación de las herramientas digitales en el estudio de la relatividad general.

# Capítulo 2

## Marco Teórico

### 2.1. Didáctica del lenguaje Python

La didáctica de la física crea relaciones entre cómo se enseña y qué se enseña en distintos campos del conocimiento. Además, teniendo en cuenta que la didáctica es una función de la enseñanza y debería diversificar los contextos de aprendizaje (Klein, 2012), la formación del futuro docente debe orientarse a desarrollar una visión que integre el conocimiento desde la interdisciplinariedad. De manera tal que, pueda enfrentar y resolver con mayor eficacia los problemas propios de la enseñanza de la Física (Castiblanco & Nardi, 2023).

Por otra parte, se hace necesario darle sentido al área de conocimiento del que el profesorado enseña. Los profesores de Física además de conocer disciplinas educativas, evidentemente también se desenvuelven en esa rama de la ciencia, y en palabras de Klein (2012) a menudo da la impresión de que el profesor de Física en realidad está impartiendo Matemáticas.

Entonces, teniendo en cuenta que la evolución de las tendencias en enseñanza según Castiblanco y Nardi (2023, pág. 52-53) abarcan desde la enseñanza por transmisión hasta la enseñanza por investigación, y que en ese tránsito, sobresale el uso de las tecnologías y de cómo han ido tomando papeles importantes en la didáctica de la Física, se le suma importancia a comprender el verdadero alcance de las tecnologías para enriquecer las interacciones en el aula y mejorar la enseñanza y el aprendizaje de la Física (Castiblanco & Nardi, 2023).

Ahora bien, la relevancia del uso de herramientas digitales en la didáctica de la física se torna un tema de discusión que no puede pasar por alto. Castiblanco y Nardi (2023) hacen una caracterización de disciplinas que pueden tener en el proceso de formación del licenciado para enseñar la física desde diferentes formas. Uno de ellos hace énfasis del conocimiento que enriquece la

interacción en el aula, basado en el conocimiento de los usos de las tecnologías. El uso de herramientas digitales en la educación es una de las corrientes más fuertes y comunes en la actualidad (Torres Cañizález & Cobo Beltrán, 2017, pág 32).

Por ello, se considera beneficioso el uso de herramientas tecnológicas y digitales como el lenguaje Python como una representación de la didáctica de la física; además de que este, en palabras de Castillo et al. (2025) es versátil y asimismo aceptado por la comunidad científica.

## 2.2. Universo de de Sitter

Einstein defendía la idea de un universo estático y que asimismo su estructura métrica dependía plenamente de la materia, aparentemente influenciado por el principio de Mach (Janssen, 2013). Todo lo contrario a la solución que W. de Sitter propuso y que además se posicionó como una de las soluciones exactas de las ecuaciones de campo de la teoría general de la relatividad (TGR). De Sitter, considerando una constante cosmológica positiva,  $\Lambda > 0$ , descubrió un posible universo vacío y en expansión acelerada, lo cual trajo consigo críticas por parte de la comunidad científica del momento, primero por la asunción indiscutible de esos tiempos, y es que sin materia no habría dinámica y, por otro lado, se le criticó la singularidad de coordenadas que presenta la ecuación.

No obstante, gracias a la solución, que hoy en día es considerada físicamente muy relevante, se pudieron estudiar los dos modelos cosmológicos actuales como la inflación y la aceleración cósmica, además de ser el modelo fundamental para estudiar teorías de campo cuánticas en expansión, los efectos de Hawking-Gibbons y también el estudio del modelo cosmológico actual, ya que este se asemeja al modelo de de Sitter por la consideración de la energía oscura (Balaguera, 2006). Al ser este un modelo altamente simétrico lo hace muy útil en estudios teóricos que requieren de condiciones ideales y calculables.

Su simplicidad estructural permite introducir a los/as estudiantes a conceptos como la expansión acelerada, curvatura del espacio-tiempo, horizonte de partículas y efectos físicos de la constante cosmológica. En este trabajo se pretende hacer un recorrido sobre las implicaciones física e históricas, procesos matemáticos, y presuntas consideraciones que tuvo W. de Sitter a la hora de solucionar las ecuaciones de campo de Albert Einstein para llegar a su solución; todo esto seguido de modelaciones en Python como recurso com-

plementario que de otro punto de vista de la rigurosidad matemática que allí se maneja. Las ecuaciones de campo para el universo de de Sitter están dadas por la expresión

## 2.3. Parámetros físicos

### 2.3.1. Constante y parámetro de Hubble

Existe una correlación entre las distancias de los objetos y el conocido corrimiento al rojo como lo menciona Weinberg (1972). Este corrimiento es un fenómeno observable que sugiere un universo en expansión que, en términos generales, se define cuando una longitud de onda observada para una misma línea de absorción en el espectro de una galaxia lejana, difiere de la original (Coles & Lucchin, 2002). De hecho fue este uno de los fenómenos que hizo que Einstein modificara sus ecuaciones de campo. Esta correlación está ligada al factor de escala  $a(t)$  como una serie de potencias ya que es una función suave (que es infinitamente derivable), dado a las condiciones físicas regulares dadas por la ecuación de campo de Einstein 1.1 (Weinberg, 2008, p. 441).

$$a(t) = a(t_0)[(1 + H_0(t - t_0) - \frac{1}{2}q_0H_0^2(t - t_0)^2 \dots)] \quad (2.1)$$

donde  $H_0$  y  $q_0$  son la constante de Hubble y el parámetro de desaceleración respectivamente; resultados que se obtienen por inspección del desarrollo de serie de potencias 2.1.

$$H_0 = \frac{\dot{a}(t_0)}{a(t_0)} \quad (2.2)$$

$$q_0 = -\ddot{a}(t_0) \frac{a(t_0)}{\dot{a}(t_0)^2} \quad (2.3)$$

Hubble seguiría intentando calcular el valor de la constante que lleva su nombre,  $H_0$ ; él encontró una relación lineal entre la distancia de la galaxia y la velocidad de recesión, o sea la velocidad con la que los objetos celestes lejanos se separan de la Tierra como consecuencia de la expansión del universo. (Weinberg, 1972).

Posteriormente, en 1960 Minkowski descubrió un objeto con un corrimiento al rojo de 0.47 y otros estudios más dieron cuenta que la magnitud de la constante de Hubble está dada por 2.4 (Weinberg, 2008, pág 11).

$$cz = H_0 r \quad (2.4)$$

donde  $c$  es la velocidad de la luz en el vacío,  $z$  es el corrimiento al rojo y  $r$  es la distancia medida desde el observador hasta el objeto.

Este valor de  $H_0$  da cuenta de la tasa fraccional de cambio de tamaño del universo por unidad de tiempo. Es decir, qué tan rápido se expande este. Si se invierte este valor, se vuelve atrás en el tiempo y se puede llegar a conocer la edad de nuestro universo. El valor de  $H_0 = (71 \pm 4)$  km/s/Mpc (Serrano González et al., 2019, p. 16). Esto significa que la edad del Universo es  $4,35 \times 10^{17}$  s, aproximadamente 13800 millones de años.

Se construyó el código que aparece en la Figura 2.1 para visualizar la implicación del valor de la constante. En conjunto, el código hace una simulación sencilla de la Ley de Hubble generando distancias, calculando velocidades siguiendo una relación lineal con ruido observacional, ajusta una recta por mínimos cuadrados y recupera una estimación del parámetro de Hubble a partir de los datos. Es una manera computacional de reproducir, en versión simplificada, el procedimiento que permitió establecer la expansión del universo.

```
import numpy as np
import matplotlib.pyplot as plt

# Parámetro de Hubble que puede variar
H0_real=float(input("Ingrese el valor de H0 en km/s/Mpc:"))

# Distancias simuladas de galaxias (Mpc) (3.96 millones de años luz)
distancias = np.linspace(10, 300, 50)

# Velocidades simuladas con un poco de ruido (km/s)
np.random.seed(0)
velocidades = H0_real * distancias + np.random.normal(0, 300, len(distancias))

# Ajuste lineal para encontrar H0
coeficientes = np.polyfit(distancias, velocidades, 1)
H0_ajustado = coeficientes[0]

# Visualización
plt.scatter(distancias, velocidades, label="Datos simulados")
plt.plot(distancias, H0_ajustado*distancias,
         color='r', label=f"Ajuste: H0 = {H0_ajustado:.1f} km/s/Mpc")
plt.xlabel("Distancia (Mpc)")
plt.ylabel("Velocidad de recesión (km/s)")
plt.title("Simulación de la Ley de Hubble con Python")
plt.legend()
plt.grid(True)
plt.show()
```

Figura 2.1: Código de constante de Hubble. Elaboración propia

La gráfica de la Figura 2.2 muestra la relación lineal  $v = H_0 d$ . A esta

relación ideal se le añade un término de ruido aleatorio que hace que los puntos no caigan exactamente sobre una recta perfecta. Esto es un punto importante, porque después el código utiliza un ajuste lineal mediante la regresión para encontrar la mejor recta que describen esos datos, obteniendo así un valor estimado de  $H_0$ . Esto se ve como una nube de puntos dispersos y una línea roja que representa el ajuste lineal.

Ingrese el valor de  $H_0$  en km/s/Mpc:30

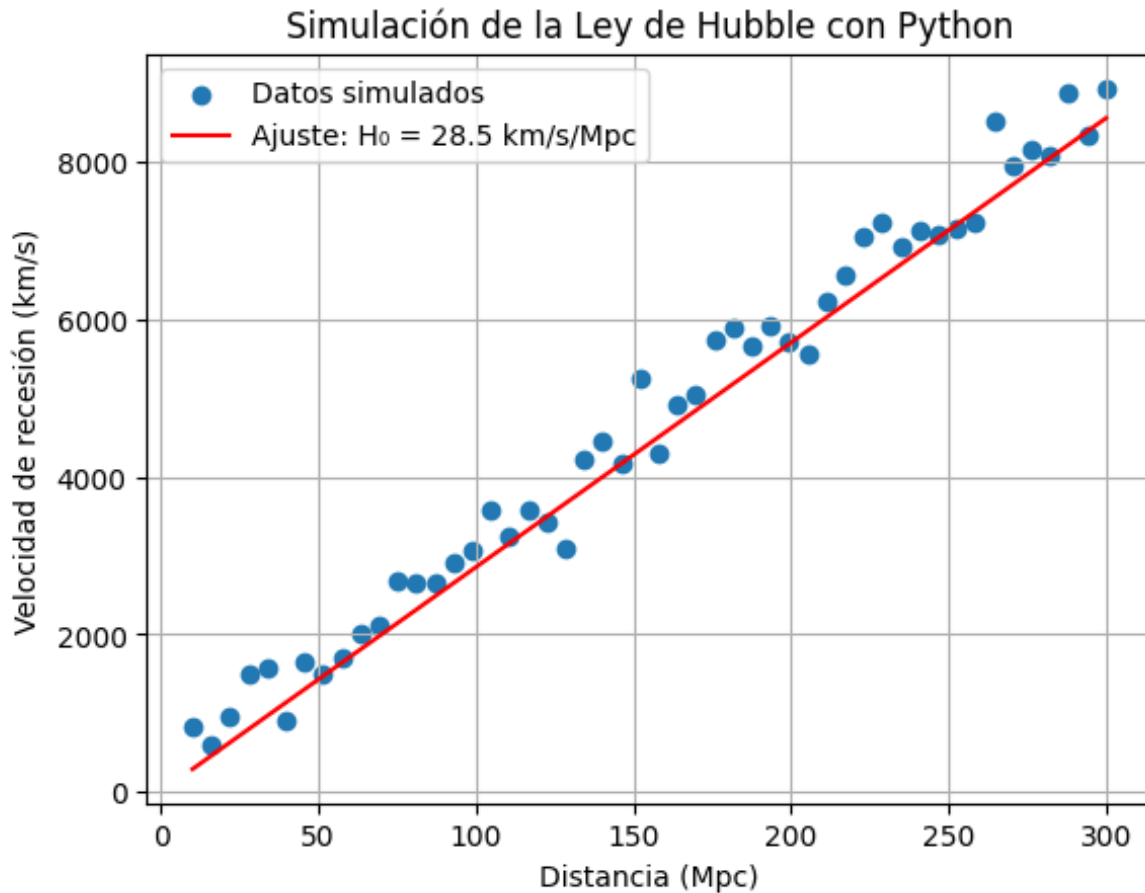


Figura 2.2: Gráfica constante de Hubble. Elaboración propia

### 2.3.2. Factor de escala

El factor de escala aparece en el momento que se le atribuyen las propiedades de homogeneidad e isotropía al universo de de Sitter. Ya que este es una función dependiente del tiempo  $a = a(t)$  que escala las distancias comóviles del espacio invariante. Las distancias comóviles son aquellos puntos en el espacio que no cambian con el tiempo y, en este caso, no se mueven con la expansión del mismo. Un ejemplo de distancias comóviles típica es suponer

un globo al que se le pegan dos monedas y este se infla, la ubicación comóvil de la moneda es la misma aunque el globo se infle, no obstante la distancia entre las dos monedas va a cambiar ya que el globo se está expandiendo, esa es la distancia física.

Estas coordenadas se transportan con la expansión dado que, la expansión es uniforme (Liddle, 2015). Entonces si se tienen dos cuerpos comóviles separados por una distancia  $x$  que es constante en el tiempo, ya que la propiedad de homogeneidad garantiza que  $a(t)$  sea una función únicamente del tiempo, su distancia física estará dada por.

$$d(t) = xa(t) \quad (2.5)$$

Por otro lado, de las ecuaciones de Friedmann se infiere que la densidad de energía total del universo  $\rho = \rho_r + \rho_m$ , en donde  $\rho_r$  es la densidad de radiación,  $\rho_m$  es la densidad de materia; a esta densidad se le agrega el término  $\rho_\Lambda$  que es la densidad de energía del vacío, siendo esta la única diferente de cero según el universo de de Sitter (Kenyon, 2023, pág 10-6).

$$\left(\frac{\dot{a}}{a}\right)^2 = \frac{8\pi G(\rho + \rho_\Lambda)}{3} - \frac{kc^2}{a^2} \quad (2.6)$$

En donde  $G$  es la constante de gravitación universal y  $k$  es la geometría del espacio. Primero se resuelve la ecuación diferencial que aparece en las ecuaciones de Friedmann 2.6 implementando las condiciones de la solución de W. de Sitter; es decir  $\rho = 0$  y  $\rho_\Lambda \neq 0$  con  $\Lambda > 0$  teniendo en cuenta las siguientes definiciones.

$$\rho_\Lambda = \frac{\Lambda c^2}{8\pi G} \quad (2.7)$$

$$H = \left(\frac{\dot{a}}{a}\right) \quad (2.8)$$

De la condición 2.8 se puede observar que, si  $a$  aumenta es porque el universo se está expandiendo, si decrece es porque se está contrayendo y si su valor es  $a = 0$  el universo es estático, ya que la distancia física entre dos cuerpos no cambia (Liddle, 2015, pág 42).

### Valores del factor de escala

Partiendo de las ecuaciones de Friedmann 2.6 y aplicando las igualdades 2.7 y 2.8 se obtienen las ecuaciones con la cuales se desarrollará este trabajo.

Estas dependen de la constante cosmológica, lo que resulta relevante para el análisis que aquí se presenta, y se pueden expresar como 2.9 y 2.10 donde  $a = a(t)$  (Liddle, 2015).

$$\left(\frac{\dot{a}}{a}\right)^2 + \frac{kc^2}{a^2} = \frac{\Lambda c^2}{3} = H^2 \quad (2.9)$$

$$\frac{\ddot{a}}{a} = \frac{\Lambda c^2}{3} = H^2 \quad (2.10)$$

Con 2.10 se obtiene una ecuación diferencial de segundo orden 2.11.

$$\ddot{a} - aH^2 = 0 \quad (2.11)$$

De la cual su solución general está dada por.

$$a(t) = Ae^{Ht} + Be^{-Ht} \quad (2.12)$$

Donde  $A$  y  $B$  son constantes de integración cuya solución de la ecuación diferencial está en el apéndice B. Haciendo uso de las expresiones 2.9 y 2.10 se puede obtener una relación en donde el producto de esas constantes dependen del parámetro de curvatura espacial  $k$ . Esto es importante ya que la expresión del factor de escala varía dependiendo de la geometría del espacio.

$$AB = \frac{kc^2}{4H^2} \quad (2.13)$$

- $k = 1$  El universo es esférico
- $k = 0$  El universo es plano
- $k = -1$  El universo es hiperbólico

Para visualizar esto, se desarrolló el código 2.3. Cabe señalar que los ejes de la simulación no poseen unidades debido a que las coordenadas representadas corresponden a variables adimensionales. En este contexto, el factor de escala actúa como un multiplicador relativo que describe la expansión del espacio sin fijar una escala métrica absoluta. Por tanto, la simulación enfatiza los aspectos geométricos de la expansión cosmológica más que una representación cuantitativa en unidades físicas como los megapársecs que aparecen en las unidades de la constante de Hubble.

```

import numpy as np
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation
from IPython.display import HTML

print("k=1 geometría esférica")
print("k=0 geometría plana")
print("k=-1 geometría hiperbólica")
# Elegir geometría
k = int(input("Elija la geometría del espacio k (1, 0, -1): "))

# Parámetros
c=3e5 #km/s
H_km_s_Mpc = 70 # km/s/Mpc
A=1 #En el universo de de Sitter

# conversiones
Mpc_to_km = 3.086e19
year_to_s = 3.154e7

# convertir H a 1/año
H = (H_km_s_Mpc / Mpc_to_km) * year_to_s

# Tiempo en años
t = np.linspace(0, 5e9, 60) # 5 mil millones de años

fig = plt.figure(figsize=(6,6))
ax = fig.add_subplot(111, projection='3d')

```

```

#factor de escala
def scale_factor(t):

    if k == 1:
        return (c / H) * np.cosh(H * (t))

    elif k == -1:
        return (c / H) * np.sinh(H * (t))

    elif k == 0:
        return A * np.exp(H * (t))
# Malla
u = np.linspace(0, 2*np.pi, 36)

if k == -1:
    v = np.linspace(0, 2, 36)
else:
    v = np.linspace(0, np.pi, 36)

u, v = np.meshgrid(u, v)

# Superficie
def get_surface(a):

    if k == 1:
        x = a * np.sin(v)*np.cos(u)
        y = a * np.sin(v)*np.sin(u)
        z = a * np.cos(v)

    elif k == 0:
        x = a * (np.cos(u))*np.sin(v)
        y = a * (np.sin(v))*np.sin(u)
        z = 0 * x

```

```

elif k == -1:
    x = a * np.sinh(v)*np.cos(u)
    y = a * np.sinh(v)*np.sin(u)
    z = a * np.cosh(v)

    return x, y, z

# Animación
def update(frame):

    ax.clear()

    tiempo = t[frame]

    a=np.exp(H*tiempo)

    #a = scale_factor(tiempo)

    x, y, z = get_surface(a)

    ax.plot_surface(x, y, z, alpha=0.6)

    ax.set_title(f"k={k} | t={tiempo/1e9:.2f} Gyr | a(t)={a:.2f}")
    #lim=np.max(np.abs(x))
    #ax.set_xlim(-lim,lim)
    #x.set_ylim(-lim,lim)
    #ax.set_zlim(-lim,lim)
    ax.set_xlim(-4,4)
    ax.set_ylim(-4,4)
    ax.set_zlim(-4,4)

if k == 1: print("Haz elegido la geometría esférica")
elif k == 0: print("Haz elegido la geometría plana")
elif k == -1: print("Haz elegido la geometría hiperbólica")
anim = FuncAnimation(fig, update, frames=len(t), interval=100)
HTML(anim.to_jshtml())

```

Figura 2.3: Código geometrías del espacio. Elaboración propia

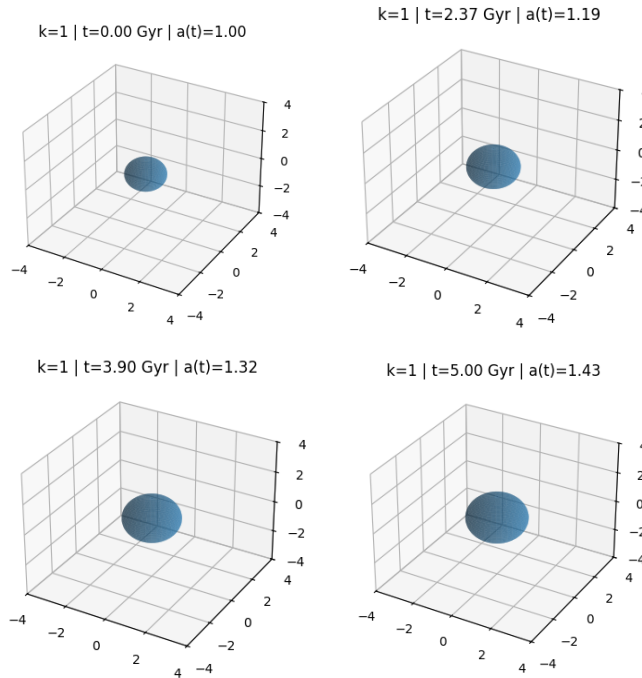
Nótese que en el código en el apartado de la animación se encuentran unas líneas a modo de comentario; una es en la expresión del factor de escala y

la otra es la parte de los ejes del cuadro, estas permiten dar cuenta de la expansión de una manera más “real” de cierto modo. De nuevo es necesario subrayar que el código muestra una representación de la geometría del espacio y no de su creación. Es decir, las gráficas allí representadas no significan que hay un espacio ya delimitado y la expansión crea espacio dentro del espacio sino que es una mera forma de visualizar cómo evoluciona la geometría.

Este código está adaptado para efectos prácticos para graficar las distintas geometrías del espacio que se verán a continuación, cuya explicación se encuentra en el apéndice F.

### **k=1**

Por su parte, la geometría esférica  $k=1$  conduce a una representación en la cual el factor de escala está dado por 2.14. En este caso, aparece de manera explícita una escala característica  $\frac{c}{H}$ , que puede interpretarse como el radio de curvatura del espacio-tiempo porque representa la escala máxima sobre la cual el universo puede considerarse “localmente plano” antes de que los efectos de la expansión y la geometría cosmológica se vuelvan importantes (Weinberg, 1972).



Cuadro 2.1: Geometría del espacio plano  $k=1$ . Elaboración propia

Esta formulación es particularmente adecuada para analizar la estructura

global del universo de de Sitter, ya que describe un espacio cerrado, finito y sin borde, que alcanza un tamaño mínimo en  $t=0$  y se expande simétricamente hacia el pasado y el futuro. La expresión del factor de escala para un universo esférico puede escribirse de la siguiente forma.

$$a(t) = \frac{c}{H} \cosh (H(t - t_0)) \quad (2.14)$$

Se obtiene una expansión de un universo cerrada, como se aprecia en las figuras del Cuadro 2.1

**k=0**

En el caso  $k=0$ , la métrica adopta una forma en la que el factor de escala evoluciona exponencialmente como aparece en 2.15 que va a ser la expresión que más se va a usar en este trabajo. Esta representación resulta especialmente útil para describir procesos de expansión acelerada como los asociados a la inflación cosmológica, puesto que enfatiza en la relación directa entre el tiempo y el crecimiento exponencial del espacio. Además, al no incorporar una escala espacial intrínseca, permite una normalización conveniente, como fijar  $a(0) = 1$  (Pascu, 2024), por esta razón en el código 2.3 la constante de  $A=1$ .

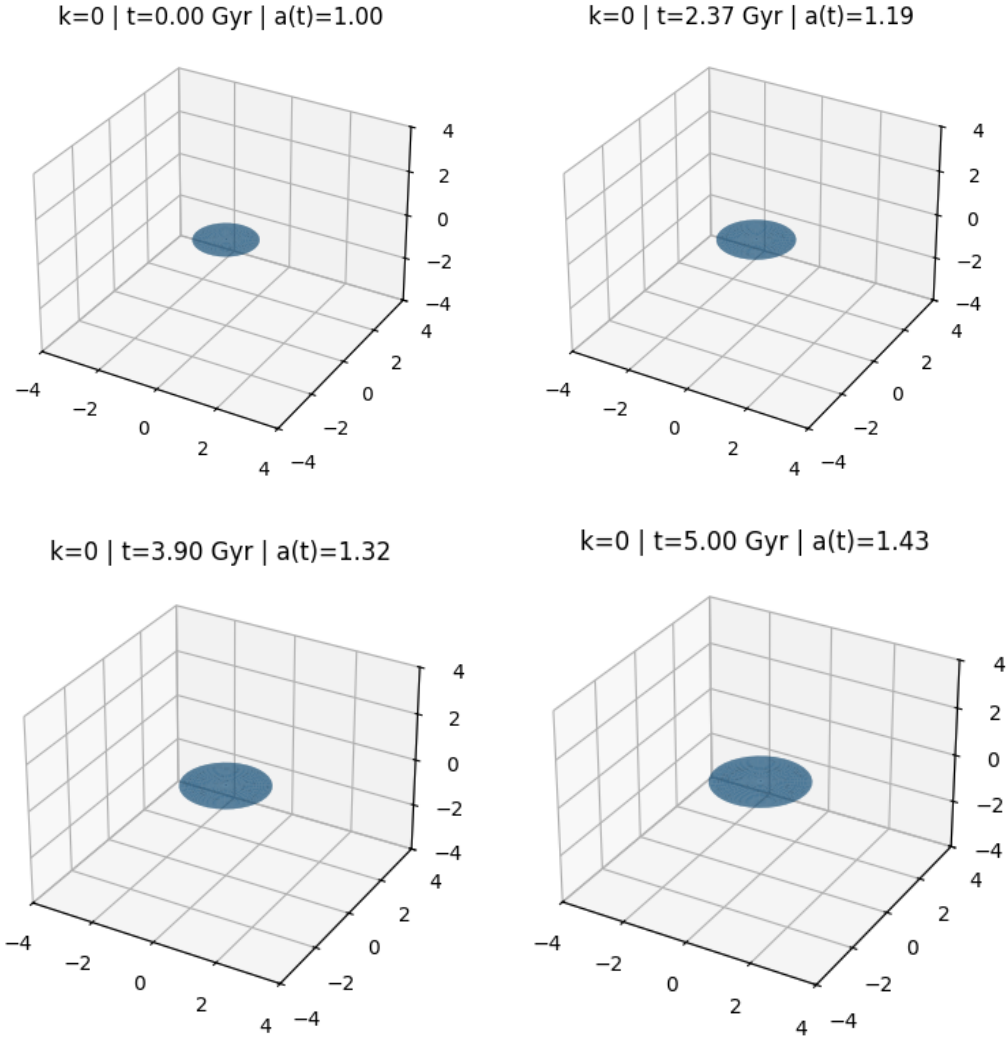
Para un universo plano el producto  $AB = 0$  y según sea el caso de estudio, se elige una de las constantes nula y la otra diferente de cero. Por ejemplo si se selecciona  $A = 0$  entonces  $B \neq 0$

$$a(t) = Be^{-H(t-t_0)} \quad (2.15)$$

Para el otro caso en donde  $A \neq 0$  y  $B = 0$

$$a(t) = Ae^{H(t-t_0)} \quad (2.16)$$

Para la expresión 2.15 este factor de escala correspondería a una contracción del espacio y para el caso 2.16 sería lo contrario, es decir, un factor de expansión. Desde el estudio del universo de de Sitter se dice que el universo está en expansión por lo tanto se trabaja con 2.16 como se muestra en el Cuadro 2.2.



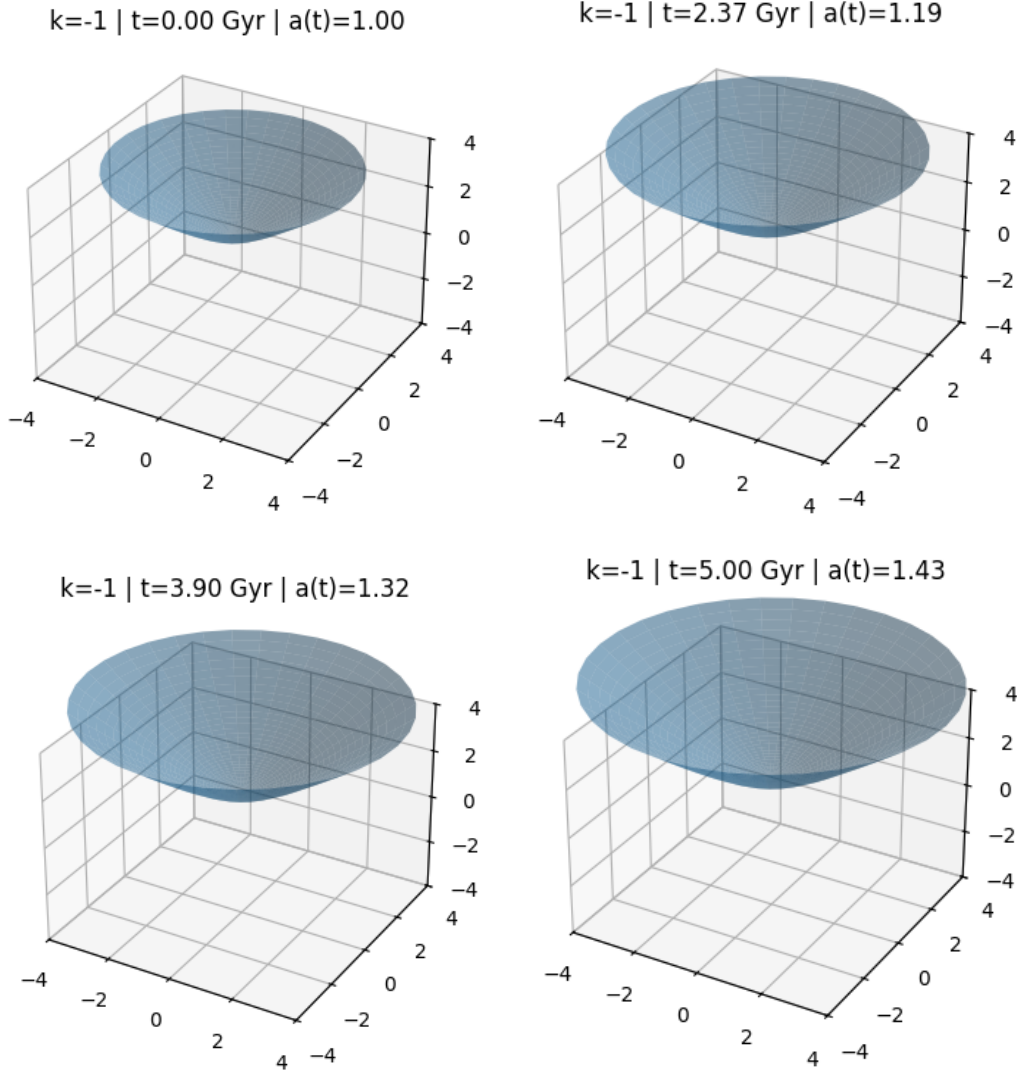
Cuadro 2.2: Geometría del espacio plano  $k=0$ . Elaboración propia

### $k=-1$

Para el caso  $k=-1$ , el factor de escala toma la forma 2.17 lo que introduce una evolución que aparenta comenzar  $a(0) = 0$ . Aunque esto podría interpretarse como una singularidad inicial, en realidad se trata de un efecto de la elección de coordenadas, ya que el espacio-tiempo de de Sitter es geodésicamente completo. Esta representación resulta útil para el análisis de horizontes cosmológicos y de regiones causalmente accesibles, es decir, partes del universo con las que se puede tener interacción física o intercambio de información, pues describe una región del espacio-tiempo relevante para observadores particulares (Ryden, 2006).

$$a(t) = \frac{c}{H} \sinh (H(t - t_0)) \quad (2.17)$$

Muestra una expansión en un universo hiperbólico lo cual se muestra en el Cuadro 3.1.



Cuadro 2.3: Geometría del espacio plano  $k=-1$ . Elaboración propia

En conjunto, estas tres representaciones ponen de manifiesto una dificultad conceptual importante en la enseñanza de la cosmología como lo es el formalismo matemático (Ayala, 2007); puede sugerir la existencia de universos distintos cuando, en realidad, se trata de diferentes descripciones de una misma entidad geométrica. En este sentido, la comprensión del papel del parámetro  $k$  en el universo de de Sitter requiere trascender la interpretación puramente algebraica del factor de escala y adoptar una perspectiva geométrica que

permita reconocer la equivalencia entre estas descripciones. Esta observación refuerza la importancia de integrar representaciones visuales y geométricas en los procesos de enseñanza, con el fin de facilitar la comprensión profunda de los modelos cosmológicos (Castillo et al., 2025).

## 2.4. Cálculo tensorial

### 2.4.1. Tensor métrico

El tensor métrico  $g_{\mu\nu}$  es un campo tensorial porque a cada punto del espacio-tiempo le asigna un tensor que permite medir distancias e intervalos. En otras palabras, define la geometría del espacio-tiempo y permite, entre muchas cosas, medir distancias y trayectorias.

El elemento de línea  $ds^2$  se define como el intervalo entre eventos en el espacio-tiempo, en términos del tensor métrico se expresa de la forma.

$$ds^2 = g_{\mu\nu} dx^\mu dx^\nu \quad (2.18)$$

La solución de W. de Sitter describe un universo vacío, homogéneo e isótropo, con una constante cosmológica positiva, lo que significa que el universo modelado así, se encuentra en expansión acelerada. La métrica de de Sitter se puede expresar en diferentes coordenadas, pero para efectos prácticos se va a hacer uso de coordenadas cuadráticas de la forma

$$ds^2 = -c^2 dt^2 + a(t)^2 [dx^2 + dy^2 + dz^2] \quad (2.19)$$

El tensor métrico es un tensor simétrico, lo que significa que su inversa también es simétrica. Entonces, para el caso del universo de de Sitter  $g_{\mu\nu}$  y su inversa  $g^{\mu\nu}$  están descritos por matrices  $4 \times 4$ :

$$g_{\mu\nu} = \begin{bmatrix} -c^2 & 0 & 0 & 0 \\ 0 & a(t)^2 & 0 & 0 \\ 0 & 0 & a(t)^2 & 0 \\ 0 & 0 & 0 & a(t)^2 \end{bmatrix} \quad (2.20)$$

y

$$g^{\mu\nu} = \begin{bmatrix} -\frac{1}{c^2} & 0 & 0 & 0 \\ 0 & \frac{1}{a(t)^2} & 0 & 0 \\ 0 & 0 & \frac{1}{a(t)^2} & 0 \\ 0 & 0 & 0 & \frac{1}{a(t)^2} \end{bmatrix} \quad (2.21)$$

A menudo, se suele encontrar el ajuste  $c = 1$  para facilitar procedimientos.

### 2.4.2. Métrica de Friedmann-Lemaître-Robertson-Walker

La métrica de Friedmann-Lemaître-Robertson-Walker (FLRW) es uno de los sistemas de coordenadas espacio-tiempo más simples, propuesto como una solución de las ecuaciones de campo de Einstein de un universo en expansión (Gürses & Heydarzade, 2020). Por otro lado, esta métrica tiene en cuenta las propiedades de homogeneidad e isotropía, características de un modelo de máxima simetría. Además el modelo de de Sitter es equivalente a una métrica de Robertson-Walker en un universo plano y con el factor de  $a \propto e^{Ht}$  (Weinberg, 2008), y el elemento de línea está descrita por:

$$ds^2 = -c^2 dt^2 + a^2(t) \left[ \frac{dr^2}{1 + kr^2} + r^2(d\theta^2 + \sin^2 \theta \cdot d\phi^2) \right] \quad (2.22)$$

A continuación se presenta una imagen con fines prácticos para el análisis de los ángulos y el radio en las coordenadas esféricas en las que está representada la métrica.

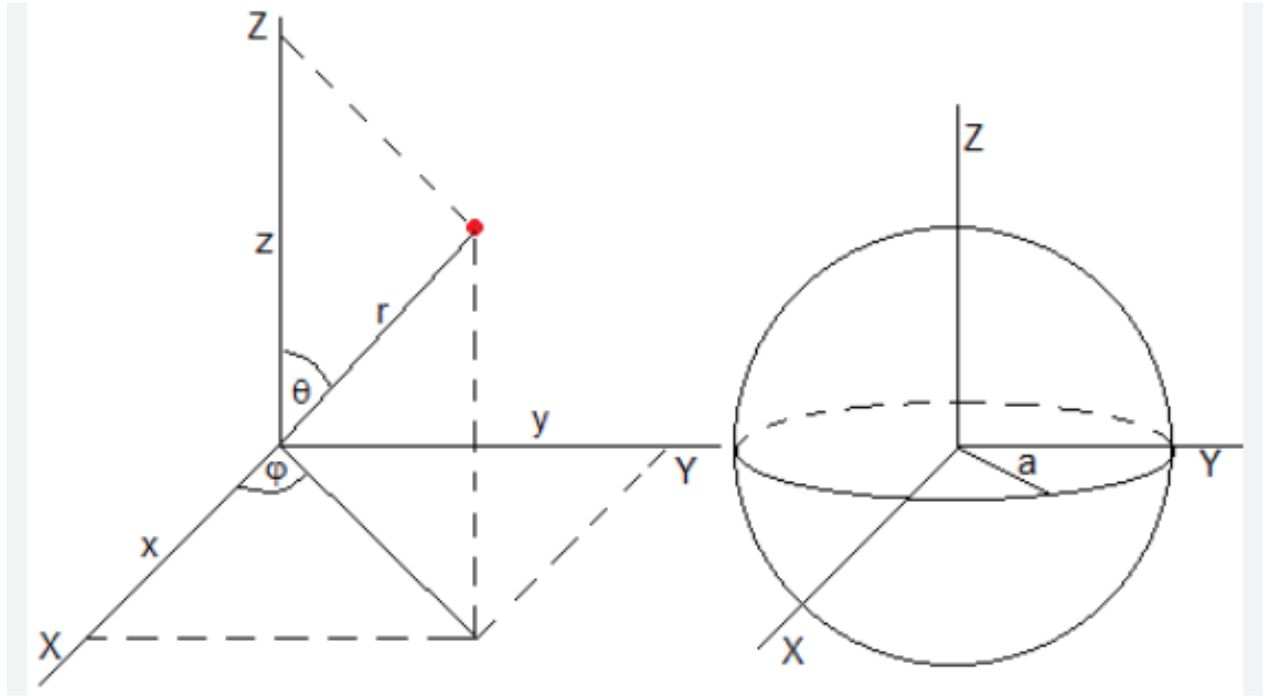


Figura 2.4: Representación gráfica de las coordenadas esféricas. Fuente : Ángel Franco García, 2016 disponible en <http://www.sc.ehu.es/sbweb/fisica3/cuantica/separacion/esferica.html>

### 2.4.3. Métrica en de Sitter

Existen distintas formas de escribir la métrica en el universo de de Sitter, dependiendo del propósito, unas resultan más convenientes que otras. Por ejemplo, es más eficiente estudiar las ondas gravitacionales haciendo uso de la métrica en coordenadas cilíndricas ya que induce a una simetría axial y reducen la cantidad de componentes no nulos en las ecuaciones en comparación con las coordenadas esféricas (Weinberg, 1972).

En términos generales, la métrica de de Sitter se puede expresar como aparece en la ecuación 2.22, en donde se puede apreciar que esta expresión varía según sea la geometría del espacio; es decir, que depende del parámetro  $k$ , lo que permite modelar el espacio-tiempo en expansión y contracción, conocida como métrica de Robertson-Walker (Liddle, 2015).

A partir de la expresión 2.22 de la métrica FRLW se reescribieron las demás en coordenadas planas, cilíndricas y esféricas.

#### Coordenadas planas $k=0$

Partiendo de las coordenadas FLRW y teniendo en cuenta la geometría plana del universo  $k = 0$  se hace el cambio de variables con las siguientes transformaciones.

$$\begin{aligned}x &= r \cdot \sin\theta \cdot \cos\phi \\y &= r \cdot \sin\theta \cdot \sin\phi \\z &= r \cdot \cos\theta\end{aligned}\tag{2.23}$$

Y se obtiene como resultado el elemento de línea en el universo de de Sitter en coordenadas planas muy útil para las simulaciones y códigos en Python que aparecen en este trabajo.

$$ds^2 = -c^2 dt^2 + a^2(t) [dx^2 + dy^2 + dz^2]\tag{2.24}$$

#### Coordenadas hiperbólicas abiertas $k=-1$

En primer lugar se hace el cambio de coordenadas con las transformaciones 2.25. Este sistema de coordenadas es muy útil para estudiar fenómenos de inflación y modelos tempranos del universo (Mukhanov, 2005).

$$\begin{aligned}d\Omega &= d\theta^2 + \sin^2(\theta) \cdot d\phi^2 \\r &= \sinh(\chi)\end{aligned}\tag{2.25}$$

Haciendo la transformación de variables, donde el ángulo  $\chi$  se define como distancia radial medida dentro de una geometría hiperbólica, y reemplazando se obtiene la siguiente expresión.

$$ds^2 = -c^2 dt^2 + \left(\frac{c}{H}\right)^2 \sinh^2(Ht) [d\chi^2 + \sinh^2(\chi) \cdot d\Omega^2] \quad (2.26)$$

### Coordenadas esféricas

Las coordenadas esféricas permiten modelar un universo finito sin borde y analizar propiedades globales del espacio-tiempo (Wald, 1984). Para obtener el elemento de línea se emplean las siguientes transformaciones.

$$\begin{aligned} d\Omega &= d\theta^2 + \sin^2(\theta) \cdot d\phi^2 \\ r &= \sin(\chi) \end{aligned} \quad (2.27)$$

Se obtiene la siguiente expresión.

$$ds^2 = -c^2 dt^2 + \left(\frac{c}{H}\right)^2 \cosh^2(Ht) [d\chi^2 + \sin^2(\chi) \cdot d\Omega^2] \quad (2.28)$$

Las expresiones se obtuvieron de la deducción hecha por Pascu (2024). Se evidencia que la deducción de estas coordenadas es complejo y extenso; por ello, se diseñó el siguiente código el cual permite expresar el elemento de línea  $ds^2$  según sea la geometría del espacio  $k$  que se elija.

```

import sympy as sp

k=float(input("Elija la geometría del espacio: "))

# Variables simbólicas

t, r, chi, theta, phi, H = sp.symbols('t r chi theta phi H', real=True)

# Función para la métrica en de Sitter
def metrica_de_sitter(k):

    if k == 0:
        # Coordenadas cartesianas
        x, y, z = sp.symbols('x y z')
        a = sp.exp(H*t)

        coords = [t, x, y, z]

        g = sp.diag(
            -1,
            a**2,
            a**2,
            a**2
        )

    elif k == 1:
        # Coordenadas esféricas cerradas (chi)
        a = sp.cosh(H*t)

        coords = [t, chi, theta, phi]

        g = sp.diag(
            -1,
            a**2,
            a**2 * sp.sin(chi)**2,
            a**2 * sp.sin(chi)**2 * sp.sin(theta)**2
        )

```

---

```

elif k == -1:
    # Coordenadas hiperbólicas abiertas
    a = sp.sinh(H*t)

    coords = [t, chi, theta, phi]

    g = sp.diag(
        -1,
        a**2,
        a**2 * sp.sinh(chi)**2,
        a**2 * sp.sinh(chi)**2 * sp.sin(theta)**2
    )

else:
    raise ValueError("k debe ser 0, 1 o -1")

return coords, g

#Elemento de línea
def elemento_linea(coords, g):
    # Crear diferenciales
    dcoords = sp.symbols(['d' + str(c) for c in coords])

    ds2 = 0
    n = len(coords)

    for i in range(n):
        for j in range(n):
            ds2 += g[i,j] * dcoords[i] * dcoords[j]

    return sp.simplify(ds2), dcoords

coords, g = metrica_de_sitter(k)
ds2, dcoords = elemento_linea(coords, g)
ds2_simplificado = sp.factor(ds2)

print("Coordenadas:")
print(coords)

print("\n Métrica g_mu_nu:")
sp.pprint(g)

ds2_simplificado = sp.factor(ds2)
print("\nds^2")
print(sp.pretty(ds2_simplificado, use_unicode=True))

```

Figura 2.5: Código de elemento de línea. Elaboración propia

El código muestra la matriz del tensor métrico  $g_{\mu\nu}$ , las coordenadas que se usan y el elemento de línea  $ds^2$ .

```

Elija la geometría del espacio: 0
Coordenadas:
[t, x, y, z]

Métrica g_mu_nu:
[ -1  0  0  0 ]
[  0  2·H·t  0  0 ]
[  0  0  2·H·t  0 ]
[  0  0  0  2·H·t ]

ds^2
= - dt^2 + dx^2 · ε + dy^2 · ε + dz^2 · ε

```

Figura 2.6: Ejemplo de ejecución del código para 2.5  $k=0$ . Elaboración propia

#### 2.4.4. Símbolos de Christoffel

Cuando se habla del cambio de un vector en un espacio cartesiano, su derivada es sencilla, simplemente se calcula la derivada de cada una de las componentes del vector. No obstante, cuando se quiere hacer el mismo proceso en un espacio vectorial en coordenadas polares o esféricas, el desarrollo se complica, debido a que, por ejemplo, en este caso el espacio es curvo, lo que genera cambios en los vectores y es que estos apuntan en distintas direcciones a medida que se mueven en él (Fleisch, 2011). En otras palabras, al mover el vector en el espacio curvo, su variación debe describir otro vector que no cambie por la curvatura.

Todos los vectores se pueden describir como una cantidad de veces sus vectores base en un espacio vectorial dado. Sin embargo, en un espacio curvo estos vectores pueden variar de un punto a otro; los símbolos de Christoffel representan esa cantidad (Afranchi, 2022). Estos valores son importantes de encontrar y, según sea el tensor métrico del sistema, es más sencillo o no de llegar a ellos; por lo general, para determinar cada símbolo se deben seguir unos procesos matemáticos extensos y propensos al error humano, por consiguiente, se realizó el siguiente código que permite hallar los símbolos de Christoffel de cualquier tensor métrico que se desee.

```

import sympy as sp

# Definir coordenadas
t, x, y, z = sp.symbols('t x y z')
coords = [t, x, y, z]
n = len(coords)

# Definir la función de escala (de Sitter:  $a(t) = \exp(Ht)$ )
H = sp.Symbol('H', real=True)
a = sp.Function('a')(t)
a_expr = sp.exp(H*t)

# Definir la métrica  $g_{\{\mu\nu\}}$ 
g = sp.Matrix([
    [-1, 0, 0, 0],
    [0, a_expr**2, 0, 0],
    [0, 0, a_expr**2, 0],
    [0, 0, 0, a_expr**2]
])

# Inversa de la métrica
g_inv = g.inv()

# Inicializar los símbolos de Christoffel  $\Gamma^{\lambda}_{\{\mu\nu\}}$ 
Gamma = [[[0 for _ in range(n)] for _ in range(n)] for _ in range(n)]

# Cálculo de los símbolos de Christoffel

Gamma = [[[0]*n for _ in range(n)] for _ in range(n)]
for rho in range(n):
    for mu in range(n):
        for nu in range(n):
            suma = 0

```

```

for sigma in range(n):
    suma += g_inv[rho, sigma]*(
        sp.diff(g[nu, sigma], coords[mu]) +
        sp.diff(g[mu, sigma], coords[nu]) -
        sp.diff(g[mu, nu], coords[sigma])
    )
Gamma[rho][mu][nu] = sp.simplify(suma*0.5)

# Mostrar símbolos de Christoffel no nulos
print(" Símbolos de Christoffel no nulos:")
for rho in range(n):
    for mu in range(n):
        for nu in range(n):
            if Gamma[rho][mu][nu] != 0:
                print(f"Γ{rho}{mu}{nu} =", Gamma[rho][mu][nu])

```

Figura 2.7: Código símbolos de Christoffel. Elaboración propia

Esto da como resultado todos los símbolos de Christoffel no nulos según sea el tensor métrico  $g_{\mu\nu}$ , para nuestro caso, se trabaja con el universo de de Sitter, obteniendo como resultado lo que se muestra en la Figura 2.8

```

Símbolos de Christoffel no nulos:
Γ011 = 1.0*H*exp(2*H*t)
Γ022 = 1.0*H*exp(2*H*t)
Γ033 = 1.0*H*exp(2*H*t)
Γ101 = 1.0*H
Γ110 = 1.0*H
Γ202 = 1.0*H
Γ220 = 1.0*H
Γ303 = 1.0*H
Γ330 = 1.0*H

```

Figura 2.8: Símbolos de Christoffel no nulos en de Sitter. Elaboración propia

Primero se definen las coordenadas (**t**, **x**, **y**, **z**). Se guardan en la lista llamada **coords**, y luego, se calcula **n = len(coords)**. Es la misma secuencia que se ha usado en los códigos anteriores.

En este caso son cuatro, así que todo lo que venga después va a recorrer un espacio-tiempo de cuatro dimensiones. Luego se introduce la constante **H**, que representa la constante de Hubble, y se define en función del tiempo con la expresión 2.16 en el universo de Sitter con geometría plana. Sin embargo, en

la métrica se mantiene  $a(t)$  como función general, lo cual hace que el cálculo sea más sencillo.

Luego se construye la matriz del tensor métrico  $g_{\mu\nu}$  que como se pudo ver en 2.20 que corresponde a la métrica FLRW plana en coordenadas cartesianas 2.5.

Como la matriz es diagonal, su inversa es sencilla de calcular. El código usa **g.inv()** para obtener la inversa del tensor métrico  $g^{\mu\nu}$ , como se apreció en 2.21 que será necesaria para construir los símbolos de Christoffel.

Después se crea una estructura tridimensional llamada  $\Gamma$  para almacenar los símbolos de Christoffel ( $\Gamma_{\mu\nu}^\rho$ ). Tiene tres índices porque estos objetos dependen de tres posiciones, uno arriba y dos abajo. El cálculo de los símbolos de Christoffel se hace con varios ciclos **for** que recorren todos los índices posibles. Lo que el programa implementa ahí es directamente la fórmula 2.29 adaptado de Janssen (2013, p. 100).

$$\Gamma_{\mu\nu}^\rho = \frac{1}{2}g^{\rho\sigma} (\partial_\mu g_{\nu\sigma} + \partial_\nu g_{\mu\sigma} - \partial_\sigma g_{\mu\nu}) \quad (2.29)$$

Para cada combinación de índices, el código calcula las derivadas correspondientes de la métrica y suma los términos. Como en esta métrica todo depende solo del tiempo porque el factor de escala que es una función del tiempo, muchas derivadas respecto a  $(x, y, z)$  son cero, así que solo sobreviven algunos términos. Por eso, cuando se imprimen los resultados, solo aparecen los pocos símbolos distintos de cero.

El programa multiplica cada componente del tensor de Ricci  $R_{\mu\nu}$  por la correspondiente componente de  $g^{\mu\nu}$  y suma todas las combinaciones posibles. Luego simplifica el resultado y sustituye explícitamente  $a(t) = e^{Ht}$ , el resultado final debe dar una constante proporcional a  $H^2$ , lo cual coincide con que el espacio-tiempo de de Sitter tiene curvatura constante como se definió en las ecuaciones de Friedmann en 2.9.

Todo el código es una traducción directa de las definiciones geométricas de relatividad general a un procedimiento automático en el cual primero se define la métrica y luego se obtienen los símbolos de Christoffel. Es exactamente el mismo camino que se sigue a mano, pero aquí lo hace el computador recorriendo sistemáticamente todos los índices.

### 2.4.5. Conexión Levi-Civita

En geometría diferencial una conexión es una regla que permite comparar vectores en distintos punto de una variedad, es decir que permite derivar tensores coherentemente con la métrica (Lee, 2018). Esto soluciona el problema que se enunció en el apartado de los Símbolos de Christoffel sobre la derivada de un vector en el espacio curvo porque en palabras de Janssen (2013) la norma del vector cambia debido al transporte paralelo.

La conexión se puede definir como la derivada espacial covariante de un campo de vectores  $X^\mu$  en la dirección de otro  $Y^\nu$  (Balaguera, 2006). La conexión Levi-Civita cumple con dos condiciones (Janssen, 2013):

1. La derivada covariante de la métrica es igual a cero  $\nabla_\mu g_{\nu\sigma} = 0$
2. La conexión es simétrica cuando  $\Gamma_{\sigma\nu}^\rho = \Gamma_{\nu\sigma}^\rho$

En TGR la gravitación no se toma como una fuerza sino como la curvatura del espacio tiempo, lo cual está medido por el tensor y escalar de Ricci. Por ende, bajo las condiciones de la conexión de Levi-Civita, las geodésicas proporcionan el extremo local de la longitud de una curva que conecta dos puntos (Nakahara, 2003). La primera condición se conoce como compatibilidad con la métrica y la segunda sugiere que solo con el tensor de Riemann es posible hallar el conmutador de dos derivadas covariantes ya que el tensor de torsión es cero. Además que la torsión nula garantiza la simetría de los símbolos de Christoffel y, teniendo en cuenta que, la solución de de Sitter está bajo condiciones de máxima simetría, esta propiedad es compatible con dicho escenario (Poisson, 2004). Esto significa que hay una geometría coherente en el espacio-tiempo y se pueden comparar vectores en diferentes puntos. Entonces las partículas libres se mueven en geodésicas determinadas por la conexión.

Ahora bien, en la métrica de Sitter el universo está en expansión acelerada, por lo tanto se hace necesario que las longitudes y los ángulos definidos en la métrica se mantengan, y por ese motivo se hace la derivada covariante de la métrica (Janssen, 2013).

$$\nabla_\mu g_{\nu\sigma} = \partial_\mu g_{\nu\sigma} - \Gamma_{\mu\nu}^\rho g_{\rho\sigma} - \Gamma_{\mu\sigma}^\rho g_{\nu\rho} = 0 \quad (2.30)$$

A partir de 2.30 se obtienen las condiciones que garantizan que el espacio no cambie arbitrariamente en ninguna dirección, es decir, que no haya torsión del vector (Castillo et al., 2025, pág 25), obteniendo las siguientes expresiones:

$$\nabla_\mu g_{\nu\sigma} = 0 \quad \nabla_\nu g_{\sigma\mu} = 0 \quad \nabla_\sigma g_{\mu\nu} = 0$$

Entonces, si se suman las 2 primeras expresiones y se le resta la tercera se obtiene la expresión para los símbolos de Christoffel vista en el apartado anterior 2.29.

#### 2.4.6. Tensor de Riemann

En primer lugar, se define un espacio de Riemann como un espacio curvo que localmente es euclídeo, conocido como variedad. Teniendo en cuenta que las ecuaciones de campo salen de medir la razón de la curvatura del espacio-tiempo (Campo, 2022), el tensor que especifica dicha curvatura es el tensor de Riemann, que está definido por (Carroll, 2004):

$$R^\rho_{\sigma\mu\nu} = \partial_\mu \Gamma^\rho_{\sigma\nu} - \partial_\nu \Gamma^\rho_{\sigma\mu} + \Gamma^\rho_{\lambda\mu} \Gamma^\lambda_{\sigma\nu} - \Gamma^\rho_{\lambda\nu} \Gamma^\lambda_{\sigma\mu} \quad (2.31)$$

Evidentemente, esto es un cálculo extenso y puede inducir al error humano, puesto que como se puede apreciar, el tensor de Riemann depende de los símbolos de Christoffel que a su vez dependen del tensor métrico del espacio. Esto, de cierto modo, dificulta su solución dependiendo del sistema de coordenadas que se emplee. Teniendo en cuenta que este tensor también es de orden 4, puede tener hasta 256 componentes (Avalos, 2000, pág 16) y aunque algunas se anulen, si se hace el proceso a mano paso a paso, demandaría tiempo y concentración. Por ello se hace mucho más útil hacer uso del tensor de Ricci dado que es la contracción del tensor de Riemann, minimizando la cantidad de componentes.

#### 2.4.7. Tensor de Ricci

El tensor de Ricci, en pocas palabras, es un tensor de orden 2 que mediante la contracción de índices del tensor de Riemann, reduce la cantidad de información necesaria para describir la curvatura. (Berzosa Montero, 2024), y está definido por la ecuación 2.32. Es un tensor que, geométricamente, describe cómo cambia el volumen de un conjunto de geodésicas en un espacio curvo (Wald, 1984) Y está expresado de la siguiente forma (Carroll, 2004, pág. 129).

$$R_{\mu\nu} = R^\rho_{\mu\sigma\nu} \quad (2.32)$$

Esta expresión se obtiene mediante la contracción igualando  $\rho = \sigma$  anulando componentes de modo tal que, el tensor de Ricci quede representado en términos de  $\mu$  y  $\nu$ , adecuado de (Wald, 1984, pág 40). Cabe recalcar que

los índices son representaciones espacio-temporales, siendo esto una parte relevante en el significado del tensor de Ricci en el universo de de Sitter porque representa la curvatura simétrica y la expansión acelerada del universo, dado que el tensor es proporcional a la métrica escalada por la energía del vacío.

### Escalar de Ricci

En ese sentido, el escalar de Ricci es un valor que representa la curvatura del espacio-tiempo, y este varía según sea el universo el cual se esté estudiando porque también depende de la geometría del espacio  $k$ . La expresión que lo define está dada por la ecuación 2.33 (Carroll, 2004, pág. 333).

$$R = 6 \left[ \frac{\ddot{a}}{a} + \left( \frac{\dot{a}}{a} \right)^2 + \frac{k}{a^2} \right] \quad (2.33)$$

Para dar cuenta de esto, al código de los símbolos de Christoffel 2.7 se le agrega el código 2.9

```
#Ricci
R = [[0]*n for _ in range(n)]
for mu in range(n):
    for nu in range(n):
        suma = 0
        for lam in range(n):

            term1 = sp.diff(Gamma[lam][mu][nu], coords[lam])
            term2 = sp.diff(Gamma[lam][mu][lam], coords[nu])

            sum1 = 0
            sum2 = 0

            for sigma in range(n):
                sum1 += Gamma[lam][lam][sigma] * Gamma[sigma][mu][nu]
                sum2 += Gamma[lam][nu][sigma] * Gamma[sigma][mu][lam]

            suma += term1 - term2 + sum1 - sum2

        R[mu][nu] = sp.simplify(suma)

# Escalar de Ricci
R_mat = sp.Matrix(R)
Ricci = sp.simplify((g_inv * R_mat).trace())
print(" Escalar de Ricci:", Ricci)
```

Figura 2.9: Código escalar de Ricci. Elaboración propia

Dando como resultado el valor de la constante de Ricci en el universo de de Sitter que es  $R = 12H^2$  tomando  $c=1$ , como se aprecia en la Figura 2.10. El

valor depende de las componentes del espacio-tiempo dado que el escalar de Ricci es proporcional al tensor métrico (Schutz, 2009, pág 164) y es la traza del Tensor de Ricci; es una cantidad agregada. Entonces el escalar de Ricci es proporcional a las  $n$  dimensiones del espacio, si es  $4 \times 4$  como en el caso de de Sitter el escalar es  $12H^2$  como se muestra en el apéndice G.

$$\text{Escalar de Ricci: } 12.0 * H^{**2}$$

Figura 2.10: Valor de la constante de Ricci en de Sitter. Elaboración propia

#### 2.4.8. Tensor Energía-Momento

En principio, el causante de la curvatura del espacio tiene relación con la gravedad, y la fuente de la fuerza gravitatoria es la masa de los cuerpos; por consiguiente, es correcto afirmar que la materia es quien curva el espacio. Ahora bien, gracias a la relatividad especial, sabemos que la masa está relacionada con la energía por la ecuación  $E = mc^2$  donde  $E$  es la energía,  $m$  la masa y  $c$  la velocidad de la luz en el vacío. La gravedad se acopla a cualquier tipo de masa y energía. Entonces, se hace necesario definir el tensor energía-momento, el cual Janssen (2013) lo define como el flujo del cuadvivector momento en una superficie constante.

Ese vector termina siendo la traza de la matriz  $T_{\mu\nu}$  y en términos generales está representado por la siguiente expresión 2.34.

$$T_{\mu\nu} = \begin{bmatrix} T_{00} & T_{01} & T_{02} & T_{03} \\ T_{10} & T_{11} & T_{12} & T_{13} \\ T_{20} & T_{21} & T_{22} & T_{23} \\ T_{30} & T_{31} & T_{32} & T_{33} \end{bmatrix} \quad (2.34)$$

De modo general, el tensor energía momento se define de la forma

$$T_{\mu\nu} = (\rho_\Lambda + P) V_\mu V_\nu - P g_{\mu\nu} \quad (2.35)$$

siendo  $\rho$  la densidad de energía del vacío,  $P$  la presión que surge de las interacciones de las partículas del fluido y  $V_\mu$  es el vector cuadvivelocity. (Janssen, 2013, pág, 155).

Para el universo de de Sitter se piensa este tensor como un fluido perfecto, de allí se infiere que el tensor métrico está dado por (Misner et al., 1973, pág, 140):

$$T_{\mu\nu} = \begin{bmatrix} \rho & 0 & 0 & 0 \\ 0 & P & 0 & 0 \\ 0 & 0 & P & 0 \\ 0 & 0 & 0 & P \end{bmatrix} \quad (2.36)$$

Teniendo en cuenta que la curvatura del espacio-tiempo no se manifiesta en un punto, sino en cómo cambian las relaciones entre puntos cercanos, el tensor métrico puede expresarse como el tensor de Minkowsky  $g_{\mu\nu} = \eta_{\mu\nu}$ .

En el universo de de Sitter se puede apreciar que lo que expande al universo es una presión negativa, como se puede inferir del trabajo de González Álvarez (s.f.) donde  $P = -\rho c^2$ . Esto puede resultar contraintuitivo; no obstante, se puede pensar en lo que pasaría si de repente la atmósfera de nuestro planeta fuera desapareciendo progresivamente. Lo que pasaría es que nuestros cuerpos empezarían a inflarse por la falta de presión, y al ir disminuyendo se puede considerar como una presión negativa. Esto, en términos cosmológicos, sería lo que impulsa la expansión del universo vacío.

## 2.5. Métrica de de Sitter

Ahora bien, la expresión general para definir la métrica de de Sitter se puede expresar de distintas formas, como se evidenció en los apartados anteriores; es una expresión que varía según la geometría del espacio. Sin embargo, cabe recordar que este trabajo enfatiza el uso de la geometría del espacio plano  $k = 0$ .

Partiendo de la métrica FLRW que se trae de nuevo para efectos prácticos.

$$ds^2 = -c^2 dt^2 + a^2(t) [dx^2 + dy^2 + dz^2]$$

Y, por otro lado, teniendo en cuenta el tensor métrico 2.20 y su inversa 2.21 para hallar los símbolos de Christoffel no nulos del universo de de Sitter; es posible obtener el tensor de Ricci a partir de la contracción del tensor de Riemann cuya componente temporal es (Carroll, 2004, pág. 333):

$$R_{00} = -\frac{3\ddot{a}}{a} \quad (2.37)$$

Y para la componente espacial, donde  $\delta_{\mu\nu}$  es la matriz identidad.

$$R_{\mu\nu} = \frac{a\ddot{a} + 2\dot{a}^2}{c^2} \delta_{\mu\nu} \quad (2.38)$$

Se sustituye con la expresiones 2.16 y de las ecuaciones de Friedmann 2.1 y 2.2, se obtiene la matriz del tensor de Ricci para el universo de de Sitter.

$$R_{\mu\nu} = \begin{bmatrix} -3H^2 & 0 & 0 & 0 \\ 0 & \frac{3H^2}{c^2}a^2(t) & 0 & 0 \\ 0 & 0 & \frac{3H^2}{c^2}a^2(t) & 0 \\ 0 & 0 & 0 & \frac{3H^2}{c^2}a^2(t) \end{bmatrix} \quad (2.39)$$

En el cual se puede inferir que  $R_{\mu\nu} = \frac{3H}{c^2}g_{\mu\nu}$ , siendo esta la generalización de las ecuaciones de campo de de Sitter (Villegas, 2019, pág 20).

Lo que es equivalente a la expresión que define el factor de expansión del universo de de Sitter en términos de la constante cosmológica (Villegas, 2019, pág 21)

$$H = \sqrt{\frac{\Lambda c^2}{3}} \quad (2.40)$$

El cálculo explícito del tensor de Ricci para la métrica de de Sitter muestra que sus componentes son proporcionales al tensor métrico, lo que evidencia que se trata de un espacio-tiempo de curvatura constante determinado únicamente por la constante cosmológica.

## Capítulo 3

# Implicaciones físicas: Expansión acelerada y constante cosmológica

### 3.1. Homogeneidad

En este apartado del trabajo no se plantea como tal una consecuencia del universo en expansión acelerada, sino más bien una presunción optada por la comunidad científica, y más específicamente hablando, por de Sitter para desarrollar su solución. La homogeneidad se puede definir como la propiedad de un sistema que se conserva en cualquier punto del espacio; por ejemplo, el universo es homogéneo si, al observarlo a gran escala, su distribución de energía y materia es la misma en cualquier punto; no hay lugares privilegiados. Entonces, con base en esa definición, se puede decir que en la métrica de de Sitter, el universo es homogéneo porque la densidad de energía en el vacío, la curvatura y las demás propiedades geométricas son las mismas en cualquier punto del espacio.

Para dar cuenta de la homogeneidad, se realizó el código 3.1 que muestra cómo se supone un universo que, a gran escala, sea uniforme, aunque a pequeña escala tenga estructuras.

```

import numpy as np
import matplotlib.pyplot as plt

# Parámetros del universo simulado
N_galaxias = 5000
L = 1000

# Generar posiciones aleatorias de galaxias
x = np.random.uniform(0, L, N_galaxias)
y = np.random.uniform(0, L, N_galaxias)

# Fluctuaciones locales (simula estructuras)
densidad_extra = np.exp(-((x - L/2)**2 + (y - L/2)**2)/(2*(L/10)**2))
prob = 0.5 + 0.5*densidad_extra / densidad_extra.max()
mask = np.random.rand(N_galaxias) < prob
x, y = x[mask], y[mask]

# Observador
x0, y0 = L/2, L/2

# Escalas de observación
escalas = [20, 80, 300]

fig, axs = plt.subplots(1, len(escalas), figsize=(15,5))

for i, r in enumerate(escalas):

    # Galaxias cercanas al observador
    dentro = (x - x0)**2 + (y - y0)**2 <= r**2
    x_loc = x[dentro]
    y_loc = y[dentro]

    # Definir dominio local
    x_min_dom, x_max_dom = x0 - r, x0 + r
    y_min_dom, y_max_dom = y0 - r, y0 + r

```

```

# Dividir el universo en celdas cuadradas de lado r
n_celdas = 40
densidades = np.zeros((n_celdas, n_celdas))

dx = (x_max_dom - x_min_dom) / n_celdas
dy = (y_max_dom - y_min_dom) / n_celdas

for j in range(n_celdas):
    for k in range(n_celdas):
        x_min = x_min_dom + j*dx
        x_max = x_min_dom + (j+1)*dx
        y_min = y_min_dom + k*dy
        y_max = y_min_dom + (k+1)*dy

        dentro_celda = (
            (x_loc >= x_min) & (x_loc < x_max) &
            (y_loc >= y_min) & (y_loc < y_max)
        )

        densidades[j,k] = np.sum(dentro_celda) / (dx*dy)

axs[i].imshow(
    densidades.T,
    origin='lower',
    extent=[x_min_dom, x_max_dom, y_min_dom, y_max_dom],
    cmap='plasma'
)

axs[i].set_title(f"Escala = {r} Mpc")
axs[i].set_xlabel("x (Mpc)")
axs[i].set_ylabel("y (Mpc)")

plt.suptitle("Demostración de homogeneidad cosmológica")
plt.tight_layout()
plt.show()

```

Figura 3.1: Código de homogeneidad. Elaboración propia

El resultado ayuda a visualizar una misma área del universo visto desde una distancia menor hasta una distancia mayor, lo que permite ver cómo el universo empieza a verse homogéneo. Ya que si un observador está desplazándose y constata que las propiedades de los distintos lugares no presentan variaciones apreciables, es correcto afirmar que el universo mantiene las mismas características en todas sus regiones y, en consecuencia, es homogéneo (Harrison, 2000).

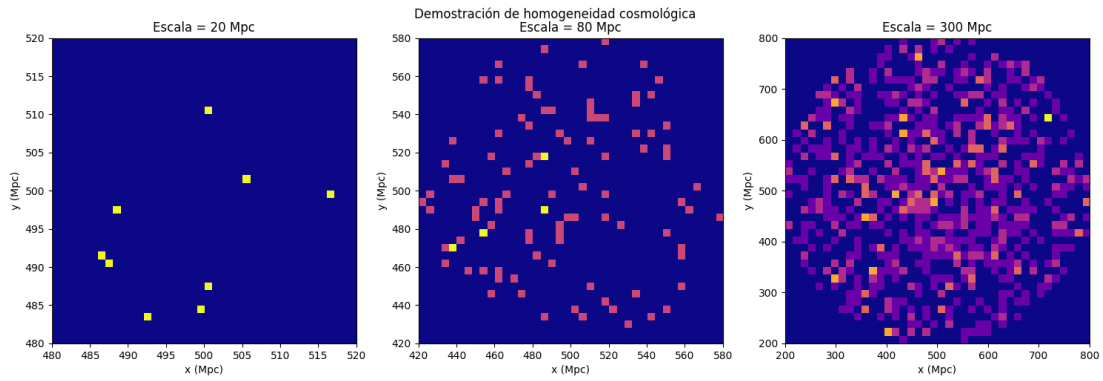


Figura 3.2: Representación de homogeneidad. Elaboración propia

El código simula un alejamiento en megapársec que permite dar cuenta de que el universo a gran escala, como se considera en el modelo de de Sitter, es homogéneo aunque a pequeña no lo sea.

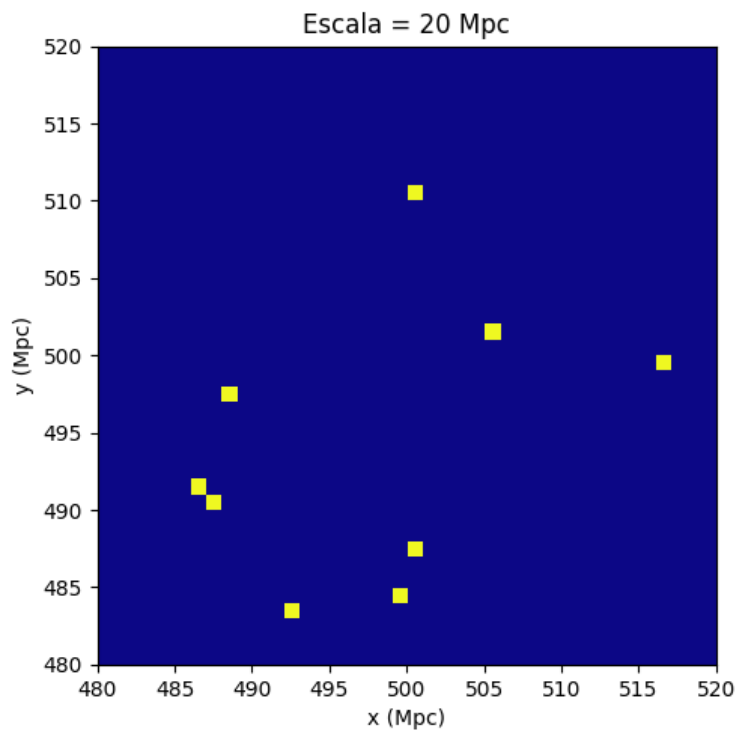


Figura 3.3: Universo a 20 Mpc. Elaboración propia

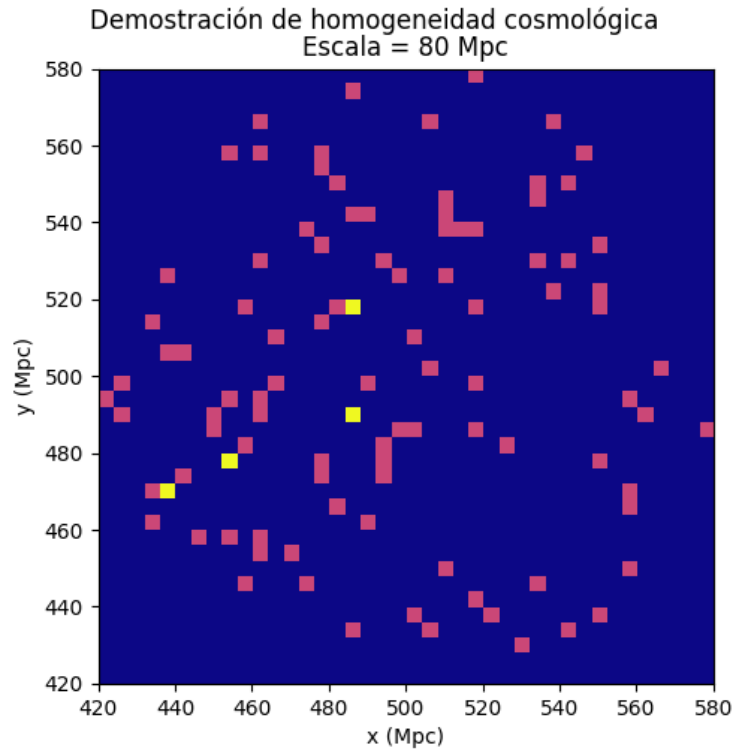


Figura 3.4: Universo a 80 Mpc. Elaboración propia

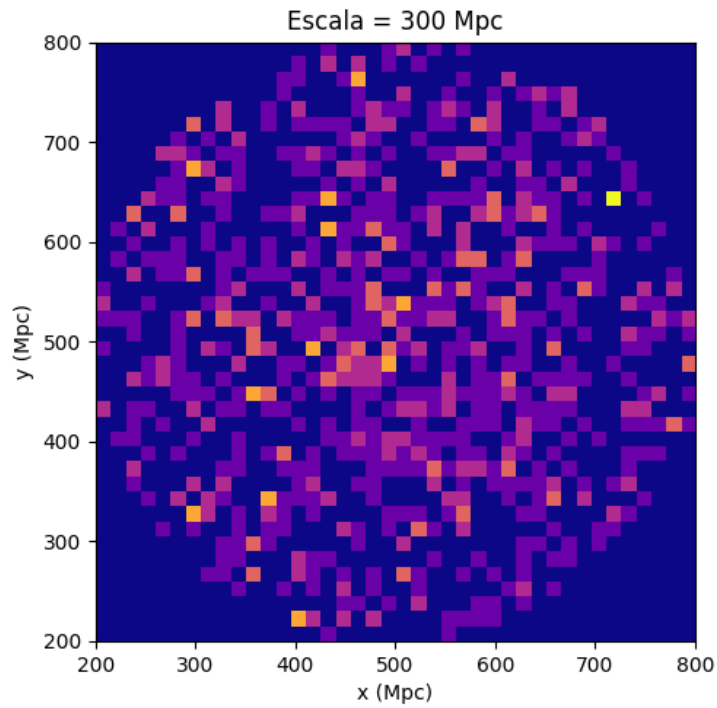


Figura 3.5: Universo a 300 Mpc. Elaboración propia

Los ejes de las gráficas representan coordenadas espaciales en unidades de

megapársecs dentro de una misma. Se mantienen constantes para todas las escalas de observación con el fin de garantizar una comparación directa entre las distintas resoluciones espaciales. De este modo, las diferencias observadas entre las gráficas no corresponden a cambios en el sistema físico, sino a variaciones en la escala a la cual se analiza la distribución de galaxias. También es importante tener en cuenta que el código genera un número de galaxias de manera aleatoria; por lo tanto, no siempre se obtendrán exactamente las mismas representaciones. Aunque el resultado será el mismo, es decir, que el código siempre mostrará homogeneidad.

### **3.2. Isotropía**

Se dice que un sistema es isótopo cuando, al observarlo desde un punto en el reposo, sus propiedades físicas son las mismas en todas las direcciones (Harrison, 2000). Sus propiedades no dependen de la dirección en la que se observe y esto determina altas nociones de simetría. El espacio isótropo es aquel en el que las leyes físicas y las observaciones no cambian con la dirección.

El modelo de de Sitter constituye una solución exacta a las ecuaciones de campo de Einstein bajo la premisa de isotropía y homogeneidad espacial. Esta configuración asume una distribución de densidad de energía uniforme. En consecuencia, es posible definir un tiempo cósmico global bajo el cual la métrica se mantiene invariante ante traslaciones y rotaciones, garantizando que las propiedades geométricas sean idénticas para cualquier observador comóvil en un instante dado. (Landau & Lifshitz, 1975).

### **3.3. Estructura causal en de Sitter**

La estructura causal en el universo en expansión conlleva a dar cuenta de que hay objetos que salen del cono de luz, es decir que, son objetos los cuales nunca podremos ver. Las líneas de universo en el espacio-tiempo describen la trayectoria de objetos que existen de manera continua, ocupando una posición en cada momento del tiempo. En cambio, los puntos en el espacio-tiempo corresponden a eventos puntuales, es decir, sucesos que ocurren en un lugar específico del espacio y en un único instante temporal (Harrison, 2000). Para dar cuenta de ello se diseñó el código 3.6.

```

import numpy as np
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation
from IPython.display import HTML

#Constantes físicas
H_km = 70 # km/s/Mpc
c = 3e5 # km/s
Mpc_km = 3.086e19 # km

# Convertir H a 1/s
H = H_km / Mpc_km

# Tiempo en segundos
Gyr = 3.154e16
t_vals = np.linspace(0, 30*Gyr, 200)

# Galaxias en coordenadas comóviles (Mpc)
x_comovil = np.linspace(100, 4000, 25)

# Precalcular trayectoria de la luz
dt = t_vals[1] - t_vals[0]
a_vals = np.exp(H * t_vals)

# Integral acumulada:  $r(t) = \int c/a(t) dt$ 
r_luz_vals = np.cumsum((c / a_vals)) * dt / Mpc_km # en Mpc

# Horizonte teórico
r_h = c / H_km # Mpc

```

```

# Gráfica
fig, ax = plt.subplots()
ax.set_xlim(0, 6000)
ax.set_xlabel("Distancia física (Mpc)")
ax.set_ylim(-1, 1)

galaxias, = ax.plot([], [], 'bo', label="Galaxias")
luz, = ax.plot([], [], 'r-', linewidth=2, label="Luz emitida")
horizonte_line, = ax.plot([], [], 'g--', linewidth=2, label="Horizonte")

texto = ax.text(0.02, 0.8, '', transform=ax.transAxes)

ax.legend()

# Animación
def update(frame):
    t = t_vals[frame]
    a = np.exp(H * t)

    # Galaxias expansión física
    x_fis = a * x_comovil
    galaxias.set_data(x_fis, np.zeros_like(x_fis))

    # Luz emitida
    r_luz = r_luz_vals[frame]
    luz.set_data([0, r_luz], [0, 0])

    # Horizonte
    horizonte_line.set_data([r_h, r_h], [-1, 1])

    texto.set_text(f"t = {t/Gyr:.2f} Gyr")

    return galaxias, luz, horizonte_line, texto

ani = FuncAnimation(fig, update, frames=len(t_vals), interval=50)
HTML(ani.to_jshtml())

```

Figura 3.6: Código horizonte cosmológico. Elaboración propia

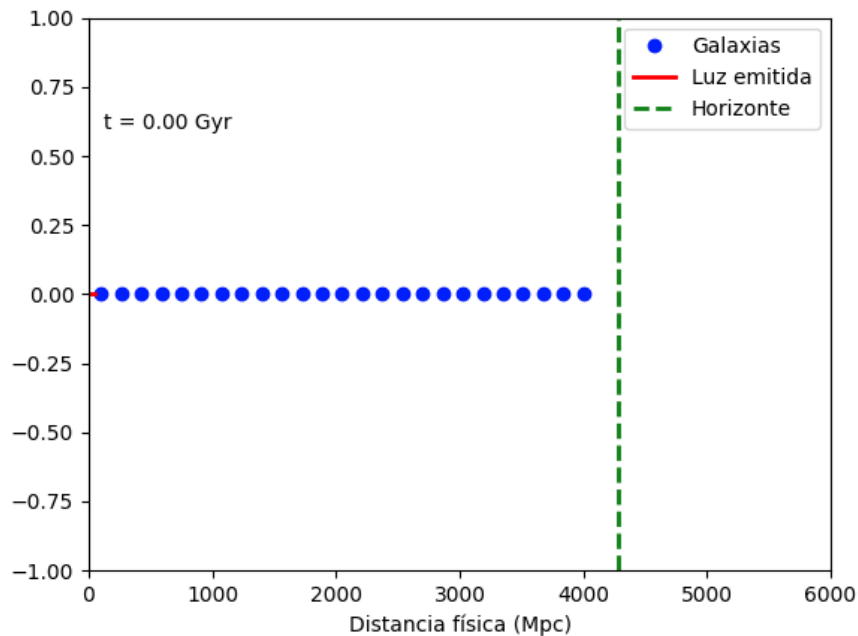
### 3.3.1. Horizonte cosmológico

En el universo hay objetos que nunca podremos llegar a ver debido a la expansión acelerada. En palabras de Harrison (2000) el universo observable normalmente es solo una porción del universo entero y el límite es a lo que

se conoce como horizonte cosmológico, más allá de él existen cuerpos que no pueden ser observados.

El horizonte cosmológico en el universo de de Sitter aparece como una distancia física constante debido a la naturaleza exponencial de la expansión. Sin embargo, esta constancia no implica que el horizonte sea una estructura estática en el espacio, sino que refleja un equilibrio dinámico entre la expansión del universo y la propagación de la luz (Bikwa et al., 2011) En donde aparece de nuevo el radio de curvatura como se mencionó en el apartado de la geometría del espacio y es el factor que acompaña el factor de escala en 2.14 y 2.17.

Al ejecutar el código se logra ver una animación como se puede apreciar en la Figura 3.7.



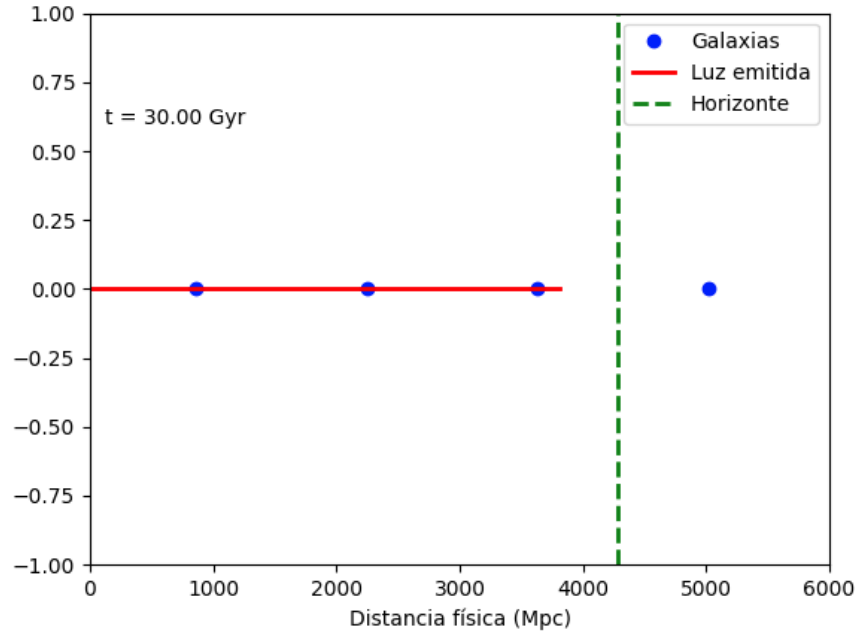
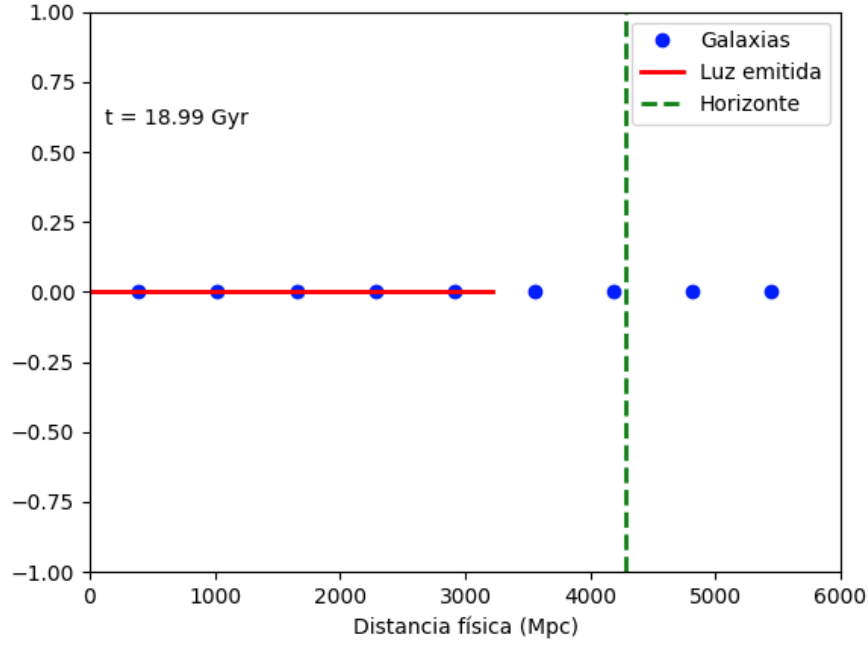


Figura 3.7: Simulación horizonte cosmológico. Elaboración propia

En el código aunque la velocidad de la luz parece que se va deteniendo, no es así, ya que  $c$  permanece constante en todo sistema de referencia local (Balaguera, 2006) En un espacio-tiempo en expansión su propagación efectiva se ve limitada por la dinámica del factor de escala. En particular, en el universo de de Sitter, el crecimiento exponencial del espacio produce una disminución

en la tasa de avance de la luz en coordenadas comóviles, lo que da lugar a la aparición de un horizonte cosmológico. Esto se debe a que el espacio está en expansión.

Por otro lado, el valor “real” del horizonte cosmológico es aproximadamente 46500 millones de años luz, es decir, unos 14300 Mpc (Illana, 2024, pág 12). La diferencia entre el valor obtenido en la simulación y el valor observacional se debe a que la simulación representa el horizonte de eventos en un universo de de Sitter, mientras que el valor mayor corresponde al horizonte de partículas del universo real, el cual depende de la historia completa de expansión y no únicamente de la fase dominada por la constante cosmológica.

### 3.4. Constante cosmológica

La constante cosmológica fue introducida por Einstein debido a la necesidad de introducir un término que hiciera posible la modelación de sus ecuaciones para obtener un universo estático con presión y curvatura positiva. “El universo estático es, entonces, análogo a las órbitas circulares y, para energías positivas, dichas órbitas son inestables” (Balaguera, 2006)[pág 1]. El descubrimiento del corrimiento al rojo desmontó toda esta suposición de un universo estático. En la actualidad la constante cosmológica se usa en el campo de la cosmología para explicar la dinámica del universo y la densidad total de energía en él.

Esta constante también puede ser definida como una variación del tensor de energía-momento usando la ecuación 2.35 que se trae de nuevo para efectos prácticos.

$$T_{\mu\nu} = (\rho_\Lambda + P) V_\mu V_\nu - P g_{\mu\nu}$$

Donde se llega a que  $P = -\rho$ . lo que significa que el tensor energía momento en el vacío está dado por

$$T_{\mu\nu}^{\text{vacío}} = -\rho_\Lambda g_{\mu\nu} \quad (3.1)$$

Ahora el tensor energía está compuesto por dos términos, uno que está definido por la densidad de energía de radiación y otro por la densidad de energía en el vacío  $\rho_{\text{vacío}}$  que está definido como se vio anteriormente en 2.7 de la forma (Liddle, 2015, pág 56).

$$\rho_\Lambda = \frac{\Lambda c^2}{8\pi G} \quad (3.2)$$

Para dar cuenta de la relación entre la expansión acelerada del universo y la constante cosmológica, se hizo uso de la relación 2.40 y con base en ello, se realizó el código 3.8 que diferencia el compromiso y la importancia de los valores positivos de  $\Lambda$  en el modelo de de Sitter.

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation
from IPython.display import HTML

# Tiempo
t = np.linspace(0, 8, 300)
dt = t[1] - t[0]

H0 = 1.0

# Inicialización
a_m = np.zeros(len(t)) # materia
a_v = np.zeros(len(t)) # vacío

a_m[0] = 0.01
a_v[0] = 0.01

# Evolución
for i in range(1, len(t)):
    # Materia:  $\rho \sim a^{-3}$ 
    H_m = H0 * np.sqrt(1 / a_m[i-1]**3)
    a_m[i] = a_m[i-1] + H_m * a_m[i-1] * dt

    # Vacío: H constante
    H_v = H0
    a_v[i] = a_v[i-1] + H_v * a_v[i-1] * dt
```

```

# Figura
fig, ax = plt.subplots()

line_m, = ax.plot([], [], label="Materia")
line_v, = ax.plot([], [], label="Vacío (de Sitter)")

ax.set_xlim(0, 8)
ax.set_ylim(0, max(a_v)*1.1)

ax.set_xlabel("Tiempo [ $H_0^{-1}$ ]\")
ax.set_ylabel("Factor de escala  $a(t)$ \")

ax.legend()
ax.grid()

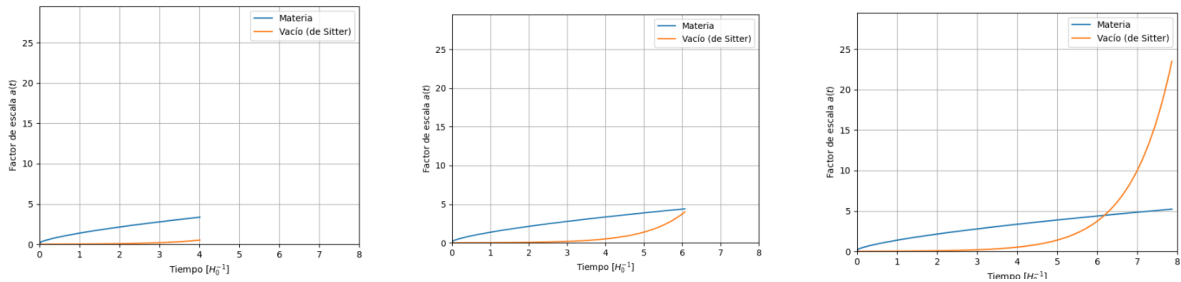
# Animación
def update(frame):
    line_m.set_data(t[:frame], a_m[:frame])
    line_v.set_data(t[:frame], a_v[:frame])
    return line_m, line_v

ani = FuncAnimation(fig, update, frames=len(t), interval=30)
HTML(ani.to_jshtml())

```

Figura 3.8: Código comparación densidad del vacío

El código muestra cómo cambia el factor de escala  $a(t)$  con respecto a la constante cosmológica y su relación con la densidad del vacío, comparándose con la densidad de la materia.



Cuadro 3.1: Gráfica factor de escala en el vacío vs materia. Elaboración propia

También se realizó el siguiente código 3.9 que muestra la implicación de una constante cosmológica positiva, en comparación con una constante cosmológica equilibrada y una negativa.

```

import numpy as np
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation
from IPython.display import HTML

# Tiempo
t_vals = np.linspace(0.1, 6, 150)
dt = t_vals[1] - t_vals[0]

# Posiciones iniciales
x0 = np.linspace(0.5, 3, 12)

# Constante cosmológica
Lambda_vals = [0.0, 1.0, -1.0]

fig, ax = plt.subplots()
ax.set_xlim(0, 15)
ax.set_ylim(-1, 4)

ax.set_xlabel("Distancia física")
ax.set_ylabel("Tiempo")

einstein_pts, = ax.plot([], [], 'bo', label="Einstein ( $\Lambda$  equilibrada)")
desitter_pts, = ax.plot([], [], 'ro', label="de Sitter ( $\Lambda > 0$ ")
ads_pts, = ax.plot([], [], 'mo', label=" $\Lambda < 0$ ")

# Líneas de luz
luz_e, = ax.plot([], [], 'b-', alpha=0.4)
luz_d, = ax.plot([], [], 'r-', alpha=0.4)
luz_ads, = ax.plot([], [], 'm-', alpha=0.4)

legend = ax.legend(loc="upper right", fontsize=9, framealpha=0.9)
for text in legend.get_texts():
    text.set_multialignment('left')

```

```

# Factor de escala
def escala(L, t):
    if L > 0:
        H = np.sqrt(L / 3)
        return np.exp(H * t)
    elif L == 0:
        return 1.0
    else:
        H = np.sqrt(abs(L) / 3)
        return np.cos(H * t)

# Luz (geodésica)
def luz_geodesica(a_vals):
    return np.cumsum(1 / a_vals) * dt

# Animación
def update(frame):
    t = t_vals[frame]

    objetos = []

    for i, L in enumerate(Lambda_vals):

        # Historia del factor de escala
        a_hist = np.array([escala(L, ti) for ti in t_vals[:frame+1]])
        a_hist = np.clip(a_hist, 0.01, None)
        a = a_hist[-1]

        # Posiciones
        x = a * x0
        y = np.ones_like(x) * (2 - i)

        # Asignar a cada modelo
        if L == 0:
            einstein_pts.set_data(x, y)
            objetos.append(einstein_pts)

        elif L > 0:
            desitter_pts.set_data(x, y)
            objetos.append(desitter_pts)

```

---

```

    # Horizonte
    H = np.sqrt(L / 3)
    r_h = 1 / H
    ax.axvline(r_h, color='gray', linestyle='--', alpha=0.2)

else:
    ads_pts.set_data(x, y)
    objetos.append(ads_pts)

# Luz
r = luz_geodesica(a_hist)
t_hist = t_vals[:frame+1]

if L == 0:
    luz_e.set_data(r, t_hist)
    objetos.append(luz_e)

elif L > 0:
    luz_d.set_data(r, t_hist)
    objetos.append(luz_d)

else:
    luz_ads.set_data(r, t_hist)
    objetos.append(luz_ads)

return objetos

ani = FuncAnimation(fig, update, frames=len(t_vals), interval=50)

HTML(ani.to_jshtml())

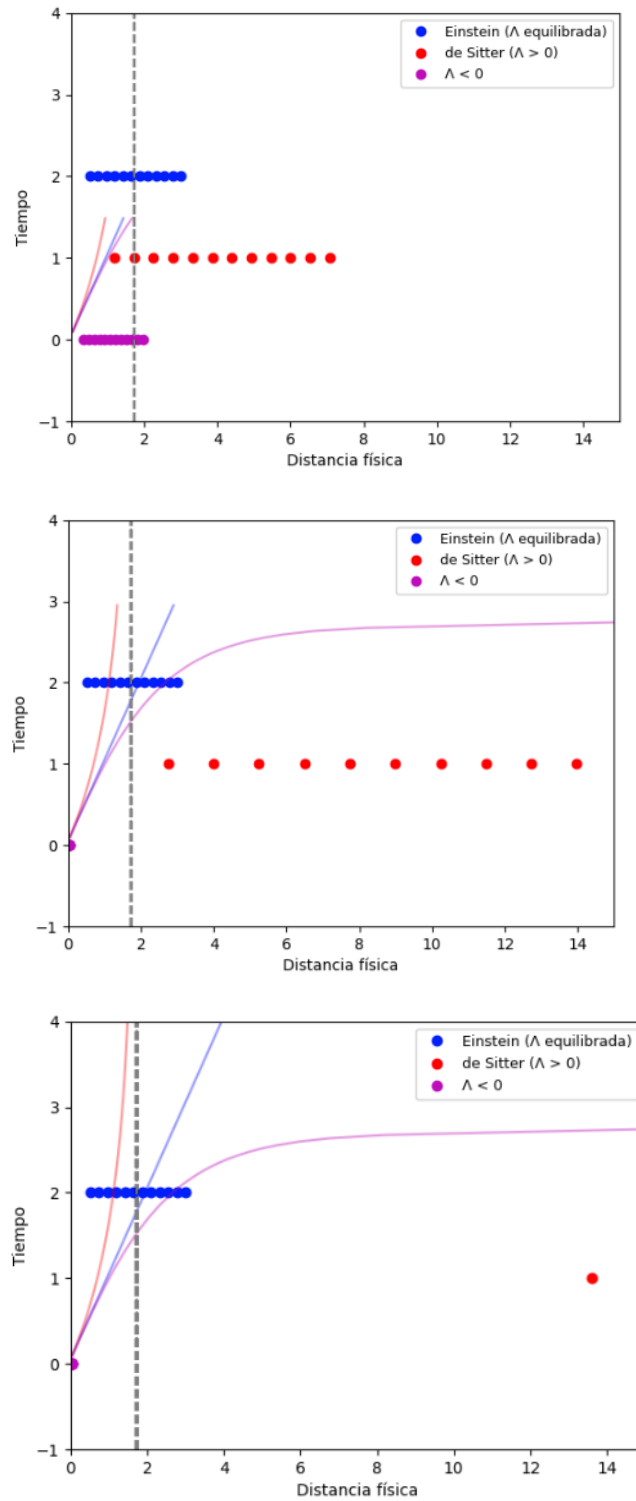
```

---

Figura 3.9: Código implicación de  $\Lambda$ . Elaboración propia

La simulación permite visualizar que la constante cosmológica no solo determina la dinámica de expansión del universo, sino que define su estructura causal estableciendo condiciones bajo las cuales la información puede o no propagarse entre distintas regiones del espacio-tiempo. Para un universo estático siempre hay conexión, no existe un horizonte. Ahora, para el universo de de Sitter existe un horizonte, por ende la luz es asíntota con el horizonte. Por otro lado, para una valor de la constante cosmológica negativa, implica que regiones que en algún momento pudieron haber estado desconectadas, pueden

volver a entrar en contacto, ya que la contracción del espacio-tiempo favorece la reconexión (Wald, 1984). Tener en cuenta que no se profundizará en los fenómenos distintos a los del universo de de Sitter.



Cuadro 3.2: Visualización compromiso de  $\Lambda$ . Elaboración propia

El código ejecuta una simulación en donde se evidencia cómo la causalidad se ve afectada según el valor de la constante cosmológica como se aprecia en la gráfica 3.2.

### 3.5. Distancia entre partículas

La distancia entre dos puntos es uno de los fenómenos que permite dar cuenta de la expansión acelerada del universo; además de ser un gran ejemplo para evidenciar cómo la distancia física en un universo en expansión es proporcional al factor de escala y dependiente del tiempo, lo que permite diferenciarla de las distancias comóviles (Dodelson, 2003). La expresión que determina estas distancias está representada en 2.5, ya que como se explicó, para un universo en expansión las distancias físicas dependen del factor de escala  $a(t)$ .

$$d(t) = x_{comovil} \cdot a(t)$$

#### 3.5.1. Distancia física

Para visualizar la distancia entre partículas en el universo de de Sitter, se tuvo en cuenta la ecuación 2.5 que define la distancia física en términos de la distancia comóvil. Y se realizó el código que aparece en la Figura 3.10.

#### 3.5.2. Distancia comóvil

La distancia comóvil, como se definió anteriormente, es la distancia independiente del tiempo, usada en modelos inflacionarios del espacio-tiempo, por lo cuál las coordenadas de los cuerpos siempre será la misma. El siguiente código permite al lector definir las coordenadas de los objetos en el espacio.

```

import sympy as sp
import numpy as np
import matplotlib.pyplot as plt

#Conversión de unidades

def hubble_km_s_Mpc_to_si(H_km_s_Mpc):
    """
    Convierte H de km/s/Mpc a 1/s (SI)
    """
    Mpc_in_m = 3.0857e22
    km_in_m = 1000
    return (H_km_s_Mpc * km_in_m) / Mpc_in_m

# Variables simbólicas

t = sp.Symbol('t', real=True)
H = sp.Symbol('H', real=True, positive=True)

# Factor de escala (de Sitter)
a = sp.exp(H * t)

# Coordenadas
x1, y1, z1 = sp.symbols('x1 y1 z1', real=True)
x2, y2, z2 = sp.symbols('x2 y2 z2', real=True)

# Distancia comóvil (en metros)
d_comovil = sp.sqrt((x1 - x2)**2 + (y1 - y2)**2 + (z1 - z2)**2)

# Distancia física
D = sp.simplify(a * d_comovil)

```

```

#Evaluación numérica

def distancia_fisica(H_val_SI, t_val_s, punto1, punto2):
    subs = {
        H: H_val_SI,
        t: t_val_s,
        x1: punto1[0], y1: punto1[1], z1: punto1[2],
        x2: punto2[0], y2: punto2[1], z2: punto2[2],
    }
    return float(D.subs(subs))

if __name__ == "__main__":

    print("\n--- Universo de de Sitter ---")
    print("Las coordenadas deben ingresarse en METROS.")
    print("El tiempo debe ingresarse en SEGUNDOS.")
    print("La constante de Hubble debe ingresarse en km/s/Mpc.\n")

    # Ingreso de H en unidades astronómicas estándar
    H_input_km = float(input("Ingrese H en km/s/Mpc (ej: 70): "))
    H_input_SI = hubble_km_s_Mpc_to_si(H_input_km)

    print(f"H convertido a unidades SI: {H_input_SI:.3e} s-1\n")

    # Tiempo cósmico en segundos
    t_input = float(input("Ingrese el tiempo cósmico en segundos: "))

```

```

print("\Ingrese las coordenadas del cuerpo 1 en metros:")

punto1 = [
    float(input("x1 (m): ")),
    float(input("y1 (m): ")),
    float(input("z1 (m): "))
]

print("\Ingrese las coordenadas del cuerpo 2 en metros:")

punto2 = [
    float(input("x2 (m): ")),
    float(input("y2 (m): ")),
    float(input("z2 (m): "))
]

# Calcular distancia
D_val = distancia_fisica(H_input_SI, t_input, punto1, punto2)

print(f"\Distancia física: {D_val:.3e} metros")

# Evolución temporal con dos puntos señalados
t_futuro = 1e17
t_final = t_input + t_futuro

t_vals = np.linspace(t_input, t_final, 300)

D_vals = [distancia_fisica(H_input_SI, t_val, punto1, punto2)
           for t_val in t_vals]

# Distancia en el tiempo dado
D_inicial = distancia_fisica(H_input_SI, t_input, punto1, punto2)

```

```

# Distancia en el tiempo futuro
D_final = distancia_fisica(H_input_SI, t_final, punto1, punto2)
print(f"Distancia física pasados 3.17 mil millones de años: {D_final:.3e} metros")

plt.figure(figsize=(8,5))
plt.plot(t_vals, D_vals, label="Distancia física  $D(t)$ ")

# Punto en tiempo inicial
plt.scatter(t_input, D_inicial)
plt.annotate(
    f"D( $t_0$ ) = {D_inicial:.3e} m",
    (t_input, D_inicial),
    textcoords="offset points",
    xytext=(10,10)
)

# Punto en tiempo futuro
plt.scatter(t_final, D_final)
plt.annotate(
    f"D( $t_f$ ) = {D_final:.3e} m",
    (t_final, D_final),
    textcoords="offset points",
    xytext=(-80,-20)
)

plt.axvline(x=t_input, linestyle=":")
plt.axvline(x=t_final, linestyle=":")

plt.xlabel("Tiempo cósmico (s)")
plt.ylabel("Distancia física (m)")
plt.title("Evolución de la distancia física en de Sitter")
plt.legend()
plt.grid(True)
plt.show()

```

Figura 3.10: Código de distancia entre puntos. Elaboración propia

En donde se aplican todas las funciones y métodos anteriormente vistos a lo largo del trabajo. Por otro lado este código está explicado en el apéndice E.

```

\ Universo de de Sitter
Las coordenadas deben ingresarse en METROS.
El tiempo debe ingresarse en SEGUNDOS.
La constante de Hubble debe ingresarse en km/s/Mpc.

Ingrese H en km/s/Mpc (ej: 70): 70
H convertido a unidades SI: 2.269e-18 s^-1

Ingrese el tiempo cósmico en segundos: 60
\Ingrese las coordenadas del cuerpo 1 en metros:
x1 (m): 0
y1 (m): 0
z1 (m): 0
\Ingrese las coordenadas del cuerpo 2 en metros:
x2 (m): 1
y2 (m): 0
z2 (m): 0
\Distancia física: 1.000e+00 metros
Distancia física pasados 3.17 mil millones de años: 1.255e+00 metros

```

Figura 3.11: Datos de ejemplo de distancias físicas. Elaboración propia

En la Figura 3.11 se presentan los datos que el lector haya elegido para comparar las distancias. En el ejemplo se graficó el cuerpo 1 ubicado en el origen y el cuerpo 2 con coordenadas  $x=1$  y  $z=y=0$ . Al ejecutar, el código permite variar los parámetros de la constante de Hubble y el tiempo además de las coordenadas.

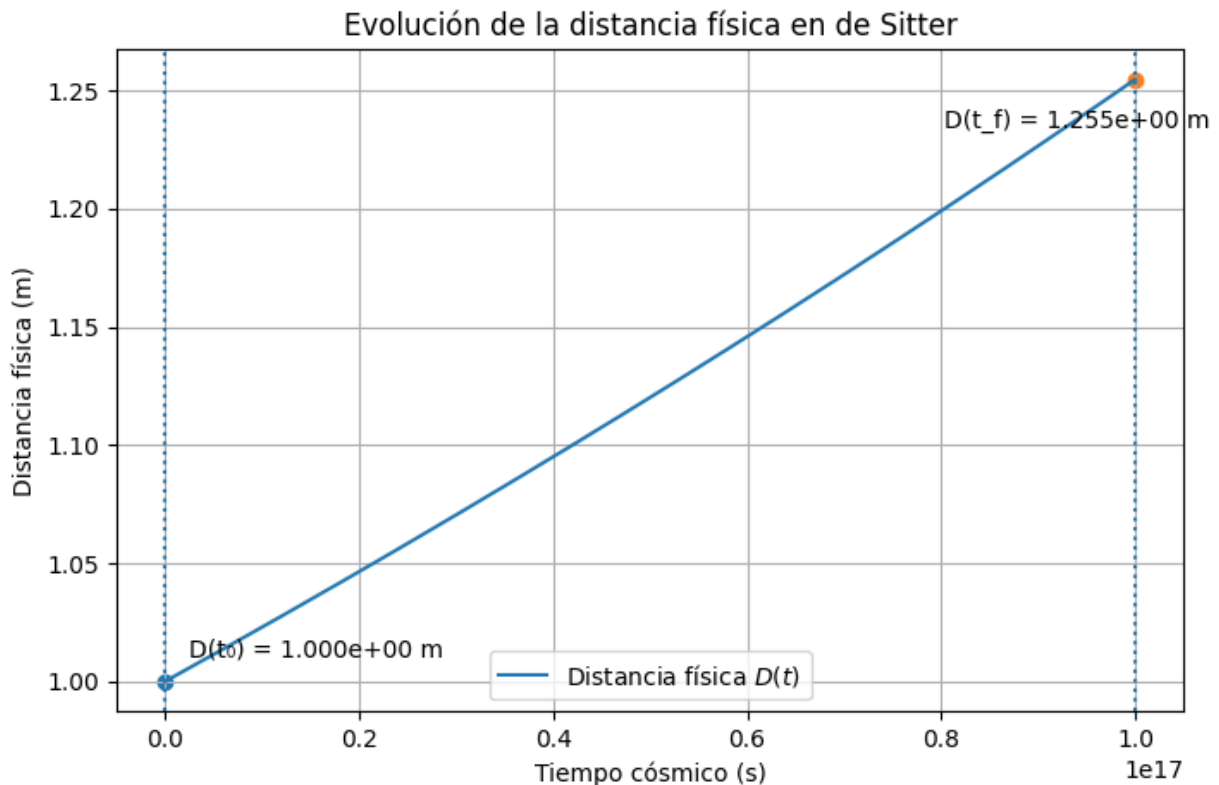


Figura 3.12: Distancia entre puntos

El programa tiene la capacidad de distinguir entre los dos puntos, graficando la distancia que existe entre dos partículas en el espacio en un punto inicial, dado por las condiciones que el lector elija, y la distancia futura pasados unos 3.17 mil millones de años; lo que permite apreciar lo imperceptible que se hace la expansión en tiempos tempranos como se muestra en la Figura 3.12.

### 3.6. Simulación constante de Hubble

Para el código del parámetro de Hubble, se le da la opción al lector de modificar a su antojo el valor de dicho parámetro; esto ayuda a dar cuenta de los ajustes y de los valores cercanos a la constante de Hubble con respecto a las evidencias científicas.

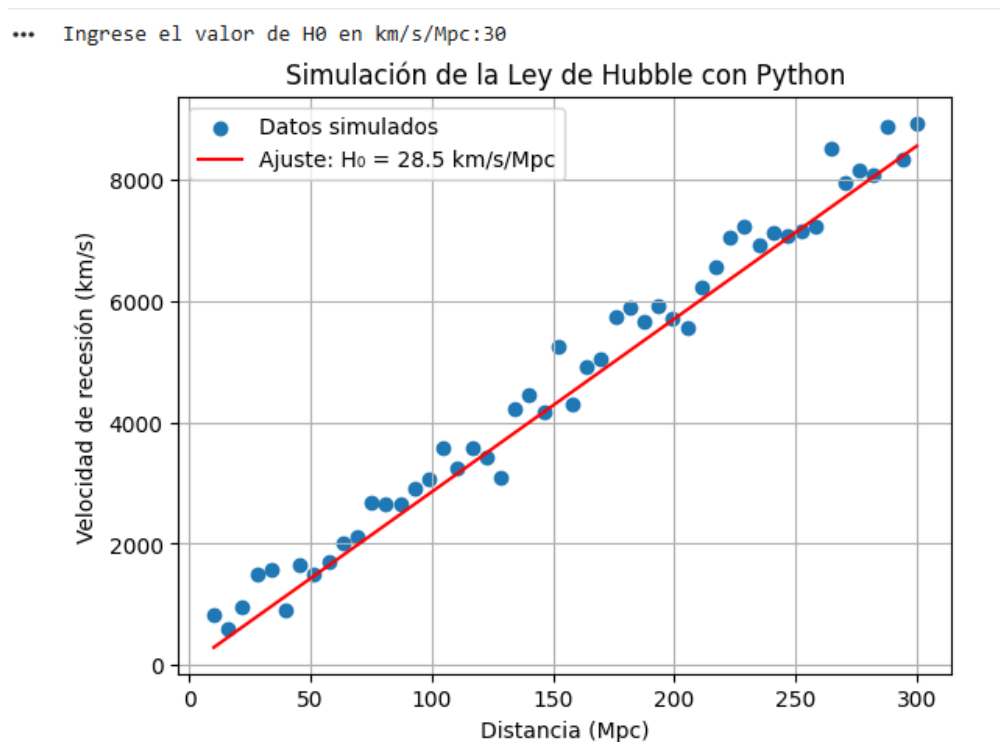


Figura 3.13: Gráfica de la constante de Hubble

Como se puede apreciar en la figura 3.13, al ejemplo con el que se ejecutó el programa se le da un valor alejado del valor de la constante actual; por ende, se logra evidenciar una gran dispersión. Se vuelve a enseñar el producto del código 2.1 para efectos prácticos y comodidad.

# Conclusiones

La intención del diseño de los códigos, además de dar un punto de partida distinto al formalismo matemático es minimizar el procedimiento cuantitativo para permitir que a quien lea el trabajo se pueda enfocar en el razonamiento cualitativo de la relatividad general. Por lo tanto, a lo largo del trabajo se concluye que es útil el uso de herramientas computacionales ya que, en primera instancia, optimiza el tiempo de los procesos matemáticos que exigen calcular cambios de coordenadas, constantes o simplemente expresiones que requieren desarrollos extensos y rigurosos, lo cual le permite al lector enfocarse, en mayor medida, al estudio del fenómeno físico en cuestión.

Por otro lado, es importante señalar que con el desarrollo de los códigos también se pretende no saltarse del todo el formalismo matemático, ya que es este mismo la base de los códigos que allí aparecen, y entender el funcionamiento del código es entender el tratamiento matemático de fondo. Además, se requiere un análisis conceptual en cuanto a la comprensión de las gráficas que varios códigos muestran al ejecutarse y, como concluye Castillo et al. (2025), esto resulta en una aplicación no solo para la relatividad general, sino también para estudios de física teórica contemporáneos.

El lector en el trabajo encuentra librerías útiles para otros tipos de aplicaciones diferentes a las que aquí se proponen. Esto permite ampliar y reconocer los alcances del lenguaje Python, importante en una era en la que ha tomado fuerza la programación (Prado Palacios et al., 2023).

El estudio de temas cosmológicos, como propone Castañeda-Zapata et al. (2024), puede ser una excelente fuente de motivación para el aprendizaje de las ciencias para los estudiantes de educación media y superior.

Se evidenciaron desventajas con respecto a la implementación de ciertos códigos, dado que a lo largo del desarrollo del trabajo se experimentaron conflictos en cuanto al posible significado físico erróneo que podrían inducir los resultados de estos, y de cómo ello conllevaba al conflicto entre decidir si construir códigos fieles al universo de de Sitter o códigos prácticos para

visualizar los fenómenos que se pretendían mostrar. En la expansión acelerada del universo, que bajo el modelo de de Sitter exponencial, las dimensiones tienden a ser enormes, lo que indujo a distintos arreglos en los códigos para priorizar la representación y no la simulación de un universo real.

Por otra parte, el trabajo busca, de cierta forma, acortar la brecha entre los procesos matemáticos extensos, y de acuerdo con Horvath y Moraes (2021), quienes señalan que prescindir del rigor matemático en la instrucción de la relatividad general puede debilitar la conexión de sus fundamentos teóricos con otras áreas de la física. En ese sentido, si bien el trabajo está enfocado en ser un aporte para la comprensión conceptual de la relatividad general, podría no ser suficiente para desarrollar una comprensión más profunda de la estructura matemática implícita, lo que limita su alcance en contextos de formación más avanzados.

# Bibliografía

- Afranchi, L. (2022). *Aplicación de los Símbolos de Christoffel en la Teoría Relativista* [Tesis de Licenciatura]. Universidad de Panamá, Facultad de Ciencias Naturales, Exactas y Tecnología, Escuela de Física [Profesor Asesor: Miguel Silvera].
- Ángel Franco García. (2016). La ecuación de Schrödinger en coordenadas esféricas [Curso de Física por Ordenador. Universidad del País Vasco (EHU). Accedido el: 25 de mayo de 2026].
- Avalos, E. P. (2000, julio). *Aplicación del formalismo lagrangiano ADM a un modelo cosmológico* [Trabajo de graduación]. Universidad de San Carlos de Guatemala, Facultad de Ingeniería, Escuela de Ciencias.
- Ayala, M. M. (2007). Consideraciones sobre la formalización y matematización de los fenómenos físicos. *Praxis Filosófica*, (25), 39-54.
- Balaguera, A. (2006). *Cosmological and astrophysical structures in the Einstein-de Sitter universe* [Tesis doctoral, Universidad de los Andes] [Disponible en: <https://hdl.handle.net/1992/22701>].
- Berzosa Montero, G. (2024). *La curvatura de Ricci* [Trabajo Fin de Grado]. Universitat d'Alacant, Facultad de Ciencias [Grado en Matemáticas. Curso académico 2023-2024].
- Bikwa, O., Melia, F., & Shevchuk, A. (2011). Photon Geodesics in FRW Cosmologies. *Monthly Notices of the Royal Astronomical Society*. <https://arxiv.org/abs/1112.4774>
- Campo, T. (2022). *Análisis geométrico del modelo cosmológico de de Sitter*.
- Carroll, S. M. (2004). *Spacetime and Geometry: An Introduction to General Relativity*. Addison-Wesley.
- Castañeda-Zapata, E. A., López-Ríos, S. Y., & Osorio-Vélez, J. A. (2024). La relatividad general en la educación básica y media: una revisión sistemática. *Revista Eureka sobre Enseñanza y Divulgación de las Ciencias*, 21(2), 2303. [https://doi.org/10.25267/Rev\\_Eureka.ensen\\_divulg\\_cienc.2024.v21.i2.2303](https://doi.org/10.25267/Rev_Eureka.ensen_divulg_cienc.2024.v21.i2.2303)
- Castiblanco, O., & Nardi, R. (2023, julio). *Didáctica de la Física* [Versión digital disponible en ResearchGate]. Cultura Académica Digital. <https://www.researchgate.net/publication/372337117>
- Castillo, E., Palacios, L., & Soto, C. (2025). Herramienta computacional en Python para cálculos de relatividad general. *Universidad Autónoma del Estado de Hidalgo*, 13, 24-31.
- Coles, P., & Lucchin, F. (2002). *Cosmology: The Origin and Evolution of Cosmic Structure* (2.<sup>a</sup> ed.). John Wiley & Sons.
- Dodelson, S. (2003). *Modern Cosmology*. Academic Press.
- Fleisch, D. A. (2011). *A Student's Guide to Vectors and Tensors*. Cambridge University Press. <https://www.cambridge.org/highereducation/books/a-students-guide-to-vectors-and-tensors/CC1736FAE454A3BA2B56FDBBBCECF267C>

- González Álvarez, J. (s.f.). El universo en expansión y el modelo de Sitter [Documento de divulgación científica, no publicado formalmente].
- Gürses, M., & Heydarzade, Y. (2020). FLRW-Cosmology in Generic Gravity Theories. *arXiv preprint arXiv:2009.12825*. <https://arxiv.org/abs/2009.12825>
- Harrison, E. (2000). *Cosmology: The Science of the Universe* (2.<sup>a</sup> ed.). Cambridge University Press.
- Henriksen, E. K., Bungum, B., Angell, C., Tellefsen, C. W., Frågåt, T., & Bøe, M. V. (2014). Relativity, Quantum Physics and Philosophy in the Upper Secondary Curriculum: Challenges, Opportunities and Proposed Approaches. *Physics Education*, 49(6), 678-684. <https://doi.org/10.1088/0031-9120/49/6/678>
- Hernández, R., Fernández, C., & Baptista, P. (2014). *Metodología de la investigación*. McGraw Hill España.
- Horvath, J. E., & Moraes, P. H. R. S. (2021). Should we teach general relativity in high school? Why and how? *Astronomy Education Journal*, 1(1), 49-51. <https://doi.org/10.32374/AEJ.2021.1.1.008>
- Illana, J. I. (2024). Distancias y tiempos en el cosmos. *NaDiR: Revista de la Asociación para la Enseñanza de la Astronomía*, (58), 3-21.
- Janssen, B. (2013). *Teoría de la Relatividad General*. Universidad de Granada.
- Kenyon, I. R. (2023). *Introduction to General Relativity and Cosmology* (2nd). IOP Publishing. <https://doi.org/10.1088/2514-3433/acc3ff>
- Kersting, M., Henriksen, E. K., Bøe, M. V., & Angell, C. (2018). General Relativity in Upper Secondary School: Design and Evaluation of an Online Learning Environment Using the Model of Educational Reconstruction. *Physical Review Physics Education Research*, 14(1), 010130. <https://doi.org/10.1103/PhysRevPhysEducRes.14.010130>
- Kersting, M., Toellner, R., Blair, D., & Burman, R. (2020). Gravity and Warped Time—Clarifying Conceptual Confusions in General Relativity. *Physics Education*, 55(1), 015023. <https://doi.org/10.1088/1361-6552/ab56d7>
- Klein, G. (2012, mayo). *Didáctica de la Física* [Material de apoyo docente, recopilación de artículos y documentos para formación en profesorado de Física]. file:///F:/DidF%C3%ADsica.docx
- Kraus, U., & Zahn, C. (2018). Sector Models—A Toolkit for Teaching General Relativity: III. Spacetime Geodesics. *European Journal of Physics*, 39(5), 056016. <https://doi.org/10.1088/1361-6404/aac8f7>
- Landau, L. D., & Lifshitz, E. M. (1975). *The Classical Theory of Fields* (4th, Vol. 2). Pergamon Press.
- Lee, J. M. (2018). *Introduction to Riemannian Manifolds* (2.<sup>a</sup> ed., Vol. 176). Springer. <https://doi.org/10.1007/978-3-319-91755-9>
- Liddle, A. R. (2015). *An Introduction to Modern Cosmology* (3.<sup>a</sup> ed.). John Wiley & Sons.
- Misner, C. W., Thorne, K. S., & Wheeler, J. A. (1973). *Gravitation*. W. H. Freeman; Company.
- Mukhanov, V. (2005). *Physical Foundations of Cosmology*. Cambridge University Press.
- Nakahara, M. (2003). *Geometry, Topology and Physics* (2.<sup>a</sup> ed.). Institute of Physics Publishing.
- Niels, O., & Saavedra, T. (2020). *Tensores y ecuaciones de Einstein para la métrica de Friedmann-Lemaître-Robertson-Walker* (inf. téc.). Universidad Mayor de San Simón.

- Pascu, G. (2024). Atlas of Coordinate Charts on the de Sitter Spacetime [PhD report, unpublished manuscript].
- Peebles, P. J. E., & Ratra, B. (2003). The Cosmological Constant and Dark Energy. *Reviews of Modern Physics*, 75, 559-606. <https://doi.org/10.1103/RevModPhys.75.559>
- Pérez, H., & Solbes, J. (2003). Algunos problemas en la enseñanza de la relatividad. *Revista de Investigación y Experiencias Didácticas*, 21(1), 135-146.
- Poisson, E. (2004). *A Relativist's Toolkit: The Mathematics of Black-Hole Mechanics*. Cambridge University Press.
- Prado Palacios, K. F., Carvajal Rabelo, D. A., & Robles Pico, C. O. (2023). La importancia de la programación en la actualidad. *Convicciones*, 10(20), 187-195. [est\\_kf\\_prado@fesc.edu.co](mailto:est_kf_prado@fesc.edu.co)
- Ryden, B. (2006). *Introduction to Cosmology*. Addison-Wesley.
- Sánchez, J. (1985). Problemas ontológicos de la física: interacción y dimensionalidad del espacio. *Universidad Autónoma de Madrid*, 15, 1-2.
- Schutz, B. F. (2009). *A First Course in General Relativity* (Second Edition). Cambridge University Press. [www.cambridge.org/9780521887052](http://www.cambridge.org/9780521887052)
- Serrano González, B., Cuesta, A. J., & Ortiz Mora, A. (2019). Determinación de la constante de Hubble mediante observaciones del Universo Local [Trabajo académico, Universidad de Córdoba]. *Revista de Física*.
- Torres Cañizález, P. C., & Cobo Beltrán, J. K. (2017). Tecnología educativa y su papel en el logro de los fines de la educación. *Educere*, 21(68), 31-40. <http://www.redalyc.org/articulo.oa?id=35652744004>
- Villegas, F. (2019). Espacio-tiempo Anti-de Sitter (AdS) [Recibido: 30 noviembre 2019; Aceptado: 27 diciembre 2019]. *Revista de Investigación de Física*, 22(2), 19-24.
- Wald, R. M. (1984). *General Relativity*. University of Chicago Press.
- Weinberg, S. (1972). *Gravitation and cosmology: Principles and applications of the general theory of relativity*. John Wiley & Sons.
- Weinberg, S. (2008). *Cosmology*. Oxford University Press.
- Zahn, C., & Kraus, U. (2014). Sector Models—A Toolkit for Teaching General Relativity. Part 1: Curved Spaces and Spacetimes. *European Journal of Physics*, 35(5), 055020. <https://doi.org/10.1088/0143-0807/35/5/055020>

# Apéndice A

## Código constante de Hubble

Se define **H0\_real = float**. Esta variable representa el valor “real” del parámetro de Hubble en unidades de  $km/s/Mpc$  que el lector elija. En el contexto físico, este número expresa cuántos kilómetros por segundo aumenta la velocidad de recesión por cada megapársec de distancia. La línea **distancias = np.linspace(10, 300, 50)** crea un arreglo de 50 valores igualmente espaciados entre 10 y 300. La función **linspace** significa que genera números distribuidos uniformemente en un intervalo. Aquí se están simulando distancias de galaxias medidas en megapársecs. Luego aparece **np.random.seed(0)**. Esta instrucción fija la “semilla” del generador de números aleatorios. Esto significa que cada vez que se ejecute el código, el ruido generado será el mismo. Es una forma de hacer que la simulación sea reproducible. La siguiente línea construye las velocidades **velocidades = H0\_real\*distancias + np.random.normal(0, 300, len(distancias))** Aquí ocurre algo importante. Primero se calcula la relación ideal de la Ley de Hubble: **velocidad = H0\_real distancia**. Eso daría una línea perfectamente recta. Sin embargo, en observaciones reales siempre hay dispersión en los datos. Por eso se añade **np.random.normal(0, 300, len(distancias))**, que genera ruido aleatorio con distribución normal (gaussiana), media 0 y desviación estándar 300  $km/s$ . El tercer argumento indica cuántos valores aleatorios se generan, uno por cada distancia. El resultado es un conjunto de datos que simula mediciones reales con dispersión observacional. Luego se realiza un ajuste lineal con **coeficientes = np.polyfit(distancias, velocidades, 1)**. La función **polyfit** ajusta un polinomio a los datos mediante mínimos cuadrados. El último argumento, 1, indica que se desea un polinomio de grado 1, es decir, una recta. El resultado es un arreglo con dos coeficientes: la pendiente y la ordenada al origen. La línea **H0\_ajustado = coeficientes[0]** toma la pendiente de esa recta ajustada. En el contexto físico, esa pendiente es precisamente una es-

timación del parámetro de Hubble obtenida a partir de los datos simulados. Después comienza la parte de visualización. Con **plt.scatter(distancias, velocidades, label="Datos simulados")** se crea un gráfico de dispersión, donde cada punto representa una galaxia con su distancia y velocidad. La línea **plt.plot(distancias, H0\_ajustado\*distancias, color='r', label=f.^ajuste:  $H_0 = \text{H0\_ajustado:.1f km/s/Mpc}$ ")** dibuja la recta correspondiente al ajuste lineal. Se multiplica la pendiente ajustada por las distancias para construir los valores de la recta. El parámetro **color='r'** indica que la línea será roja. Luego se agregan etiquetas a los ejes con **plt.xlabel** y **plt.ylabel**, se define el título del gráfico con **plt.title**, se añade la leyenda con **plt.legend()** y se activa la cuadrícula con **plt.grid(True)**. En conjunto, el código hace una simulación sencilla de la Ley de Hubble: genera distancias, calcula velocidades siguiendo una relación lineal con ruido observacional, ajusta una recta por mínimos cuadrados y recupera una estimación del parámetro de Hubble a partir de los datos. Es una manera computacional de reproducir, en versión simplificada, el procedimiento que permitió establecer la expansión del universo.

## Apéndice B

### Solución ecuación diferencial 2.11

Se tiene una ecuación diferencial de segundo orden la cual se puede resolver por el método de coeficientes constantes.

$$\ddot{a} - aH^2 = 0 \quad (\text{B.1})$$

Se propone una solución de tipo exponencial

$$a = e^{rt} \quad (\text{B.2})$$

Se hallan sus derivadas

$$\dot{a} = re^{rt} \quad (\text{B.3})$$

$$\ddot{a} = r^2 e^{rt} \quad (\text{B.4})$$

Se reemplaza en la ecuación 2.11 y se factoriza el exponencial  $e^{rt}$ .

$$e^{rt} [r^2 - H^2] = 0 \quad (\text{B.5})$$

El factor exponencial debe ser diferente de cero para no tener una solución trivial. Entonces se tiene que  $r = H$ . Se expresa la solución general que está dada por la expresión B.6

$$a(t) = Ae^{+rt} + Be^{-rt} \quad (\text{B.6})$$

Donde A y B son constantes que se hallan con las condiciones iniciales. Y reemplazando r por H se obtiene la solución general de la ecuación diferencial.

$$a(t) = Ae^{+Ht} + Be^{-Ht} \quad (\text{B.7})$$

## Apéndice C

### Solución de los factores para obtener 2.17

Para este caso la expresión 2.13 queda expresada de la siguiente manera:

$$AB = \frac{c^2}{4H^2} \quad (\text{C.1})$$

Es decir que el producto entre A y B debe resultar en C.1 a lo que conviene parametrizar A y B de la siguiente forma:

$$A = \frac{c}{2H} e^{Ht_0} \quad (\text{C.2})$$

$$B = \frac{c}{2H} e^{-Ht_0} \quad (\text{C.3})$$

Reemplazando en 2.12 se obtiene la expresión del factor de escala para un universo esférico.

$$a(t) = \frac{c}{H} \cosh (H(t - t_0)) \quad (\text{C.4})$$

## Apéndice D

### Solución de los factores para obtener 2.23

En una geometría de un universo hiperbólico se usa el factor D.1.

$$AB = \frac{-c^2}{4H^2} \quad (\text{D.1})$$

Para la solución de este es similar a la de un universo esféricos, solo que A y B tienen signos opuestos, entonces se parametrizan de la siguiente forma.

$$A = \frac{c}{2H} e^{Ht_0} \quad (\text{D.2})$$

$$B = \frac{-c}{2H} e^{-Ht_0} \quad (\text{D.3})$$

Y se obtiene como resultado:

$$a(t) = \frac{c}{H} \sinh (H(t - t_0)) \quad (\text{D.4})$$

# Apéndice E

## Código distancia física

Primero se importa **Sympy**, que permite trabajar con expresiones matemáticas simbólicas; es decir, con variables algebraicas sin asignarles aún valores numéricos. Luego importa **NumPy**, que se utiliza para cálculos numéricos y para construir arreglos de datos. Finalmente importa el módulo **pyplot** de **Matplotlib**, que servirá para generar la gráfica de la evolución temporal de la distancia.

Después de las importaciones, el código define una función que tiene como objetivo convertir las unidades de Hubble de (km/s/Mpc) a unidades del Sistema Internacional (1/s) **hubble km s Mpc to si**. Dentro de la función se establecen dos factores de conversión, primero calcula cuántos metros tiene un megaparsec y luego, cuántos metros tiene un kilómetro. Con esos factores se realiza la conversión dimensional correspondiente y se devuelve el valor ya transformado.

Luego se define el tiempo cósmico **t** como un símbolo real usando **sp.Symbol**. También se define la constante de Hubble **H** como símbolo real y positivo.

Se construye el factor de escala del universo de de Sitter mediante la expresión **sp.exp(H \* t)** Esto representa matemáticamente  $a(t)=e^{Ht}$  que aparece en 2.16 ya que se hace el estudio desde una geometría plana.

Después se definen seis símbolos adicionales que representan las coordenadas espaciales de dos puntos distintos en el espacio comóvil: **x1, y1, z1, x2, y2, z2**. Con esas variables se construye la distancia comóvil utilizando la fórmula euclidiana estándar, implementada mediante la raíz cuadrada simbólica y potencias al cuadrado.

Una vez definida la distancia comóvil, el programa construye la distancia física multiplicándola por el factor de escala. Esto implementa directamente la relación cosmológica  $D(t)=a(t)*d_{comovil}$  como el la ecuación 2.5. La función **sp.simplify** se utiliza para dejar la expresión en su forma algebraica

más simple posible.

Posteriormente se define una función llamada **distancia fisica**. Esta función recibe valores numéricos concretos para la constante de Hubble, el tiempo y las coordenadas de ambos puntos. Dentro de la función se reemplaza cada símbolo por su valor correspondiente. Luego se utiliza el método **subs** para realizar la sustitución dentro de la expresión simbólica de la distancia física.

Finalmente, el resultado se convierte en un número decimal estándar mediante **float**, permitiendo usarlo en cálculos numéricos y gráficos.

El bloque **if \_\_name\_\_ == "\_\_main\_\_":** indica que lo que sigue solo se ejecutará si el archivo se corre directamente. Entonces primero se ingresa la constante de Hubble en unidades astronómicas y se convierte a unidades SI llamando a la función de conversión. Luego, se solicita el tiempo cósmico en segundos. Y por ultimo, se piden las coordenadas de los dos puntos en metros, que se almacenan como listas numéricas.

Con todos los datos disponibles, el programa llama a la función **distancia fisica** para calcular la distancia física en el instante dado y la imprime en notación científica.

Posteriormente, se construye la parte dinámica del programa definiendo un intervalo de tiempo hacia el futuro y se calculando el tiempo final sumando ese intervalo al tiempo inicial. Con **np.linspace** se generan 300 valores de tiempo igualmente espaciados entre el tiempo inicial y el final. Luego, se construye una lista de distancias físicas correspondientes a cada uno de esos tiempos utilizando una comprensión de lista, es decir, evaluando la función para cada instante del arreglo temporal. También se calculan explícitamente la distancia inicial y la distancia futura para destacarlas en la gráfica.

Finalmente, el programa entra en la fase de visualización con las funciones anteriormente vistas. Se añaden puntos destacados en los tiempos inicial y final usando **plt.scatter** y con **plt.annotate** se escriben etiquetas junto a esos puntos indicando el valor numérico de la distancia.

## Apéndice F

### Código de geometrías del espacio $k$

El programa importa las librerías necesarias, empezando con **NumPy** para cálculos numéricos, **Matplotlib** para graficar, y una función especial, distantes a las anteriores que ya se han trabajado, **FuncAnimation** para crear animaciones. También importa HTML para poder mostrar esa animación en un entorno tipo Jupyter Notebook.

Luego, imprime en pantalla las tres opciones de la geometría del espacio según el valor de  $k$ ; geometría esférica  $k = 1$ , plana  $k = 0$  e hiperbólica  $k = -1$ . Después le pide al usuario que elija una de esas geometrías. Ese valor se guarda en la variable  $k$ , que será clave porque define cómo será el universo que se va a dibujar.

Posteriormente, el código define algunos parámetros físicos. Introduce una constante  $c_s = 3e5$ , que representa el valor de la velocidad de la luz en el vacío en **km/s**. También define la constante de Hubble **H\_km\_s\_Mpc** = **70**, en sus unidades habituales de **km/s/Mpc**, y un parámetro  $A=1$ , que se usa en el caso plano como normalización del factor de escala  $a(t)$ . Después aparecen factores de conversión que pasan de kilómetros (**Km**) a megapársec (**Mpc**) **Mpc\_to\_km** y de segundos a años **year\_to\_s**. Con esto, el código intenta convertir la constante de Hubble a unidades de **1/año**. Lo hace dividiendo por los kilómetros de un **Mpc** para pasar de **km/s/Mpc** a **1/s** y luego multiplicando por los segundos de un año para pasar de **1/s** a **1/año**. Eso queda almacenado en la variable  $H$ . Luego hace algo similar con la velocidad de la luz multiplicando  $c_s$  por **year\_to\_s** para obtener  $c$ . La intención es tener  $c$  en **km/año**.

Después se construye el tiempo con la variable  $t$ , que es un arreglo de 60 valores que van desde 0 hasta 5 mil millones de años. Es decir, el código va a simular cómo evoluciona el universo en ese intervalo.

Con ello se crea una figura y un eje en 3D donde se va a dibujar la anima-

ción. Luego, la función **scale\_factor(t)** define cómo crece el universo en el tiempo según la geometría:

- Si  $k = 1$  usa una función coseno hiperbólico, lo que describe un universo esférico que nunca colapsa.
- Si  $k = -1$  usa un seno hiperbólico.
- Si  $k = 0$  usa una exponencial, que es el caso típico de de Sitter plano.

Esta función encapsula la física del problema que va a mostrar cómo evoluciona el tamaño del universo.

Después el código hace la parametrización de puntos (u y v) que sirven como coordenadas para dibujar superficies. Dependiendo de la geometría, cambia el rango de v, aunque para el caso hiperbólico usa valores distintos porque la superficie no es cerrada como una esfera. Con **np.meshgrid**, convierte esos vectores en una red bidimensional de puntos que luego se usarán para construir la superficie en 3D.

La función **get\_surface(a)** toma un valor del factor de escala a y genera las coordenadas x,y,z de la superficie:

- Para  $k = 1$  construye una esfera de radio a
- Para  $k = 0$  construye un disco que simula la expansión homogénea
- Para  $k = -1$  Construye una superficie hiperbólica. la famosa "silla de montar"

Es decir, aquí es donde la geometría se vuelve visual. Luego viene la función más importante para la animación con **update(frame)**. Esta función se ejecuta en cada cuadro de la animación.

Primero limpia la gráfica anterior (**ax.clear()**), luego toma el tiempo correspondiente a ese cuadro (**tiempo = t[frame]**). Después calcula el factor de escala, pero aquí es donde para efectos prácticos en lugar de usar la función **scale\_factor**, usa directamente una exponencial **a = np.exp(H\*tiempo)**. Esto físicamente significa que, sin importar la geometría elegida, siempre esté usando el caso plano. Con ese valor de a, calcula la superficie **get\_surface(a)**, la dibuja con cierta transparencia y le pone un título que muestra la geometría, el tiempo (en miles de millones de años) y el valor de a(t). Finalmente fija límites en los ejes de -4 a 4.

Ahora bien, esto es problemático porque el factor de escala, si el lector hace uso de la función `real`”**scale\_factor** puede ser muchísimo más grande, así que muchas veces la superficie ni siquiera se ve bien. Para ello se dejó a modo de comentario la función que permite que los ejes crezcan en función de la expansión con la función `lim=np.max(np.abs(x))` la cual permite que los límites de los ejes vayan desde `-lim` a `lim`.

Después de definir la animación, el código imprime un mensaje confirmando la geometría elegida, crea la animación con **FuncAnimation** indicando que hay tantos cuadros como valores de tiempo y finalmente intenta mostrarla como HTML.

# Apéndice G

## Escalar de Ricci

Dado que el escalar de Ricci está definido por la siguiente ecuación (Schutz, 2009, p. 164)

$$R = g^{\mu\nu} R_{\mu\nu}$$

Y que a su vez la traza del tensor métrico depende de las  $n$  dimensiones.

$$g^{\mu\nu} g_{\mu\nu} = n \quad (\text{G.1})$$

Entonces partiendo de las ecuaciones de campo de Einstein en de Sitter.

$$R_{\mu\nu} - \frac{1}{2}R \cdot g_{\mu\nu} + \Lambda g_{\mu\nu} = 0 \quad (\text{G.2})$$

Si se multiplica por la inversa del tensor métrico y sustituyendo las definiciones G.1 y G.2 se obtiene la siguiente expresión.

$$\begin{aligned} R_{\mu\nu} g^{\mu\nu} - \frac{1}{2}R \cdot g_{\mu\nu} g^{\mu\nu} + \Lambda g_{\mu\nu} g^{\mu\nu} &= 0 \\ R - \frac{1}{2}R \cdot n + \Lambda n &= 0 \end{aligned}$$

Despejando  $R$  se obtiene la siguiente ecuación

$$R = \frac{2n \cdot \Lambda}{n - 2}$$

Si  $n = 4$  Entonces  $R = 4\Lambda$  Teniendo en cuenta lo visto en la sección de **Métrica de Sitter** en el capítulo 3 con la expresión 2.40 en donde  $c=1$  se sabe que:

$$\Lambda = 3H^2$$

Entonces  $R = 12H^2$