



UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE CIÊNCIAS E TECNOLOGIA
UNIDADE ACADÊMICA DE FÍSICA
PROGRAMA DE PÓS-GRADUAÇÃO EM FÍSICA

George Dias Alves

TENSÃO DE HUBBLE E SUAS IMPLICAÇÕES

Campina Grande, Paraíba, Brasil

Fevereiro, 2024

George Dias Alves

TENSÃO DE HUBBLE E SUAS IMPLICAÇÕES

Dissertação apresentada ao Programa de Pós-Graduação em Física da Universidade Federal de Campina Grande, em cumprimento as exigências desta, como requisito parcial para a obtenção do título de Mestre em Física.

Orientador: Prof. Dr. Luciano Barosi de Lemos

Campina Grande, Paraíba, Brasil

Fevereiro, 2024

A474t

Alves, George Dias.

Tensão de Hubble e suas implicações / George Dias Alves. –
Campina Grande, 2024.

90 f. : il. color.

Dissertação (Mestrado em Física) – Universidade Federal de
Campina Grande, Centro de Ciências e Tecnologia, 2024.

"Orientação: Prof. Dr. Luciano Barosi de Lemos".

Referências.

1. Cosmologia. 2. Constante de Hubble. 3. Tensão de Hubble.
I. Lemos, Luciano Barosi de. II. Título.

CDU 524.8(043)



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
POS-GRADUACAO EM FISICA

Rua Aprigio Veloso, 882, - Bairro Universitario, Campina Grande/PB, CEP 58429-900

FOLHA DE ASSINATURA PARA TESES E DISSERTAÇÕES

GEORGE DIAS ALVES □

Tensão de Hubble e suas Implicações

Dissertação apresentada ao Programa de Pós-Graduação em Física como pré-requisito para obtenção do título de Mestre em Física.

Aprovada em: 07/02/2024

Luciano Barosi de Lemos
Presidente da Comissão e Orientador

Francisco de Assis de Brito
Examinador Interno

Morgana Lígia de Farias Freire
Examinador Externo



Documento assinado eletronicamente por **LUCIANO BAROSI DE LEMOS, PROFESSOR 3 GRAU**, em 20/02/2024, às 22:51, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da [Portaria SEI nº 002, de 25 de outubro de 2018](#).



Documento assinado eletronicamente por **FRANCISCO DE ASSIS DE BRITO, COORDENADOR DE POS-GRADUACAO**, em 05/03/2024, às 05:49, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da [Portaria SEI nº 002, de 25 de outubro de 2018](#).



Documento assinado eletronicamente por **Morgana Ligia de Farias Freire, Usuário Externo**, em 07/03/2024, às 16:50, conforme horário oficial de



Brasília, com fundamento no art. 8º, caput, da [Portaria SEI nº 002, de 25 de outubro de 2018](#).



A autenticidade deste documento pode ser conferida no site <https://sei.ufcg.edu.br/autenticidade>, informando o código verificador **4192086** e o código CRC **CCCFF166**.

Referência: Processo nº 23096.006053/2024-01

SEI nº 4192086

Dedicatória:

À minha família, aos meus amigos e minhas amigas e a todos (as) que me apoiaram nos dias difíceis.

Agradecimentos

Chegar até esse momento foi uma tarefa difícil. Mas com o apoio de pessoas especiais a estrada percorrida fica menos tortuosa. Então, gostaria de agradecer:

Agradeço imensamente a todas as pessoas que tornaram esta jornada possível. Em primeiro lugar, expresso minha profunda gratidão à minha família, que sempre esteve ao meu lado nos dias bons e nos desafiantes: minha mãe Maria dos Remédios, meu pai Georlando Dias e meu irmão Gean Dias. Estendo meu agradecimento a todos os outros familiares, em especial à minha tia Zeneide e ao meu tio João Dias, por seu apoio constante.

Um agradecimento especial é dedicado ao professor Dr. Luciano Barosi, cuja confiança em meu trabalho e orientações foram fundamentais para a realização deste projeto.

Também não posso deixar de expressar minha gratidão aos amigos e amigas que, com seu incentivo constante, me ajudaram a continuar nessa jornada: Alâne Marques, Aline Vanara, Brígida Alves, Francisca Lopes, Graziela Ribeiro, Guilherme Angelo, Jonas Barbosa, Lena Rufino, Lucas Leonel, Márcia Gomes e Renato Ramalho.

Também estendo meus agradecimentos a todo o corpo docente do Programa de Pós-graduação em Física da UFCG campus Campina Grande – PB e do curso de graduação em Física do campus Cajazeiras – PB da UFCG. Por fim, expresso minha gratidão a todas as pessoas que, direta ou indiretamente, contribuíram para minha jornada pessoal e acadêmica.

A todos e todas, meu muito obrigado !!!

"Hoje, ainda almejamos saber por que estamos aqui e de onde viemos. O desejo profundo da humanidade pelo conhecimento é justificativa suficiente para nossa busca contínua".

Stephen Hawking

Resumo

A Cosmologia é uma ciência que lida com questões que há muito tempo vêm despertando a atenção da humanidade. Ela busca compreender o universo tentando encontrar pistas que revelem dados sobre sua origem, estrutura e evolução e assim chegar a um modelo cosmológico mais preciso. Atualmente, o modelo mais aceito, o Lambda - Matéria Escura Fria, considera que nosso universo evoluiu de um estado de alta temperatura e densidade e encontra-se em processo de expansão e o parâmetro que mede a taxa de expansão é conhecido como constante de Hubble. A constante de Hubble pode ser determinada de várias maneiras, mas nos últimos anos dados medidos através de técnicas distintas, vem apresentando diferentes valores e isso configura a chamada: tensão de Hubble, nesse sentido, no presente trabalho buscamos através de uma revisão bibliográfica, estudar os métodos e dados experimentais para a obtenção da constante de Hubble, tornando clara a tensão gerada na divergência dos valores obtidos na sua medição.

Palavras-chave: Cosmologia. Constante de Hubble. Tensão de Hubble.

Abstract

Cosmology is a science that deals with questions that have long been of interest to humanity. It seeks to understand the universe by trying to find clues that reveal data about its origin, structure, and evolution, thus arriving at a more precise cosmological model. Currently, the most accepted model, the Lambda-Cold Dark Matter model, considers that our universe evolved from a state of high temperature and density and is in a process of expansion, with the parameter measuring the rate of expansion known as the Hubble constant. The Hubble constant can be determined in various ways, but in recent years, data measured through different techniques have been presenting different values, leading to what is known as the Hubble tension. In this regard, in the present work, we aim to study, through a literature review, the methods and experimental data for obtaining the Hubble constant, making clear the tension generated by the divergence of the values obtained in its measurement.

Keywords: Cosmology. Hubble Constant. Hubble Tension.

Lista de ilustrações

Figura 1 – Mapa celeste em projeção Mollweide, em coordenadas equatoriais, com as galáxias utilizadas por E. Hubble [1] para a determinação pioneira do valor da constante de Hubble. As coordenadas dos objetos celestes foram obtidas acessando o serviço SIMBAD [2].	28
Figura 2 – Diagrama esquerdo: Reprodução do gráfico original de Hubble para a determinação da constante de Hubble, utilizando as velocidades de recessão e distâncias citadas em [1] e com coordenadas celestes modernas. Diagrama direito: Determinação da constante de Hubble utilizando informações modernas de velocidade e distância obtidas pelo serviço SIMBAD [2].	30
Figura 3 – Distâncias cosmológicas comparadas para três modelos de universo. Modelo De-Sitter, dominado por energia escura, modelo de concordância Λ CDM e universo fechado. São indicadas as distâncias de diâmetro angular, distância comóvel e distância de luminosidade.	33
Figura 4 – Curva de brilho de 4 estrelas cefeidas típicas na pequena nuvem de magalhães. Eixo vertical indica a magnitude no filtro K e o eixo horizontal representa a fase equivalente do período, obtida a partir do tempo juliano heliocêntrico com a expressão $\phi(t) = \frac{t-\text{epoch}}{P} - \text{int}\left(\frac{t-\text{epoch}}{P}\right)$. Para a época foi utilizado o valor 2450000 e os dados foram obtidos a partir de [3], acessando o catálogo VIZIER [4].	37
Figura 5 – Demonstração da relação de Leavitt (período-luminosidade) para as cefeidas das galáxias Grande Nuvem de Magalhães, Andrômeda, NGC4258 e NGC5584, da esquerda para a direita. No eixo vertical, a magnitude de Wesenheit ($m_i^W = m_H - 0.34(V - I)$) e no eixo horizontal o logaritmo do período em dias. A reta vermelha indica o ajuste linear simples entre as duas variáveis. Os dados foram obtidos a partir de [5].	39
Figura 6 – Diagrama de Hertzsprung-Russel para estrelas do catálogo Hipparcos [6] com magnitudes inferiores 9. A Coloração das estrelas indica a percepção visual correspondente a temperatura de corpo negro no eixo superior e o tamanho das estrelas representa a classe de luminosidade, na sequência decrescente O, Ia, Iab, Ib, II, III, IV, V, VI, VII. Em vermelho estão destacadas as estrelas cefeidas, mostrando o seu acúmulo na região de instabilidade.	41
Figura 7 – Reprodução do diagrama de Hubble para as supernovas, com os dados obtidos em [7].	44

Figura 8 – Espectro de potência da radiação cósmica de fundo. Curva representa os valores teóricos com um universo com os parâmetros indicados em [8] e pontos são determinados pela colaboração Planck. No painel inferior encontra-se a fração de diferença entre os valores experimentais e os valores teóricos.	47
Figura 9 – Quatro simulações de universos com espectro de potência e mapas de anisotropias indicados, mostrando o energia escura, matéria e bárions para a formação de estruturas do universo.	49
Figura 10 – Parâmetros cosmológicos obtidos com as cadeias de markov-chain fornecidas pelo experimento Planck2018 [8], indicando os 6 parâmetros independentes do modelo de concordância e a constante de Hubble. . .	52
Figura 11 – Parâmetros cosmológicos obtidos com as cadeias de Markov-Chain disponibilizadas em [5] indicando os espaços de parâmetros para a quantidade de matéria, constante de Hubble e Magnitude fiducial da supernova tipo IA. A cadeia considera um universo tipo Λ CDM plano. . .	53
Figura 12 – A discrepância observada nas diferentes medições da Constante de Hubble H_0 hoje, conhecida como Tensão de Hubble [9].	59
Figura 13 – Esquema de evolução estelar, fora de escala [10].	68
Figura 14 – Representação de um diagrama HR, apresentando a localização de algumas estrelas conhecidas [11].	69

Lista de tabelas

Tabela 1 – A tabela exibe diferentes medições de H_0 realizadas por uma variedade de experimentos. Alguns desses estudos analisam objetos astronômicos próximos, como cefeidas e supernovas, enquanto outros se baseiam em dados da RCF obtidos por meio de satélites cosmológicos: WMAP e Planck.	55
--	----

Lista de abreviaturas e siglas

Λ CDM:	Lambda-Cold Dark Matter (em inglês lambda-matéria escura fria).
RCF:	Radiação Cósmica de Fundo
H_0 :	Constante de Hubble;
HR:	Herzsprung-Russell
COBE:	Cosmic Background Explorer.
WMAP:	Wilkinson Microwave Anisotropy Probe.
ESA:	Agência Espacial Europeia.
MCMC:	Markov Chain Monte Carlo.
EUA:	Estados Unidos da América.
Sne Ia:	Supernovas do tipo Ia.
RGV:	Ramo das Gigantes Vermelhas.
RH:	Ramo Horizontal.
MPC:	Modelo Padrão da Cosmologia.

Sumário

1	INTRODUÇÃO	23
2	LEI DE HUBBLE NA COSMOLOGIA	25
2.1	A Construção da Lei de Hubble	25
2.2	Distâncias, Redshift e H_0	30
2.2.1	Distâncias cosmológicas	31
2.2.2	O desvio espectral da luz	33
3	MEDIÇÕES LOCAIS DE H_0	35
3.1	Variáveis Cefeidas	35
3.2	Ponta do ramo da Gigante Vermelha	40
3.3	Supernovas do Tipo Ia	42
4	MEDIDAS DE H_0 COM A RCF	45
4.1	RCF e o modelo Λ CDM	49
4.2	Medidas de H_0 fornecidas pelos dados da RCFM	52
5	TENSÃO DE HUBBLE	55
6	CONCLUSÕES	61
	REFERÊNCIAS	63
	APÊNDICE A – ESTRELAS	67
A.1	Estrelas	67
A.2	Evolução estelar	67
A.3	Diagrama HR	68
	APÊNDICE B – CÓDIGOS PYTHON	71
B.1	Códigos da Figura 1	71
B.2	Códigos da Figura 2	72
B.3	Códigos da Figura 3	73
B.4	Códigos da Figura 4	74
B.5	Códigos da Figura 5	76
B.6	Códigos da Figura 6	78
B.7	Códigos da Figura 7	80
B.8	Códigos da Figura 8	81

B.9	Códigos da Figura 9	82
B.10	Códigos da Figura 10	84
B.11	Códigos da Figura 11	85
B.12	Códigos da Figura 12	85

1 Introdução

A Cosmologia é a uma das ciências que se dedicada à compreensão do universo, buscando pistas que revelem informações sobre sua origem, estrutura e evolução. Para desenvolver seus conhecimentos, essa área utiliza uma variedade de técnicas e ferramentas tecnológicas que possibilitam a obtenção dos dados essenciais para a construção de modelos cosmológicos [11, 12, 13].

Ao longo dos anos, diversos modelos cosmológicos foram propostos. No entanto, o atualmente mais amplamente aceito é o modelo Lambda- Matéria Escura Fria Λ CDM. Segundo ele, o universo teve sua origem a partir de um ponto extremamente quente e denso, expandindo-se de maneira violenta por meio de uma grande explosão conhecida como Big Bang. A validação experimental desse modelo ocorreu pela primeira vez em 1929, através das medições de distâncias entre galáxias realizadas por E. Hubble. Posteriormente, o reforço experimental veio com a descoberta, em 1964, da Radiação Cósmica de Fundo (RCF), realizada acidentalmente por dois engenheiros, Arno Allan Penzias e Robert Woodrow Wilson [12, 11].

O parâmetro que mede a proporção da expansão do universo é conhecido como constante de Hubble H_0 e é derivada da Lei de Hubble. H_0 também é o indicador cosmológico que define a escala, o tamanho e a idade do universo; é uma das formas mais diretas que temos de quantificar como o universo evolui. Atualmente, o valor de H_0 pode ser determinado de várias maneiras, mas classicamente as duas técnicas mais utilizadas envolve o uso das chamadas velas padrão, que são objetos bastante luminosos que possuem brilho conhecido e a outra forma envolve o manuseio dos dados da RCF que são obtidos pelos satélites observacionais [11, 14].

Usando a primeira técnica que envolve o uso das velas padrão, os cosmólogos chegaram a um valor para H_0 através das Cefeidas de $73, 8_{-2,4}^{+2,4}$ $\text{Km s}^{-1}\text{Mpc}^{-1}$, já o valor usando a ponta do ramo da gigante vermelha está fixado em $72, 3_{-1,1}^{+1,1}$ $\text{Km s}^{-1}\text{Mpc}^{-1}$ e a melhor estimativa para o valor usando supernovas do tipo Ia é igual $74, 03_{-1,42}^{+1,42}$ $\text{Km s}^{-1}\text{Mpc}^{-1}$. Por outro lado, o valor a partir dos dados da Radiação Cósmica de Fundo RCF nos fornece uma medida aproximada igual $67, 23_{-0,60}^{+0,60}$ Km s^{-1} para H_0 [14, 5, 15].

Essa divergência entre os valores obtidos para H_0 usando essas duas técnicas é conhecido como Tensão de Hubble. E é justamente esse aspecto o tema central da nossa pesquisa. Nela, buscaremos investigar, através de uma revisão bibliográfica e construção de gráficos os principais métodos utilizados para determinação da constante de Hubble.

Dessa forma, este trabalho está organizado em cinco capítulos que abordam distintos aspectos relacionados à Constante de Hubble. No segundo capítulo, dedicamo-nos a

apresentar os elementos fundamentais na elaboração da Lei de Hubble. Recriamos alguns dos gráficos originais propostos por Hubble em 1929, e oferecemos uma breve visão sobre as principais ferramentas empregadas na determinação de distâncias cosmológicas, destacando o papel crucial de H_0 na cosmologia.

O terceiro capítulo concentra-se nas técnicas utilizadas para a determinação de H_0 explorando medições locais e o papel desempenhado por velas padrão, como as Variáveis Cefeidas e as Supernovas do Tipo Ia, assim como a Ponta do Ramo da Gigante Vermelha. Apresentamos, também, os valores mais recentes obtidos para H_0 por meio dessas abordagens.

No quarto capítulo, abordamos a (RCF) e H_0 . No quinto capítulo, realizamos uma breve discussão sobre a Tensão de Hubble, contextualizando-a no panorama cosmológico. Apresentamos, por fim, ainda nesse capítulo, uma compilação dos resultados mais recentes para H_0 .

Em cada capítulo, destacamos de maneira especial os processos envolvidos na construção das figuras, utilizando a linguagem de programação Python. Com isso, buscamos oferecer uma visão prática e detalhada de como os dados cosmológicos podem ser manipulados usando essa linguagem de programação.

2 Lei de Hubble na Cosmologia

Naquela seção, foram examinados alguns dos elementos intrínsecos à elaboração da Lei de Hubble, assim como a consideração de parâmetros essenciais para a determinação de H_0 .

2.1 A Construção da Lei de Hubble

A curiosidade sobre os fenômenos cosmológicos está presente no imaginário das pessoas há bastante tempo. Atualmente, fenômenos desta natureza são estudados pela Cosmologia. A Cosmologia é a Ciência que estuda a estrutura, a evolução e a composição química do universo, seu principal objetivo é entender como o universo se formou, como ele está organizado e traçar os possíveis caminhos para seu destino no futuro [15, 13, 12].

A Cosmologia trabalha com os chamados modelos cosmológicos e em cada modelo os cosmólogos buscam caracterizar a geometria e a distribuição de matéria no universo. Até a década de 20, a tendência dos modelos cosmológicos indicavam que o universo era composto por estrelas em um espaço estático e vazio, mas toda essa ideia foi modificada com a constatação do fenômeno da expansão do universo, algo que já era previsto e proposto teoricamente nos trabalhos de W. de Sitter (1872-1934), A. Friedmann (1888-1925) e do belga G. Lemaître (1894-1966) e que foi posteriormente constatado nas observações realizadas pelos astrônomos G. Lemaître (1894-1966) e E. Hubble (1889-1953) e na descoberta (RCF) [12, 11, 16].

Em 1923, o astrônomo americano E. Hubble utilizando telescópio de 2,5 m de diâmetro do monte Wilson, na Califórnia concluiu que a nossa galáxia não era a única, como antes se pensava. Além disso, usando a recém descoberta relação período-luminosidade de H. Leavitt (1868–1921) ele também foi capaz de estimar a distância de Andrômeda e outras galáxias observando o brilho aparente e os períodos de pulsação de estrelas cefeidas nessas galáxias. [12, 11].

Desta observação Hubble hipotetizou a relação, escrita em coordenadas esféricas, entre distância e velocidade representada abaixo:

$$rK + X \cos \alpha \cos \delta + Y \sin \alpha \cos \delta + Z \sin \alpha = v \quad (2.1)$$

onde, α representa a ascensão reta e δ a declinação e representam as coordenadas das galáxias e X , Y e Z esta associada ao movimento relativo da terra e sua consequência é a construção de um termo de velocidade corrigida. Na hipótese de Hubble, poderia existir

uma relação de dependência entre a velocidade de afastamento das galáxias e a distância e com base nessa teoria se os dados observados são compatíveis com a Equação 2.1.

Nesse contexto, nossa abordagem consistiu em replicar os procedimentos seguidos por Hubble e compreender os métodos empregados no cálculo de H_0 que tem influência direta com dos modelos cosmológicos. Para dar início a esse processo, baseamo-nos no conhecimento disponível na literatura atual, referenciando fontes como [11], [12], [14], [13], [17]. Além disso, foram utilizados pacotes de software amplamente reconhecidos na área de análise de dados cosmológicos. Para manipulação e análise de dados, foram empregados os pacotes MATPLOTLIB [18], NUMPY [19], e PANDAS [20]. Já para acesso e processamento de dados astronômicos, foram utilizados os pacotes ASTROPY [21] e ASTROQUERY [22], enquanto o pacote GETDIST [23] foi empregado para análises específicas

A fim de acessar e manipular bancos de dados, a primeira etapa consiste na importação de um conjunto essencial de bibliotecas. Para concretizar essa ação, utiliza-se o `import`, seguida do nome específico da biblioteca desejada, como se apresenta a seguir:

```

1 import glob
2 from IPython.core.interactiveshell import InteractiveShell
3 from IPython.display import display
4 #-----
5 from adjustText import adjust_text
6 from astroquery.simbad import Simbad
7 import astropy.coordinates as coord
8 import astropy.units as u
9
10 import matplotlib.pyplot as plt
11 import numpy as np
12 import pandas as pd
13 import scipy as sp
14 from sklearn.linear_model import LinearRegression
15 import statsmodels.formula.api as smf
16 from sklearn.linear_model import Ridge
17 from sklearn.preprocessing import PolynomialFeatures, SplineTransformer
18 from sklearn.pipeline import make_pipeline
19 import getdist
20 from getdist import plots as gdplt
21 #-----
22 %matplotlib inline
23 plt.rcParams["figure.figsize"] = (16, 8)
24 InteractiveShell.ast_node_interactivity = "all"

```

Primeiro, foram obtidos os dados originais do artigo de Hubble [1] e colocamos em uma tabela, depois incluímos dados de catálogos modernos das posições das a distâncias das galáxias, utilizando o pacote SIMBAD.

A finalidade foi produzir um gráfico em coordenadas equatoriais que incluíse os objetos

celestes utilizados por Hubble e em seguida, um gráfico de velocidade em relação à distância, comparando os valores originais e modernos. Cabe ressaltar, que Hubble possuía dados de posição astrométrica, velocidade de recessão, medida por métodos espectroscópicos e distâncias obtidas através da observação de cefeidas nas galáxias hospedeiras. Mas, para utilizar a Equação 2.1 torna-se necessário fazer uma incrementação nos dados de Hubble com as informações modernas das coordenadas das posições das galáxias fornecidas pelo SIMBAD.

Primeiro, utilizamos um objeto abstrato chamado SIMBAD em Python que atuou como uma interface para consultas das propriedades específicas das galáxias. Para acessar as informações necessárias foi definido um filtro pelo qual acessamos dados como a ascensão reta, declinação, velocidade radial, distância e tipo de objeto:

```
1 s = Simbad('ra(d;icrs;2000)', 'dec(d;icrs;2000)', rv_value, distance,
           typed_id')
2 s.add_votable_fields
3 s.remove_votable_fields('coordinates')
```

Tínhamos os dados de Hubble [24] e construímos uma nova tabela com os valores obtidos do SIMBAD. O próximo passo foi juntar as informações e realizar as manipulações necessárias para trabalharmos na construção das figuras:

```
1 ra = coord.Longitude(df_hubble.RAJ2000, unit=u.deg).wrap_at(180*u.deg).
   radian
2 dec = coord.Latitude(df_hubble.DEJ2000, unit=u.deg).radian
3 OBJECTS = df_hubble['OBJECT'].to_numpy()
4 #
5 fig = plt.figure(figsize=(16, 9))
6 ax1 = fig.add_subplot(111, projection="mollweide")
7 ax1.set_title('Galáxias Utilizadas por Hubble')
8 ax1.scatter(x=-ra, y=dec, s=12, c='darkblue')
9 texts = []
10 for ra, dec, nome in zip(-ra, dec, OBJECTS):
11     texts.append(plt.text(ra, dec, nome, color='blue', fontsize=10))
12 adjust_text(texts, arrowprops=dict(arrowstyle="-", color='lightgray', lw
   =0.1))
13 ax1.set_xticklabels(['14h', '16h', '18h', '20h', '22h', '0h', '2h', '4h', '6h', '
   8h', '10h'])
14 ax1.set_xlabel(r'$\alpha$(h): Ascensão Reta')
15 ax1.set_ylabel(r'$\delta$(\textcircled{)}$: Declinação')
16 ax1.grid(True)
17 plt.savefig("fig01_george.png")
18 plt.tight_layout();
```

A Figura 1 mostra uma representação das galáxias analisadas por Hubble através da projeção de Mollweide. Esta visualização inclui as galáxias discutidas por Hubble em seu artigo.

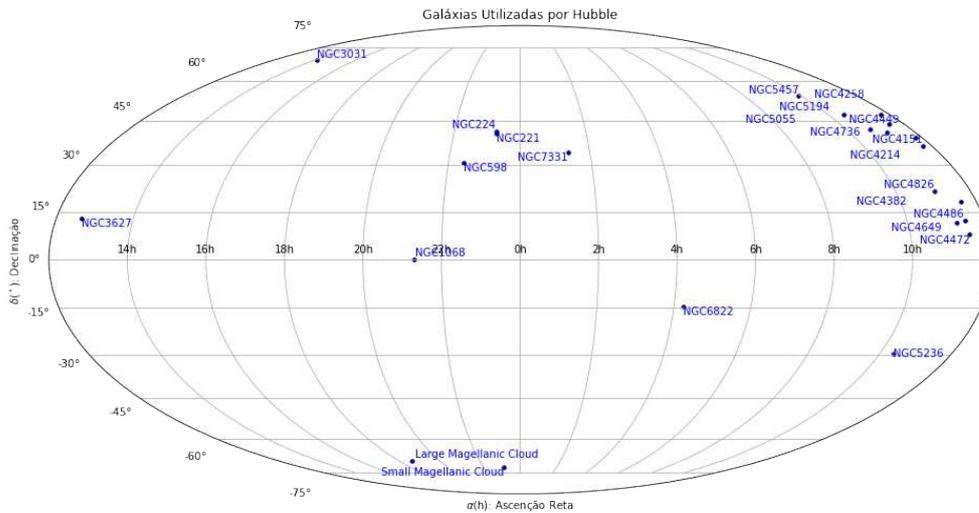


Figura 1 – Mapa celeste em projeção Mollweide, em coordenadas equatoriais, com as galáxias utilizadas por E. Hubble [1] para a determinação pioneira do valor da constante de Hubble. As coordenadas dos objetos celestes foram obtidas acessando o serviço SIMBAD [2].

A etapa seguinte envolveu a busca pela reprodução do diagrama de Hubble, o qual estabelece uma relação entre velocidade e distância. Nessa fase, obteve-se uma representação gráfica desse diagrama usando os dados de [1] e comparou-se esses dados com a base de dados modernos do SIMBAD [2]. Para tanto, foi necessário a produção de uma relação funcional que visa solucionar a Equação 2.1, encontrando os melhores valores para $'K'$, $'X'$, $'Y'$ e $'Z'$. Procedemos à criação de um novo conjunto de dados no qual utilizamos as colunas $'v'$, $'K'$, $'X'$, $'Y'$ e $'Z'$. Após a construção desse conjunto de dados, estabeleceu-se uma relação linear entre esses parâmetros, o que pode ser facilmente realizado de maneira intuitiva por meio do pacote `statsmodel`. Esse pacote nos permite definir uma fórmula linear que descreve a dependência linear da velocidade em relação aos parâmetros $'v'$, $'K'$, $'X'$, $'Y'$ e $'Z'$:

Na fase seguinte, recorreremos às medidas contemporâneas de posição e distância das mesmas galáxias que Hubble havia utilizado. Essa abordagem, proporcionou o uso de resultados substancialmente aprimorados, uma vez que a determinação das distâncias foi refinada por meio de métodos modernos de observação. E por fim, comparamos os resultados dos dados modernos com os dados usados por Hubble:

```

1 x, y, texto = df_hubble.distance, df_hubble.V_r, df_hubble.OBJECT
2 fig, ax = plt.subplots(ncols=2, figsize=(16,9))
3
4 ax[0].set_title('Diagrama de Hubble com dados históricos')
```

```

5 ax[0].scatter(df_hubble['r'], df_hubble['v'], marker = 'x', s=15, color=
    'violet',
6             label="velocidades medidas")
7 ax[0].scatter(df['K'], df['v_fit'], linewidth=2, color='yellow',
8             label=r'$v = r K + X \cos \alpha \cos \delta + Y \sin \alpha \cos \delta + Z \sin \delta$')
9 ax[0].plot(dis_corrected, model_corrected.predict(dis_corrected),
10          label=r'Ajuste Linear H={:.1f} Km/Mpc s'.format(model_corrected.
11                coef_[0]),
12          linewidth=2, color='brown')
13 ax[0].set_xlabel('d (MPc)')
14 ax[0].set_ylabel('v (Km/s)')
15 ax[0].legend()
16 ax[0].grid(True)
17 ax[1].set_title('Diagrama de Hubble atuais')
18 ax[1].scatter(df_hubble['distance'], df_hubble['V_r'], marker='o', s=15,
19             color='blue',
20             label="dados atuais")
21 texts = []
22 for a,b,c in zip(x,y,texto):
23     texts.append(ax[1].text(a, b, c))
24 adjust_text(texts, arrowprops=dict(arrowstyle="-", color='lightgray', lw
25     =0.1))
26 ax[1].plot(distance, model_current.predict(distance),
27          label=r'Ajuste Linear H={:.1f} Km/Mpc s'.format(model_current.
28                coef_[0]),
29          linewidth=2, color='brown')
30 ax[1].set_xlabel('d (MPc)')
31 ax[1].set_ylabel('v (Km/s)')
32 ax[1].legend()
33 ax[1].grid(True)
34 plt.savefig("fig02_george.png")
35 plt.show();

```

chegamos na Figura 2. Nela temos a representação do gráfico original de Hubble e utilizando informações modernas de velocidade e distância obtidas pelo serviço SIMBAD temos a representação do diagrama com os dados atuais. O uso dos dados das medidas modernas da posição e distância das mesmas galáxias, utilizadas por Hubble, permite obter resultados mais precisos. As técnicas de determinação das distâncias melhoraram significativamente com métodos mais modernos de observação.

Após a compilação e ajuste dos dados em um gráfico de velocidade em função da distância, Hubble chegou-se na relação linear:

$$v = H_0 d \quad (2.2)$$

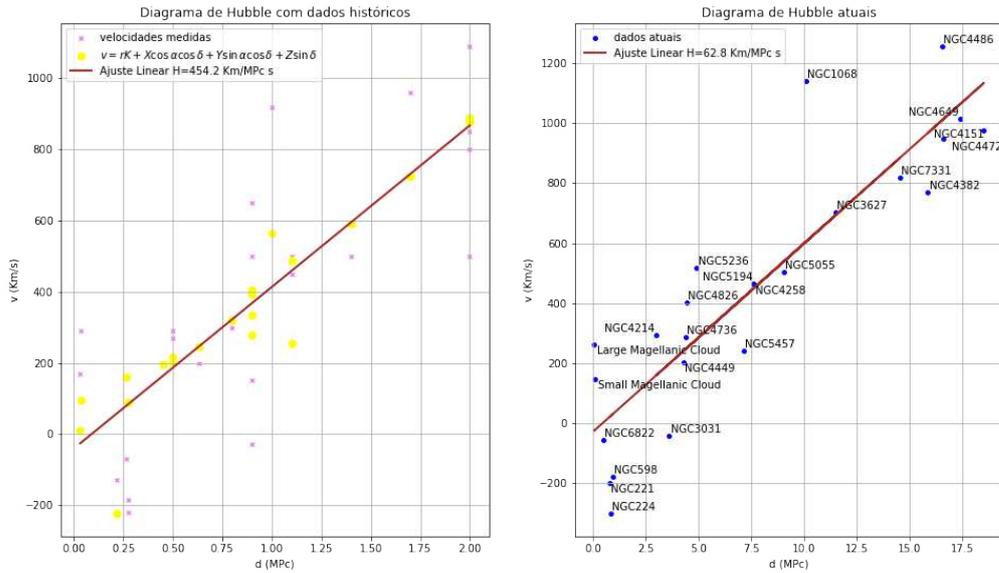


Figura 2 – Diagrama esquerdo: Reprodução do gráfico original de Hubble para a determinação da constante de Hubble, utilizando as velocidades de recessão e distâncias citadas em [1] e com coordenadas celestes modernas. Diagrama direito: Determinação da constante de Hubble utilizando informações modernas de velocidade e distância obtidas pelo serviço SIMBAD [2].

onde v é a velocidade de afastamento da galáxia, H_0 é a constante de Hubble e d é a distância.

É H_0 que nos fornece a atual taxa de expansão do universo, além disso, define sua escala de tamanho e idade. O valor de H_0 pode ser determinado localmente com medições de distâncias e desvios espectrais, e também pode ser previsto a partir de um modelo cosmológico calibrado com dados no início do universo com medições da RCF [5, 12, 11].

Os parâmetros principais para a determinação de H_0 são a distância e o desvio espectral. Esses parâmetros foram utilizados por E. Hubble para calcular o valor inicial de H_0 , que era de $500 \text{ Km s}^{-1} \text{ Mpc}^{-1}$. Nesse sentido, é importante discutir como ocorre a medição e como eles estão associados na determinação de H_0 .

2.2 Distâncias, Redshift e H_0

A Cosmologia observacional é uma área que trabalha com dados observacionais. Para calcular a distância até um objeto astronômico distante nos fazemos o uso da observação das propriedades observáveis deste objeto, como cores ou tipo espectral, análise espectral e luminosidade [12, 11].

2.2.1 Distâncias cosmológicas

Na cosmologia observacional, ou mais especificamente a parte da cosmografia, diversas abordagens são empregadas para especificar as distâncias entre dois objetos. O ponto comum em todas as técnicas é que mensuram a separação entre eventos ao longo das trajetórias de fótons [25]. Nesta seção, abordaremos inicialmente os aspectos relacionados à "escada de distância" e às diferentes medidas de distância, como a diâmetro angular, de luminosidade e a comóvel.

Para mapear o universo, os astrônomos utilizam a chamada escada de distância cósmica. Essa escada representa um conjunto de técnicas que permitem a determinação de distâncias em diferentes escalas do universo. Ela é chamada de escada devido ao fato de que cada degrau ou técnica de medição depende da etapa anterior para calibração. Logo, quanto maior a distância medida, maior a quantidade de passos [14, 26].

Dentro do sistema solar, os astrônomos determinam a distância entre os objetos, como os planetas, por exemplo, com precisão de 1 parte em um bilhão usando radares. Na ocasião, ondas de rádio de comprimentos variando de centímetros a metros são emitidas e, após a reflexão destas ondas em superfícies sólidas, é medido o tempo transcorrido entre a emissão da onda por uma antena e a detecção do sinal refletido. A distância ao objeto é então determinada pela relação:

$$d = c \frac{\Delta t}{2} \quad (2.3)$$

onde Δt é o tempo transcorrido e c é a velocidade do fóton emitido e refletido [12].

Para medir a distância fora do sistema solar, ou seja, para escalas acima de ≈ 10 UA, é utilizado o método da paralaxe estelar, que está baseado nos princípios da trigonometria. A Terra orbita em torno do Sol, daí existe um ligeiro deslocamento da posição das estrelas próximas em relação às estrelas mais distantes. A quantidade desse deslocamento é chamada ângulo de paralaxe, p (rad), e pode ser medido diretamente. De posse do valor deste deslocamento, poderemos determinar o valor do ângulo de paralaxe, p (rad) e, conseqüentemente, a sua distância, pela relação:

$$d = \frac{B}{p[\text{rad}]} \quad (2.4)$$

onde B é raio da órbita da Terra e p [rad] é o ângulo de paralaxe.

O método de Paralaxe Estelar, aplicado ao cálculo da distância de estrelas mais remotas, não mostra-se preciso, pois o ângulo de paralaxe é muito pequeno para medir a distância com precisão. Nessa situação, uma alternativa é a aplicação de métodos mais abrangentes, como: as distâncias de diâmetro angular, distância de luminosidade e distância de comóvel.

A distância de diâmetro angular, d_A é definida como a razão entre o tamanho transversal

físico, tamanho real do objeto (l), e seu tamanho angular em radianos, tamanho aparente ($\delta\theta$):

$$d_A = \frac{l}{\delta\theta} \quad (2.5)$$

Essa técnica é eficaz para calcular a distância de estrelas, galáxias e quasares, contanto que o tamanho real dessas estruturas seja conhecido ou estimado.

A distância de luminosidade, representada por d_L , relaciona a luminosidade L intrínseca do objeto ao fluxo medido de uma vela padrão f e pode ser obtida pela expressão:

$$d_L = \left(\frac{L}{4\pi f} \right)^{\frac{1}{2}} \quad (2.6)$$

Ela é eficaz para calcular a distância de objetos muito brilhantes, mas tem uma limitação significativa devido à absorção da luz ao longo de sua trajetória.

A distância comóvel, representada por d_C , é uma medida cosmológica que leva em consideração a expansão do universo ao tentar determinar a separação física entre dois objetos ou fenômenos cósmicos, e é dada por:

$$d_C = \frac{c}{H_0} \int_0^z \frac{dz'}{\sqrt{E(z')}} \quad (2.7)$$

$$E(z') = \Omega_r(1+z')^4 + \Omega_m(1+z')^3 + \Omega_k(1+z')^2 + \Omega_\Lambda$$

onde, d_C é a distância comóvel, c é a velocidade da luz, H_0 é a constante de Hubble, z é o redshift, os parâmetros: Ω_r , Ω_m e Ω_Λ representam respectivamente a densidade de radiação, de matéria e de energia escura e Ω_k é o parâmetro de curvatura. Se $\Omega_k < 0$ a curvatura é negativa, $\Omega_k > 0$ curvatura é positiva e se $\Omega_k = 0$ a curvatura é nula. Essa técnica é eficaz para calcular a distância de objetos em largas escalas cósmicas, como galáxias e aglomerados de galáxias, e depende da compreensão da taxa de expansão e do universo.

A relação utilizada para a distância comóvel (Equação 2.7) serve como base para reescrever as Equações 2.5 e 2.6 em termos dos parâmetros cosmológicos, considerando as relações [25]:

$$d_A = \frac{d_C}{(1+z)} \quad (2.8)$$

e

$$d_L = (1+Z) \cdot d_A \quad (2.9)$$

Como:

$$d_A = \frac{c}{(1+z) \cdot H_0} \int_0^z \frac{dz'}{\sqrt{E(z')}} \quad (2.10)$$

e

$$d_L = (1+z) \cdot \frac{c}{H_0} \int_0^z \frac{dz'}{\sqrt{E(z')}} \quad (2.11)$$

Na Figura 3, são apresentadas comparações de distâncias obtidas com três modelos de universo: o modelo De Sitter, caracterizado pela predominância de energia escura, o de concordância Λ CDM e o de universo fechado.

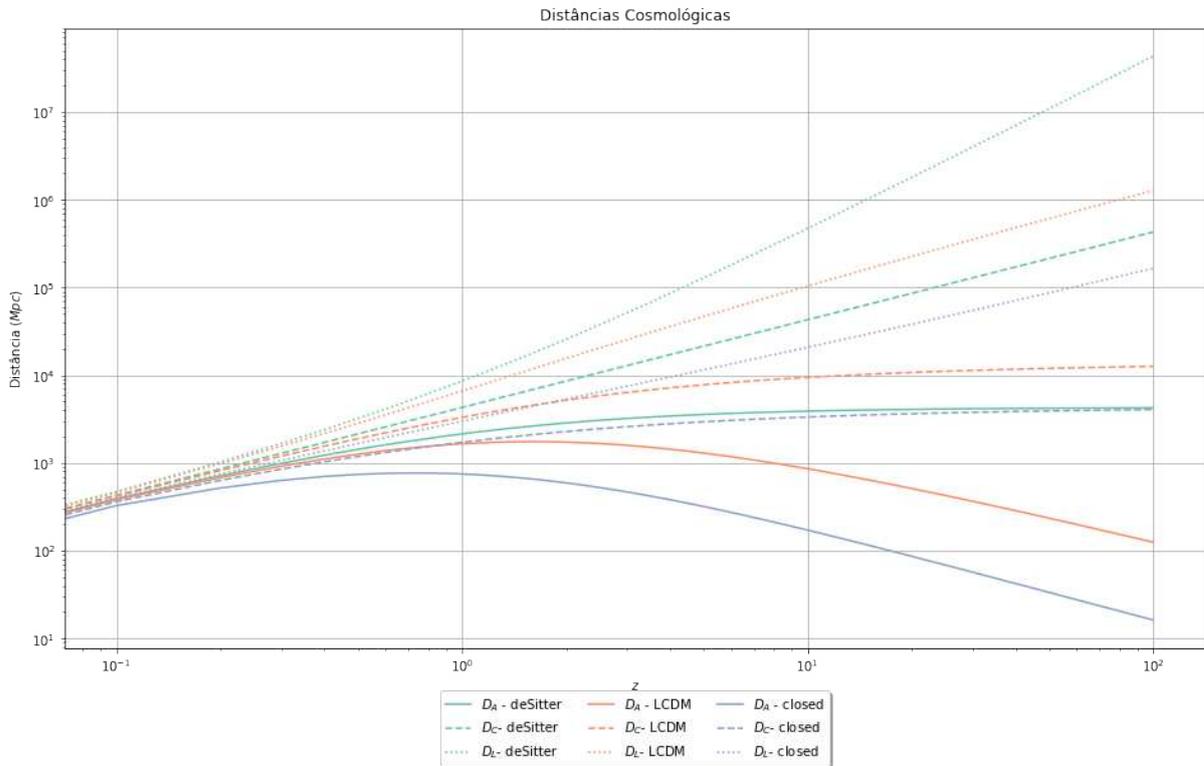


Figura 3 – Distâncias cosmológicas comparadas para três modelos de universo. Modelo De-Sitter, dominado por energia escura, modelo de concordância Λ CDM e universo fechado. São indicadas as distâncias de diâmetro angular, distância comóvel e distância de luminosidade.

2.2.2 O desvio espectral da luz

O desvio espectral da luz é outro parâmetro importante de medição na cosmologia. O Redshift (em português, desvio para o vermelho) e o Blueshift (desvio para o azul) caracterizam uma mudança na frequência detectada em relação à frequência emitida por uma fonte, devido ao movimento relativo entre a fonte e o detector[17].

Para medir o Redshift, é necessário, inicialmente, estabelecer um "espectro de referência". Como sabemos, cada elemento químico ou átomo possui um espectro único que o caracteriza.

O passo seguinte consiste na medição do deslocamento produzido no "espectro de referência", que surge devido ao movimento relativo da fonte ao observador.

Matematicamente, a relação que mede o desvio espectral da luz (z) é:

$$z = \frac{\lambda_{ob} - \lambda_{em}}{\lambda_{em}} \quad (2.12)$$

onde λ_{ob} é o comprimento de onda observado e λ_{em} é o comprimento de onda emitido pela galáxia. Nesse caso, quando $z > 0$, observamos o Redshift e quando $z < 0$ teremos Blueshift.

Conforme o observador se aproxima da fonte de ondas, as ondas têm os comprimentos de diminuídos, resultando em um aumento de suas frequências; caso haja um afastamento, as ondas são aumentam o comprimento de ondas, diminuindo as frequência. Quando os objetos são observados com tons avermelhados, significa que, sob seu ponto de vista, elas estão aparentemente se afastando; o contrário acontece caso seja percebido tons azulados.

A maioria das galáxias observadas aqui da terra apresentam Redshift. No entanto, não podemos considerar que estamos localizados em um lugar especial do universo, de onde todos os corpos celestes se afastam, pois observadores em qualquer outra parte também veriam galáxias distantes se afastando deles. Isso é fruto do processo de expansão homogênea e isotrópica [12].

3 Medições Locais de H_0

Este capítulo aborda a importância das Cefeidas na determinação de distâncias cósmicas, destacando a relação período-luminosidade estabelecida por Leavitt. A utilização dessas estrelas como velas padrão contribui para medir a constante de Hubble e compreender a expansão do universo, além de explorar alternativas, como Supernovas do Tipo Ia e a Ponta do Ramo da Gigante Vermelha. Dados recentes e métodos utilizados na pesquisa sobre a constante de Hubble são discutidos, fornecendo uma visão abrangente do cenário atual.

3.1 Variáveis Cefeidas

As variáveis cefeidas são estrelas variáveis pulsantes que brilham e escurecem em intervalos regulares. Por serem cerca de 10^5 vezes mais brilhantes que o sol, podem ser vistas mesmo estando a uma grande distância. Um dos primeiros registros da observação deste tipo de estrela foi feito em 1595 por um astrônomo amador. Ele notou que uma certa estrela, a Ceti que pertence à constelação Cetus, em determinadas vezes não era visível, mas depois voltava a aparecer. Em 1784, a chamada Delta Cephei que apresentava o mesmo comportamento foi descoberta. A explicação dada para esse comportamento da estrela na época sugeria que a oscilação no brilho ocorria porque havia manchas escuras nas superfícies destas estrelas que, às vezes, ficavam voltadas para a Terra [27].

Hoje, sabemos que as cefeidas são pulsantes radiais que aumentam e diminuem de tamanho periodicamente. Isso ocorre devido a expansão e contração das camadas superficiais da estrela. Nesse tipo de estrela, a energia interna produzida pela pressão da radiação gera desequilíbrio na estrela, modificando seu estado. No entanto, a força gravitacional atua como um agente restaurador, levando o sistema de volta ao estado inicial. É esse o agente que gera a pulsação [27]. (No Apêndice 6, temos a descrição da vida e evolução de estrelas).

Nó início dos anos 1900, dentro do Harvard College Observatory (E.U.A.), a astrônoma norte-americana H. Leavitt (1868-1921), catalogou cerca 1777 variáveis Cefeidas da Pequena e Grande Nuvem de Magalhães. De suas observações, surgiu a relação período-luminosidade, que define a existência de uma correlação entre o tempo necessário para uma cefeida escurecer e depois torna-se brilhante novamente com a sua magnitude aparente média.

Para exemplificar o comportamento das cefeidas, realizamos a aquisição dos dados relativos às curvas de luz dessas estrelas do catálogo [3] utilizando a biblioteca `ASTROQUERY`. Posteriormente, procedemos à construção de uma tabela contendo os valores de fase,

derivados das informações de período das estrelas. Com base nestes dados, elaboramos um gráfico representativo das curvas de luz, focalizando na magnitude K. Este procedimento visa proporcionar uma visão aprofundada das variações na luminosidade das cefeidas ao longo do tempo:

```

1 from astroquery.vizier import Vizier
2 catalog_list = Vizier.find_catalogs('Gieren W. , Storm J., Konorski P',
   max_catalogs=1000)
3 print({k:v.description for k,v in catalog_list.items()})
4
5 SMC_cepheids = list(catalog_list.keys())[2]
6 Vizier.ROW_LIMIT = -1
7 catalog = Vizier.get_catalogs(SMC_cepheids)
8 # Converte tabela de catalogo para pandas dataframe
9 cepheids = catalog[0].to_pandas()
10 lightcurves = catalog[1].to_pandas()
11 # Grafico da curva de luz para as 4 primeiras estrelas
12 fig, ax = plt.subplots(ncols=2, nrows=2)
13 df, XX, YY = lightcurve(0)
14 ax[0, 0].scatter( df.phase, df.Kmag, label=df.ID.values[0], marker = "o"
   )
15 ax[0, 0].plot( XX, YY, color="black", label="B-Spline interpolation")
16 ax[0, 0].legend()
17 df, XX, YY = lightcurve(1)
18 ax[1, 0].scatter( df.phase, df.Kmag, label=df.ID.values[0], marker = "o"
   )
19 ax[1, 0].plot( XX, YY, color="black", label="B-Spline interpolation")
20 ax[1, 0].legend()
21 df, XX, YY = lightcurve(2)
22 ax[0, 1].scatter( df.phase, df.Kmag, label=df.ID.values[0], marker = "o"
   )
23 ax[0, 1].plot( XX, YY, color="black", label="B-Spline interpolation")
24 ax[0, 1].legend()
25 df, XX, YY = lightcurve(3)
26 ax[1, 1].scatter( df.phase, df.Kmag, label=df.ID.values[0], marker = "o"
   )
27 ax[1, 1].plot( XX, YY, color="black", label="B-Spline interpolation")
28 ax[1, 1].legend()
29 plt.savefig("fig04_george.png")
30 plt.show();

```

na Figura 4, temos a representação da curva de brilho de 4 estrelas cefeidas típicas da Pequena nuvem de Magalhães em função de sua magnitude [27, 26].

O próximo passo consistiu na reconstrução da relação de Leavitt seguindo os passos do artigo de Riess et al [5]. Para isso, foram realizados um conjunto de imports das bibliotecas de softwares: MATPLOTLIB [18], NUMPY [19], PANDAS [20], SCIPY [28] e a aplicação

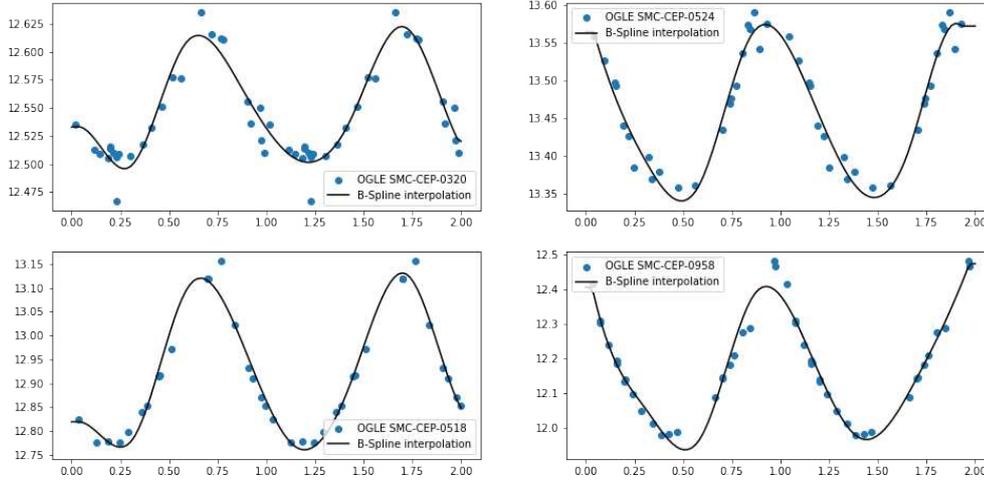


Figura 4 – Curva de brilho de 4 estrelas cefeidas típicas na pequena nuvem de magalhães. Eixo vertical indica a magnitude no filtro K e o eixo horizontal representa a fase equivalente do período, obtida a partir do tempo juliano heliocêntrico com a expressão $\phi(t) = \frac{t - \text{epoch}}{P} - \text{int}\left(\frac{t - \text{epoch}}{P}\right)$. Para a época foi utilizado o valor 2450000 e os dados foram obtidos a partir de [3], acessando o catálogo VIZIER [4].

do método de ajuste linear o `LinearRegression`.

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 import pandas as pd
4 import scipy as sp
5 from sklearn.linear_model import LinearRegression
```

Riess, et al [5] definem a expressão:

$$m_i^W = m_H - 0.34(V - I) = zp_{WN4258} + b_W \log P_i + Z_w \Delta(O/H)_i \quad (3.1)$$

onde o termo m_i^W representa a magnitude Wesenheit, que tem a função corretiva da metalicidade, o termo m_H é um filtro óptico o F160W, $(V - I)$ representa o índice de cor, zp_{WN4258} é um termo constante, $b_W \log P_i$ termo proporcional ao logaritmo e $Z_w \Delta(O/H)_i$ representa a incerteza de m_i^W .

Através dessas informações, o próximo passo consistiu em relacionar m_H , $b_W \log P_i$ e $Z_w \Delta(O/H)_i$. Através da aplicação da relação de Leavitt, encontrar o valor das constantes de forma a produzir o melhor ajuste linear minimizando o desvio dos resultados:

```
1 def fit_leavitt(data):
2     X = data[["logP", "metal"]]
3     y = data["mw"]
```

```

4 linear_regression = LinearRegression()
5 linear_regression.fit(X, y)
6 bw, zw = linear_regression.coef_
7 zpw = linear_regression.intercept_
8 logp_min, logp_max = data["logP"].min(), data["logP"].max()
9 logp_range = np.linspace(logp_min, logp_max, 100)
10 metal = data.metal.mean()*np.ones(100)
11 predict = pd.DataFrame({"logP": logp_range, "metal": metal})
12 YY = linear_regression.predict(predict)
13 return YY, logp_range, bw, zw, zpw

```

Após isso, utilizando a biblioteca Matplotlib e os comandos, a seguir:

```

1 fig, ax = plt.subplots(ncols=4, nrows=1, figsize=(16, 4))
2 ii = 0
3 for host, df in data.groupby("Host"):
4     YY, logp, bw, zw, zpw = fit_leavitt(df)
5     kk = ii % 4
6     ax[kk].errorbar(df["logP"], df["mw"], df["sigma_mw"], label=host,
7     fmt=".", alpha=0.6, linewidth=.3)
8     ax[kk].plot(logp, YY, color="red")
9     ax[kk].legend()
10    ii = ii + 1
11    ax[kk].invert_yaxis()
12 ax[0].set_ylabel(r"$m^W$")
13 fig.supxlabel(r'$Log\; P$')
14 plt.savefig("fig04_george.png")
15 plt.show();

```

geeramos a Figura 5 que é composta por quatro subgráficos de dispersão e linhas de ajuste que ilustram a relação de Leavitt (período-luminosidade) produzido para as cefeidas das galáxias Andrômeda, NGC4258 e NGC5584 da Grande Nuvem de Magalhães.

A relação permite identificar que a magnitude aparente média de uma cefeida é proporcional ao logaritmo de seu período. Quanto mais longo o período de uma cefeida, mais brilhante é a estrela. Essa observação foi algo importante porque fez nascer um novo método para medição de distâncias no universo [26, 3].

Na prática, a estratégia para determinar a distância de uma cefeida desconhecida consiste em acompanhar as variações de brilho e obter os valores para as magnitudes relativas correspondentes ao brilho máximo e ao brilho mínimo. Conhecendo o período, pode-se obter o valor da magnitude absoluta. Uma vez que a magnitude absoluta é conhecida, podemos inferir o módulo da distância dado pela relação:

$$\mu = m - M = 5 \log_{10} \left(\frac{d_L}{Mpc} \right) + 25 \quad (3.2)$$

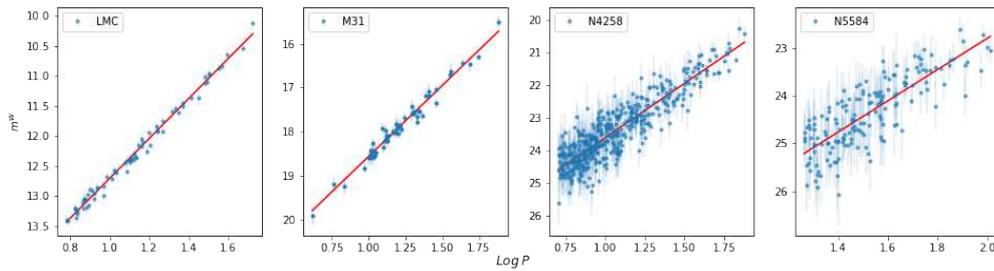


Figura 5 – Demonstração da relação de Leavitt (período-luminosidade) para as cefeidas das galáxias Grande Nuvem de Magalhães, Andrômeda, NGC4258 e NGC5584, da esquerda para a direita. No eixo vertical, a magnitude de Wesenheit ($m_i^W = m_H - 0.34(V - I)$) e no eixo horizontal o logaritmo do período em dias. A reta vermelha indica o ajuste linear simples entre as duas variáveis. Os dados foram obtidos a partir de [5].

onde d_L a distância de luminosidade em Mpc, m a magnitude aparente (fluxo), M a magnitude absoluta (luminosidade) e μ é a grandeza conhecida como módulo de distância.

As cefeidas também são utilizadas com ferramentas para a determinação da constante H_0 . Elas também foram usadas por Hubble durante suas análises cosmológicas na década de 20. O uso das velas cefeidas para determinação de H_0 é uma técnica que necessariamente obedece alguns passos. Para a determinação de H_0 segue os seguintes passos: [12],

- Identificar uma população de Cefeidas o com luminosidade L ;
- Medir o desvio espectral e o fluxo f para cada uma;
- Calcular $d_L = (L)/(8\pi f)$;
- Plotar um gráfico $cz \times d_L$. Onde c e z são respectivamente a velocidade da luz e o desvio espectral e d_L a distância de luminosidade;
- Medir a inclinação do gráfico $cz \times d_L$ para $z \gg 1$. O resultado da inclinação será o valor de H_0 .

O SH0ES é um experimento que usa o método da escada de distância para medir H_0 . Ele utiliza observações das Variáveis cefeidas em galáxias hospedeiras de Supernovas do Tipo Ia (SNE Ia) do Telescópio Espacial de Hubble. Em 2011, Ress et al usando 8 SNE Ia para cefeidas em NGC 4258, 13 cefeidas da Via Láctea medidas por paralaxe trigonométrica e 92 cefeidas da Grande Nuvem de Magalhães, e usando 8 SNE Ia para calibrar o desvio para o vermelho, obteve um valor para H_0 igual a $73, 8 \pm 2, 4 \text{ Km s}^{-1} \text{Mpc}^{-1}$. Em 2014, no entanto, Efstathiou et al realizaram uma reanálise no trabalho de Riess et al e sua reanálise reduziu H_0 para $70 \pm 2, 6 \text{ Km s}^{-1} \text{Mpc}^{-1}$. Examinando a correção de

Efstathiou em 2016, Riess reduziu a incerteza da medição, e H_0 passou a ter o valor de $73,24 \pm 1,74 \text{ Km s}^{-1}\text{Mpc}^{-1}$ [14].

Em 2019, Ress et al., usando os dados produzidos de 70 Cefeidas da Grande Nuvem de Magalhães pelo Telescópio Hubble, reduziram a incerteza em suas medições para 1%, e incorporando medidas aprimoradas chegaram à melhor estimativa para o valor de H_0 $74,03 \pm 1,42 \text{ Km s}^{-1}\text{Mpc}^{-1}$. Hoje, o valor de H_0 , usando as cefeidas está fixado em $73,2_{-0,8}^{+0,8} \text{ Km s}^{-1}\text{Mpc}^{-1}$ [14].

3.2 Ponta do ramo da Gigante Vermelha

A evolução estelar é um processo que ocorre em uma escala de tempo significativamente mais longa do que os padrões humanos, e o destino final de uma estrela é determinado principalmente pela sua massa. Na Figura 6, o diagrama de Hertzsprung-Russel (HR) ilustra os possíveis caminhos evolutivos da vida de uma estrela. Durante a maior parte de seu ciclo evolutivo, as estrelas estão na sequência principal, onde convertem hidrogênio em hélio em seus núcleos através da fusão nuclear. Esse processo libera uma grande quantidade de energia, criando uma pressão interna que equilibra a força da gravidade, mantendo a estrela estável em um estado conhecido como equilíbrio hidrostático [29, 11, 14].

À medida que o hidrogênio é convertido em hélio, a fonte de energia no núcleo da estrela lentamente se esgota, fazendo com que as camadas mais externas, atraídas pela gravidade, desçam em direção ao núcleo estelar. Isso acaba gerando uma redistribuição de massa através de processos convectivos, que contribuem para o aumento da estrela em dezenas de vezes, tornando-a Gigante Vermelha. A estrela no diagrama HR tem uma trajetória ascendente, em direção ao Ramo das Gigantes Vermelhas ou (RGV), o seu núcleo, rico em hélio, permanece inerte [29, 11, 14].

Na camada superior ao núcleo, o hidrogênio é convertido em hélio e lentamente esse hélio vai sendo depositado no núcleo estelar, fazendo a temperatura aumentar no processo. Ao atingir 100 milhões de Kelvin, inicia-se a fusão do hélio e a estrela sai do RGV, indo para Ramo Horizontal (RH), iniciando uma nova etapa evolutiva.

Grande parte das estrelas do (RH), principalmente as de tipo espectral A ou F, são variáveis, ou seja, brilho varia com o tempo e são facilmente detectáveis e conhecidas como RR *Lyrae*. O período de variabilidade geralmente é de 12 a 24 horas, e a amplitude de variação é de 0,5 a 1,0 magnitude. Por terem essas características elas podem ser utilizadas para determinar a distâncias no universo [29].

As estrelas RR Lyrae também são frequentemente utilizadas como velas padrão, e o cálculo da distância surge da comparação da luminosidade conhecida (a magnitude absoluta) e a luminosidade observada (sua magnitude aparente), pela expressão:

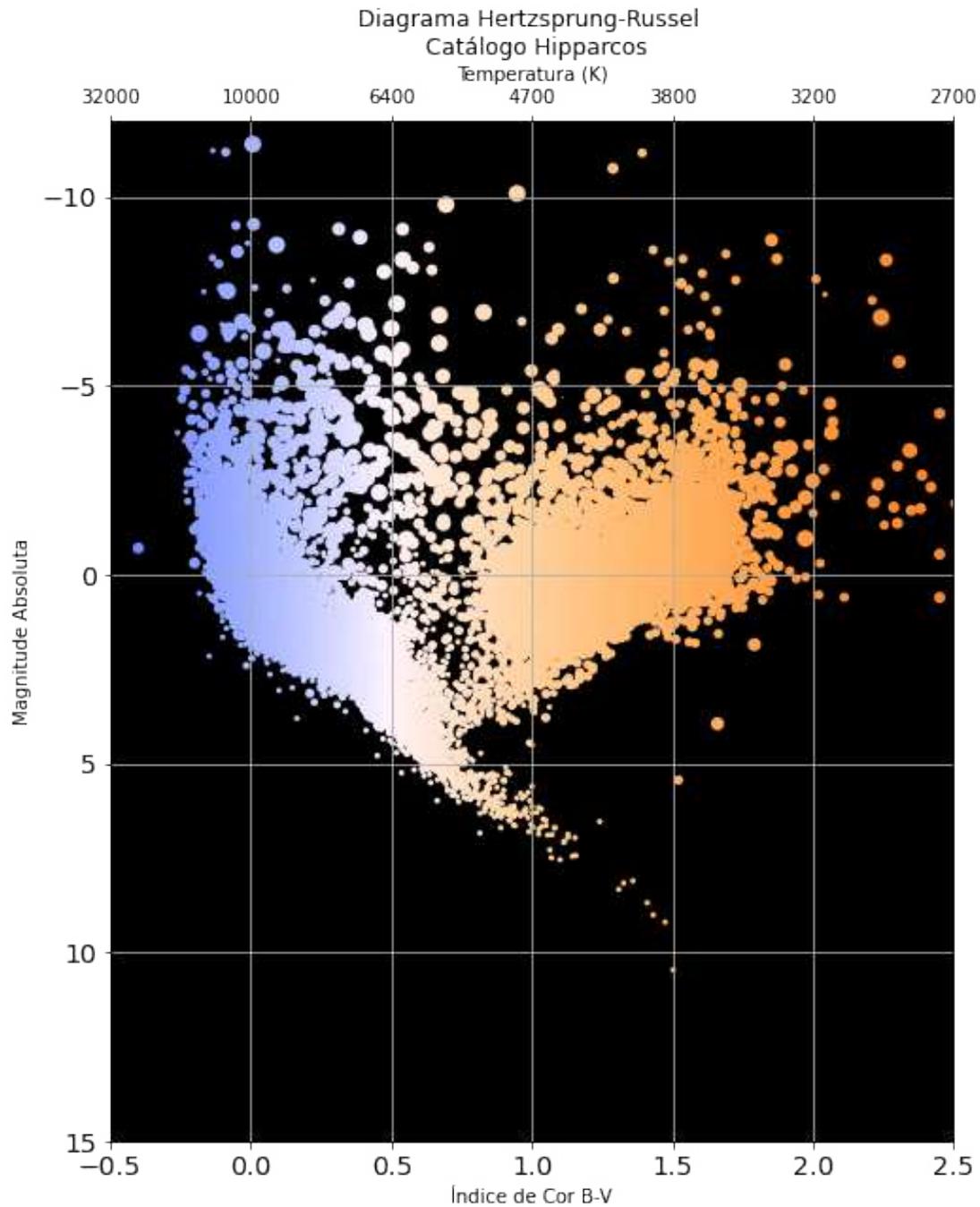


Figura 6 – Diagrama de Hertzsprung-Russel para estrelas do catálogo Hipparcos [6] com magnitudes inferiores 9. A Coloração das estrelas indica a percepção visual correspondente a temperatura de corpo negro no eixo superior e o tamanho das estrelas representa a classe de luminosidade, na sequência decrescente O, Ia, Iab, Ib, II, III, IV, V, VI, VII. Em vermelho estão destacadas as estrelas cefeidas, mostrando o seu acúmulo na região de instabilidade.

$$\mu = m - M = 5 \log_{10} \left(\frac{d_L}{\text{Mpc}} \right) + 25 \quad (3.3)$$

onde d_L a distância de luminosidade em Mpc, m a magnitude aparente (fluxo), M a magnitude absoluta (luminosidade) e μ é o módulo de distância.

A equipe da astrônoma Freedman [30], buscando formas alternativas para a escada de distância sem usar as variáveis cefeidas, pois, segundo ela, as cefeidas apresentam certos inconvenientes como a alta metalicidade e a presença de metais pesados que certamente poderiam afetar a Lei de Leavitt, utilizou as RR *Lyrae*. Ao contrário das hospedeiras das cefeidas, as galáxias hospedeiras da Ponta do Ramo da Gigante Vermelha são mais comumente encontradas em todo o universo observável nos chamados halos estelares [14], então elas são menos afetadas pela poeira estelar e pela aglomeração, ao contrário das variáveis Cefeidas [30, 31, 14]. Em 2019, [30] apresentou uma medida de H_0 , fixado em $72, 3_{-1,1}^{+1,1} \text{ Km s}^{-1} \text{ Mpc}^{-1}$. Este valor está de acordo com a medição do caminho de distância das cefeidas [14].

3.3 Supernovas do Tipo Ia

Para estudar estrelas em galáxias mais distantes, ou pelo menos alguns bilhões de anos-luz, os astrônomos necessitam de objetos mais brilhantes que cefeidas e RR *Lyrae*. A candidata que mostrou-se mais viável para isso foram as supernovas do Tipo Ia. Uma supernova é a explosão de uma estrela em sua fase final da evolução, as do Tipo Ia surgem da explosão de uma anã branca em um sistema estelar binário [27, 14].

Estrelas de massa intermediária, até 10 massas solares, após a queima total do hidrogênio, terminam sua evolução se transformando em uma anã branca. Nessa fase, a estrela não realiza reações de fusão nuclear para gerar energia, e todo material da estrela, devido a gravidade, é atraído para seu centro. O que explica o fato da estrela não colapsar é o princípio de exclusão de Pauli - os elétrons começam a se acumular e, para não violarem o princípio de Pauli, fazem pressão em direção oposta ao núcleo, gerando o equilíbrio entre as forças [32, 33, 11].

As supernovas do Tipo Ia só acontecem em sistemas binários compostos de uma anã branca e uma estrela de sequência principal. O material da estrela companheira é gradualmente depositado e absorvido pela anã branca, devido à ação da gravidade. Esse processo, faz com que a massa da anã branca aumente até ultrapassar o limite de Chandrasekhar. Após atingir esse limite ocorre a explosão. Esse evento é muito parecido para todas as configurações desta natureza e possuem um brilho bastante característico fato que permite a utilização delas como velas padrão [32, 33, 11].

Para identificar a relação entre o módulo de distância μ e o Redshift z das supernovas

do Tipo Ia realizamos a construção do diagrama de Hubble para esses objetos. Para tanto, inicialmente baixamos os dados do conjunto PANTHEON, que foram posteriormente organizados em um DataFrame usando a biblioteca PANDAS.

```
1 URL_SN = " https :// raw . githubusercontent .com/ PantheonPlusSHOES /
2 DataRelease / main / Pantheon %2 B_Data /4 _DISTANCES_AND_COVAR /
   Pantheon %2 BSHOES .dat "
3 df = pd.read_table(URL_SN , sep="\s+")
```

Na sequência, foram calculadas usando o `cosmos_astropy`, as distâncias luminosas em função dos redshifts:

```
1 zz = np.linspace(0.001, 3, 100)
2 mus = []
3 for key, cosmo in cosmos_astropy.items():
4     mu = 5 * np.log10(cosmo.luminosity_distance(zz).value) + 25
5     mus.append(mu)
```

E usando a biblioteca MATPLOTLIB, construímos a Figura 7 que é um gráfico de μ versus z de para diferentes modelos cosmológicos.

```
1 import matplotlib.ticker as ticker
2 labels = ["De - Sitter", r"$\Lambda$CDM", "Fechado"]
3 fig, ax = plt.subplots(figsize=(16, 6))
4 for name, _df in df.groupby("IDSURVEY"):
5     ax.errorbar(x=_df.zHD, y=_df.MU_SHOES, yerr=_df.MU_SHOES_ERR_DIAG,
6               label=IDSURVEY[name], fmt=".")
7     ax.plot(zz, mu, label=labels[ii])
8 ax.set_title("Supernovas")
9 ax.set_xlabel("$z$")
10 ax.set_ylabel("$\mu$")
11 ax.legend(loc='lower center', bbox_to_anchor=(0.5, -0.3), ncol=10,
12         fancybox=True, shadow=True)
13 ax.set_xscale("log")
14 ax.xaxis.set_major_formatter(ticker.ScalarFormatter())
15 ax.ticklabel_format(style='plain', axis='x', useOffset=False)
16 plt.grid()
17 plt.savefig("fig07_george.png", bbox_inches='tight')
18 plt.show()
```

Dentro da cosmologia, as supernovas apresentam um importante papel, pois através dela tem-se a possibilidade do refinamento das estimativas da constante de Hubble H_0 . Além disso, elas têm sido fundamentais para os estudos relacionados à expansão acelerada do universo [34].

O raciocínio empregado para o cálculo distâncias e H_0 utilizando supernovas do tipo Ia segue a mesma lógica usadas nas variáveis cefeidas e RR *Lyrae*. Usando a cefeidas

no degrau anterior e a luminosidade de supernovas do tipo Ia [5] encontraram a melhor estimativa atualizada para H_0 como sendo igual $74,03_{-1,42}^{+1,42}$ Km s⁻¹Mpc⁻¹ [14, 27, 35].

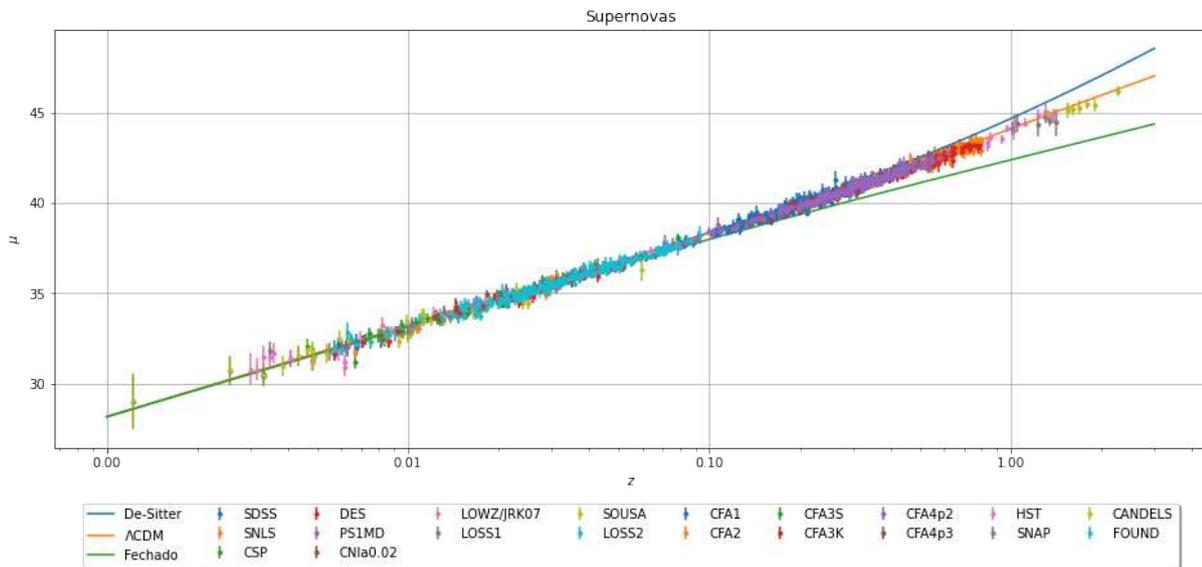


Figura 7 – Reprodução do diagrama de Hubble para as supernovas, com os dados obtidos em [7].

4 Medidas de H_0 com a RCF

A RCF é uma radiação eletromagnética que permeia todo o universo. É uma das quantidades astronômicas observáveis mais importantes no estudo da cosmologia e é remanescente do Big Bang ocorrido há aproximadamente 14 bilhões de anos [12]. Ela foi prevista teoricamente por George Gamov, Ralph Alpher e Robert Herman em 1948, mas só foi detectada acidentalmente em 1964 por dois engenheiros, Arno Allan Penzias e Robert Woodrow Wilson. A descoberta da RCF veio fortalecer o Modelo Expansionista, a ideia do Big Bang e a concepção de que o universo se expandiu de um estado altamente denso e com temperatura bastante elevada para o estado atual [36, 37].

A RCF é uma das fontes mais ricas de informação sobre como era o universo primordial e, muitas vezes, é referida como radiação fóssil pelo fato dela nos revelar o passado do universo desde os seus primeiros instantes. Pode ser descrita por uma curva de espectro de um corpo negro a uma temperatura de 2,725 K e sua frequência varia entre 10-100 GHz. A RCF é detectada através de balões atmosféricos, antenas ou por satélites astronômicos e ao longo das últimas décadas, esses detectores vem apresentando dados cada vez mais completos e precisos sobre o universo primordial [12, 11].

A RCF é observada principalmente na faixa de micro-ondas, ou seja, a uma frequência muito diferente daquela que tinha em seu último espalhamento. É justamente a expansão do universo a responsável por essa mudança. Nos últimos tempos, a análise da RCF por meio de colaborações e satélites atuais é um marco significativo na cosmologia de alta precisão. Esse progresso revolucionário permitiu a obtenção de parâmetros cosmológicos com níveis de precisão bastante elevados.

A retrospectiva a seguir ilustra a evolução da exploração desse recurso ao longo das últimas décadas. Em 1989, a NASA lançou o satélite Cosmic Background Explorer (COBE), com o objetivo de analisar detalhadamente a natureza da RCF na faixa de micro-ondas. Foi o COBE que forneceu informações cruciais sobre pequenas variações na temperatura da RCF, essenciais para entender a formação de estrelas e galáxias durante a aglomeração gravitacional da matéria recém-formada [11, 36].

Além do COBE, mais dois outros satélites foram projetados para analisar a RCF. O Wilkinson Microwave Anisotropy Probe (WMAP), lançado em 2001, possui uma resolução de 12 arcmin. Sua missão foi estudar as anisotropias na temperatura da RCF, e ele também foi capaz de detectar a polarização apresentada pela radiação [36, 14, 11].

O terceiro satélite, o Planck, afiliado à Agência Espacial Europeia (ESA), foi lançado em maio de 2009 para estudar o universo primitivo, em particular as variações de temperatura e polarização da RCF. Em comparação com os outros dois satélites, o Planck apresenta

a resolução mais elevada. Ele ficou em atividade entre 2009 e 2013, e por meio de sua colaboração, foram publicados três grandes conjuntos de dados: em 2013, 2015 e 2018 [36, 14, 11].

A partir dos dados obtidos por esses satélites é possível construir o gráfico do espectro de potência da RCF. O espectro de potência descreve as flutuações na distribuição angular de temperatura da RCF na esfera celeste, e essas flutuações foram importantes para a formação de estruturas cósmicas, como galáxias e aglomerados de galáxias. Elas são expressas por meio de uma expansão em harmônicos esféricos, que são funções matemáticas usadas para decompor padrões esféricos complexos [38].

Para isso ser feito, inicialmente, foram importadas as bibliotecas úteis para a manipulação dos dados da RCF e posteriormente construção dos gráficos: **NUMPY** (np), **PANDAS** (pd), **MATPLOTLIB** (plt), **SCIPY** (sp) e para podermos fazer a manipulação e download de dados de planK2018 as bibliotecas **GLOB**, **OS**, **REQUESTS** e **WGET** também foram importadas.

```

1 import numpy as np
2 import pandas as pd
3 import scipy as sp
4 from glob import glob
5 import itertools
6 import os
7 import requests
8 import sys
9 import warnings
10 import wget

```

Foram também criadas variáveis que armazenam URLs para o banco de dados que foram acessados (planck url: para os dados do Planck 2018; map url: para os mapas do CMB; cllbinned url: para os resultados binned (agrupados) do espectro de potência):

```

1 planck_url = 'https://irsa.ipac.caltech.edu/data/Planck/release_3/'
2 # 2018 Planck results
3 LCDM_params_url = 'ancillary-data/cosmoparams/COM_PowerSpect_CMB-base-
    plikHM-TTTEEE-lowl-lowE-lensing-minimum_R3.01.txt'
4 url = planck_url + LCDM_params_url
5 LCDM_params = pd.read_table(url, delim_whitespace=True, skiprows=3,
    names=['value', 'param', 'latex', 'fora'])
6 LCDM_params = LCDM_params[["value", "param"][:-5]]
7 LCDM_params.set_index("param", inplace = True)

```

Com os dados baixados do catálogo Planck-2018 e com os dados que obtivemos com o código **camb**, acessamos dados relativos a parâmetros específicos, como taxa de expansão do universo e características da potência inicial do espectro de densidade para calcular os resultados teóricos do espectro de potência da RCF. Os dados do espectro de potência observado e os erros associados foram carregados. A diferença relativa entre os resultados

observacionais e teóricos foi calculada e a construção dos gráficos foi realizada com o `gridspec`. Os passos para a realização dessas operações podem ser visualizados no Apêndice B na seção B.8. Na Figura 8 podemos visualizar a comparação dos resultados teóricos e observacionais do espectro de potência da RCF, destacando a diferença relativa entre eles por meio de gráficos.

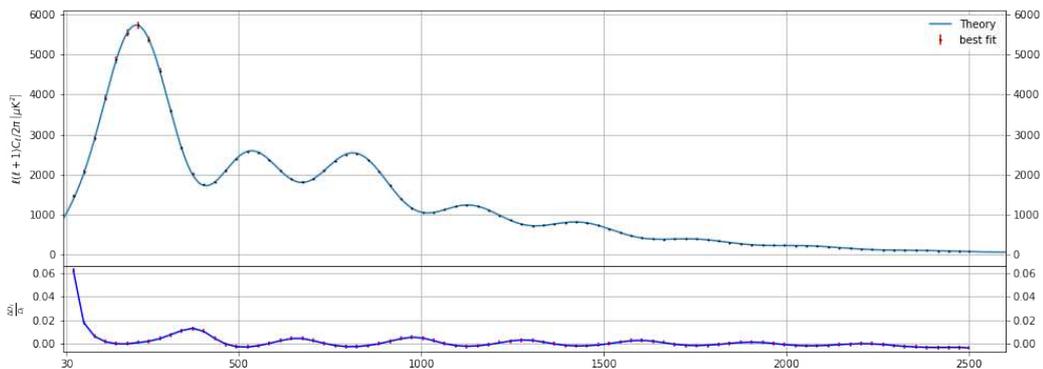


Figura 8 – Espectro de potência da radiação cósmica de fundo. Curva representa os valores teóricos com um universo com os parâmetros indicados em [8] e pontos são determinados pela colaboração Planck. No painel inferior encontra-se a fração de diferença entre os valores experimentais e os valores teóricos.

O próximo passo consistiu em criar um recorte do céu em uma aproximação conhecida como céu plano. Nesta abordagem, não é necessário realizar a expansão em harmônicos esféricos, que é computacionalmente intensiva. Em vez disso, pode-se considerar a expansão em modos de Fourier. Fez-se um recorte de 64° para ilustrar como o espectro de potência está relacionado com as estruturas que surgem em diferentes escalas angulares.

Para este propósito é definido a função `get_cll`:

```
1 def get_cll(cosmology = None, size = 64, pixels = 512):
2     """Obtem espectro de potência para visualizar efeitos cosmológicos
3     do conteúdo de matéria do universo."""
```

que recebe os parâmetros cosmológicos e retorna o espectro de potência teórico da RFC e um mapa simulado. Na sequência, definidos quatro conjuntos de parâmetros cosmológicos diferentes: `param_planck`, `param_DE`, `param_MATTER` e o `param_BARYON`:

```
1 param_planck = { "H0": H0,
2                 "Om_L" : float(LCDM_params.loc['omegal'].value),
3                 "Om_M" : float(LCDM_params.loc['omegam'].value),
4                 "Om_K" : float(LCDM_params.loc['omegak'].value),
5                 "Om_Bh2" : float(LCDM_params.loc['omegab2'].value)}
6 param_DE = { "H0": H0,
7             "Om_L" : float(LCDM_params.loc['omegal'].value)+0.3,
8             "Om_M" : float(LCDM_params.loc['omegam'].value)-0.3,
```

```

9         "Om_K" : float(LCDM_params.loc['omegak'].value),
10        "Om_Bh2" : float(LCDM_params.loc['omegab2'].value)}
11 param_MATTER = { "H0": H0,
12                 "Om_L" : float(LCDM_params.loc['omegal'].value)-0.6,
13                 "Om_M" : float(LCDM_params.loc['omegam'].value)+0.6,
14                 "Om_K" : float(LCDM_params.loc['omegak'].value),
15                 "Om_Bh2" : float(LCDM_params.loc['omegab2'].value)}
16 param_BARYON = { "H0": H0,
17                 "Om_L" : float(LCDM_params.loc['omegal'].value),
18                 "Om_M" : float(LCDM_params.loc['omegam'].value),
19                 "Om_K" : float(LCDM_params.loc['omegak'].value),
20                 "Om_Bh2" : float(LCDM_params.loc['omegab2'].value)*4}

```

Novamente, a função `get_c11` é aplicada a cada conjunto de parâmetros, resultando em quatro conjuntos de dados que representam diferentes modelos cosmológicos:

```

1 nomes = ["planck2018",
2          r"$\Omega_\Lambda = \Omega_\Lambda^{\mathrm{planck}} + 0.3$",
3          r"$\Omega_M = \Omega_M^{\mathrm{planck}} + 0.6$",
4          r"$\Omega_{bh}^2 = 4; (\Omega_{bh}^2)^{\mathrm{planck}}$"]
5 cosmologias = [param_planck, param_DE, param_MATTER, param_BARYON]
6 universos = [ get_c11(cosmology=cosmologia) for cosmologia in
7              cosmologias]

```

É por fim, usando a função `synfast_flat` da biblioteca PYMASTER:

```

1 fig, ax = plt.subplots(nrows = 2, ncols = 4, figsize = (16,9))
2 cmap=cm.get_cmap('viridis')
3 #normalizer=colors.Normalize(-500,500)
4 normalizer=colors.SymLogNorm(linthresh = 10, vmin = -500, vmax = 500,
5                               base = 10)
6 im=cm.ScalarMappable(norm=normalizer, cmap = cmap)
7 for ii, universo in enumerate(universos):
8     D11 = universo[0]
9     ell = np.arange(0,D11.shape[0])
10    ax[0,ii].plot(ell[2:], D11[2:], label = nomes[ii])
11    ax[0,ii].grid()
12    ax[0,ii].legend(loc = "upper center")
13    ax[0,ii].set_ylim([0,10000])
14    ax[1,ii].imshow(universo[1][0], interpolation='nearest', origin='
15                    lower',
16                    cmap = cmap, norm = normalizer)
17 fig.colorbar(im, ax=ax.ravel().tolist(), orientation="horizontal")
18 plt.savefig("fig09_george.png")
19 plt.show();

```

Chegou-se a Figura 9. No painel superior, cada gráfico mostra o espectro de potência teórico da RCF para os diferentes conjuntos de parâmetros cosmológicos e no painel

inferior, cada gráfico exibe um mapa simulado do RCF correspondente a cada modelo cosmológico. A variação de temperatura associada a cada mapa é indicada pela faixa de cores na parte inferior.

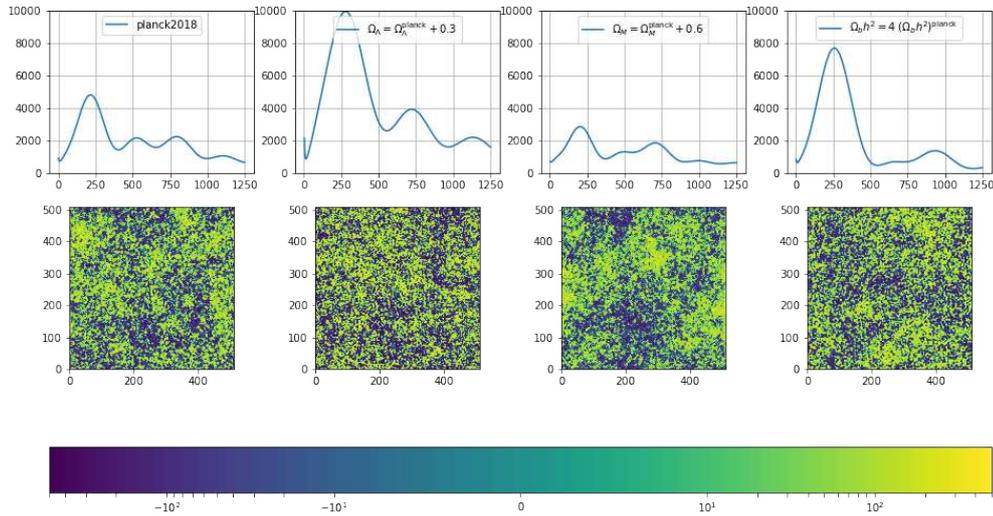


Figura 9 – Quatro simulações de universos com espectro de potência e mapas de anisotropias indicados, mostrando o energia escura, matéria e bárions para a formação de estruturas do universo.

4.1 RCF e o modelo Λ CDM

Entre as teorias cosmológicas que buscam explicar a origem e a evolução do universo, temos o Modelo Padrão da Cosmologia (MPC), também conhecido pela sigla Λ CDM, onde Λ representa a componente de energia escura e a sigla CDM significa “Cold Dark Matter”, em português, Matéria Escura Fria. O modelo Λ CDM, também costuma ser chamado LCDM. Os parâmetros cosmológicos que descrevem esse modelo, tais como $\Omega_B h^2$, Ω_m , t_0 , n_s , $\log A$, τ e H_0 podem ser estimados a partir de informações da RCF capitadas com precisão através do satélite Planck [38, 37]. Nesse contexto, torna-se extremamente necessário saber quais desses parâmetros estão correlacionados, para o planejamento do melhor experimento que forneça dados confiáveis dentro da estatística.

Mas o que faz um modelo ser realmente aceitável estatisticamente é ele corroborar teoricamente com os dados obtidos experimentalmente. No campo da estatística na cosmologia, há duas principais abordagens: a frequentista e a bayesiana. A abordagem frequentista, associada ao passado, baseia-se em repetições de experimentos para inferir resultados. Por exemplo, ao lançar uma moeda 100 vezes, espera-se que ela caia cara

aproximadamente 50% das vezes. Por outro lado, a abordagem bayesiana trata eventos como únicos, atribuindo probabilidades a hipóteses. No contexto cosmológico, a estatística bayesiana é usada tanto por razões históricas quanto pelo fato dela fornecer os intervalos de confiança para parâmetros cosmológicos sendo útil na interpretação de observações [39].

No caso de dados observacionais cosmológicos, o método bayesiano é recomendado apenas quando os erros não são estatisticamente independentes, como é comum em observações astrofísicas. Em muitos casos, como no projeto Planck, podem surgir potenciais erros na posição dos objetos ou até mesmo na computação dos dados. A abordagem bayesiana, expressa através do teorema de Bayes, é fundamental nesse processo, pois fornece a probabilidade condicional, ou seja, a probabilidade de um modelo ser verdadeiro dado um conjunto específico de dados experimentais disponíveis.

Nesta situação, é necessário aplicar uma função de regressão linear bayesiana - a função de verossimilhança que deve ser maximizada e para isso a exploração do espaço de parâmetros deve ser realizada com a técnica das cadeias de Markov (MCMC) que oferece uma maneira de varrer o espaço de parâmetros. A função de verossimilhança desempenha um papel central nesse contexto, sendo equivalente ao χ^2 ¹ quando os erros são estatisticamente independentes.

Atualmente, Λ CDM é um modelo amplamente aceito pelos cosmólogos para descrever a evolução do universo. Esse modelo específico possui um conjunto definido de parâmetros que o caracterizam. Para aprofundar a compreensão e examinar a variabilidade da função de verossimilhança associada a esses parâmetros, recorreremos a Monte Carlo via Cadeias de Markov - (MCMC), obtidas por [5], utilizando a função de verossimilhança disponibilizada. Posteriormente, construímos gráficos, proporcionando uma representação visual do intervalo de confiança desses parâmetros e suas correlações.

De acordo com a análise dos parâmetros cosmológicos relativos aos resultados da missão Planck-2018 [8], acessamos os dados de cadeias de MCMC referentes ao modelo Λ CDM [40]. Em seguida, construímos um gráfico com os seis parâmetros do MPC acrescidos da constante de planck para efeitos de comparação. Para tanto, inicialmente, fizemos o carregamento dos dados de diferentes cadeias de Markov gerados pelo experimento Planck:

```
1 samples1 = np.loadtxt("../data/raw/plikHM_TT/base_plikHM_TT_1.txt")
2 samples2 = np.loadtxt("../data/raw/plikHM_TT/base_plikHM_TT_2.txt")
3 samples3 = np.loadtxt("../data/raw/plikHM_TT/base_plikHM_TT_3.txt")
4 samples4 = np.loadtxt("../data/raw/plikHM_TT/base_plikHM_TT_4.txt")
```

Na sequência, os dados carregados foram convertidos em **DataFrames** do **PANDAS** para facilitar a manipulação e análise dos dados. Após isso, selecionou um subconjunto específico de parâmetros para análise e renomeação das colunas para nomes mais legíveis:

¹ É um teste de hipóteses que se destina a encontrar um valor da dispersão para duas variáveis categóricas nominais e avaliar a associação existente entre variáveis qualitativas.

```

1 samples1 = pd.DataFrame(samples1[:,2:], columns=column_names)
2 samples2 = pd.DataFrame(samples2[:,2:], columns=column_names)
3 samples3 = pd.DataFrame(samples3[:,2:], columns=column_names)
4 samples4 = pd.DataFrame(samples4[:,2:], columns=column_names)
5 samples = pd.concat([samples1, samples2, samples3, samples4])

1 use_params = ["omegab2", "omegam*", "age*", "ns", "logA", "tau", "H0*"]
2 labels = [ r"\Omega_b h^2", r"\Omega_m", r"t_0", r"n_s", r"\log A",
            r"\tau", r"H_0"]
3 df_planck = samples[use_params]
4 df_planck.columns = ["omegab2", "omegam", "age", "ns", "logA", "tau", "
                       H0"]
5 gsamples = getdist.MCSamples(samples=df_planck.to_numpy(), names=
                               use_params, labels=labels)

```

Por fim, os dados foram convertidos em um objeto `MCSamples` para posterior visualização com a biblioteca `GetDist`. Com a biblioteca `GetDist` foi gerado o gráfico da Figura 10:

```

1 gsamples = getdist.MCSamples(samples=df_planck.to_numpy(), names=
                               use_params, labels=labels)
2 g = gdplt.get_subplot_plotter()
3 g.settings.figure_legend_frame = False
4 g.settings.alpha_filled_add=0.4
5 g.settings.title_limit_fontsize = 14
6 g.triangle_plot(gsamples, use_params, filled=True, legend_loc='upper
                   right', title_limit=3)
7 plt.savefig("fig10_george.png")
8 plt.show()

```

Com a Figura 10, podemos visualizar as distribuições marginais e as relações entre os parâmetros selecionados. Ela também indica os valores mais prováveis para cada um dos parâmetros cosmológicos e as relações entre eles. Quando a forma da Figura assemelha-se a um "ovo frito", como ocorre, por exemplo em $H_0 \times \Omega h^2$ onde não há correlação entre as variáveis. No entanto, se a figura apresentar achatamento, como ocorre, por exemplo, em $H_0 \times \Omega_m$ existe correlação, indicando que as medidas dos parâmetros são dependentes entre si.

Também foi construída as cadeias de Markov para um modelo cosmológico considerado, o Λ CDM plano. Esta é uma versão do MPC onde a geometria do universo é assumida como sendo plana. Para tal procedimento, acessamos dados de Cadeias de Markov (MCMC) relacionados aos parâmetros cosmológicos disponíveis em <https://github.com/PantheonPlusSH0ES>, mais especificamente para os parâmetros associados à densidade de matéria Ω_m , H_0 e M_B da supernova tipo Ia, uma vez que o experimento é sensível apenas a esses parâmetros. O código desse processo pode ser visualizado em no Apêndice B.11. Podemos visualizar a distribuição probabilística desses parâmetros, dentro

do contexto do modelo cosmológico adotado, e seus intervalos de confiança e correlações na Figura 11.

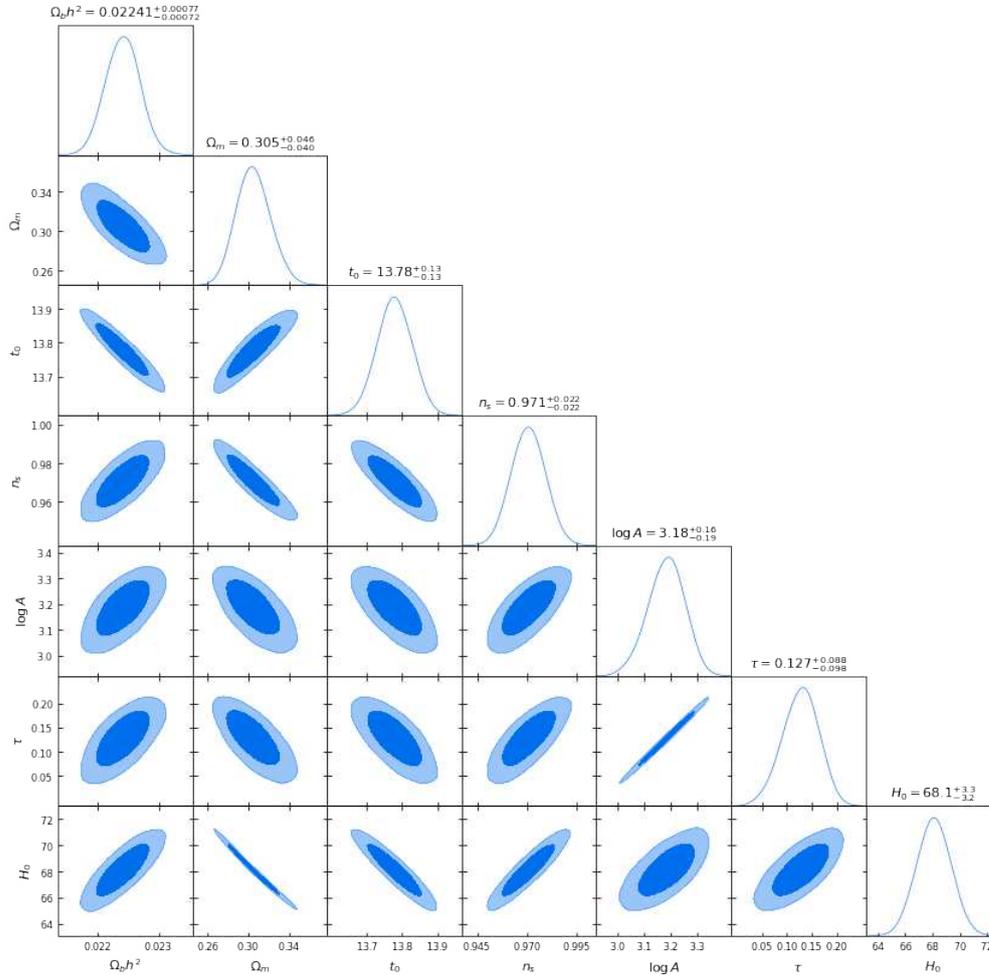


Figura 10 – Parâmetros cosmológicos obtidos com as cadeias de markov-chain fornecidas pelo experimento Planck2018 [8], indicando os 6 parâmetros independentes do modelo de concordância e a constante de Hubble.

4.2 Medidas de H_0 fornecidas pelos dados da RCFM

Ajustando os dados da RCF a um modelo Λ CDM com seis parâmetros, conforme ilustrado na Figura 10, diferentes estudos foram conduzidos utilizando dados do satélite WMAP. Utilizando dados de 5 anos do WMAP, resultou em uma estimativa para H_0 , de $71,9 \pm 2,6 \text{ Km s}^{-1} \text{ Mpc}^{-1}$ [41]. Posteriormente, [42], com resultados de 7 anos do WMAP, obteve uma estimativa ligeiramente diferente, $71,0 \pm 2,5 \text{ Km s}^{-1} \text{ Mpc}^{-1}$. Por fim, o ajuste

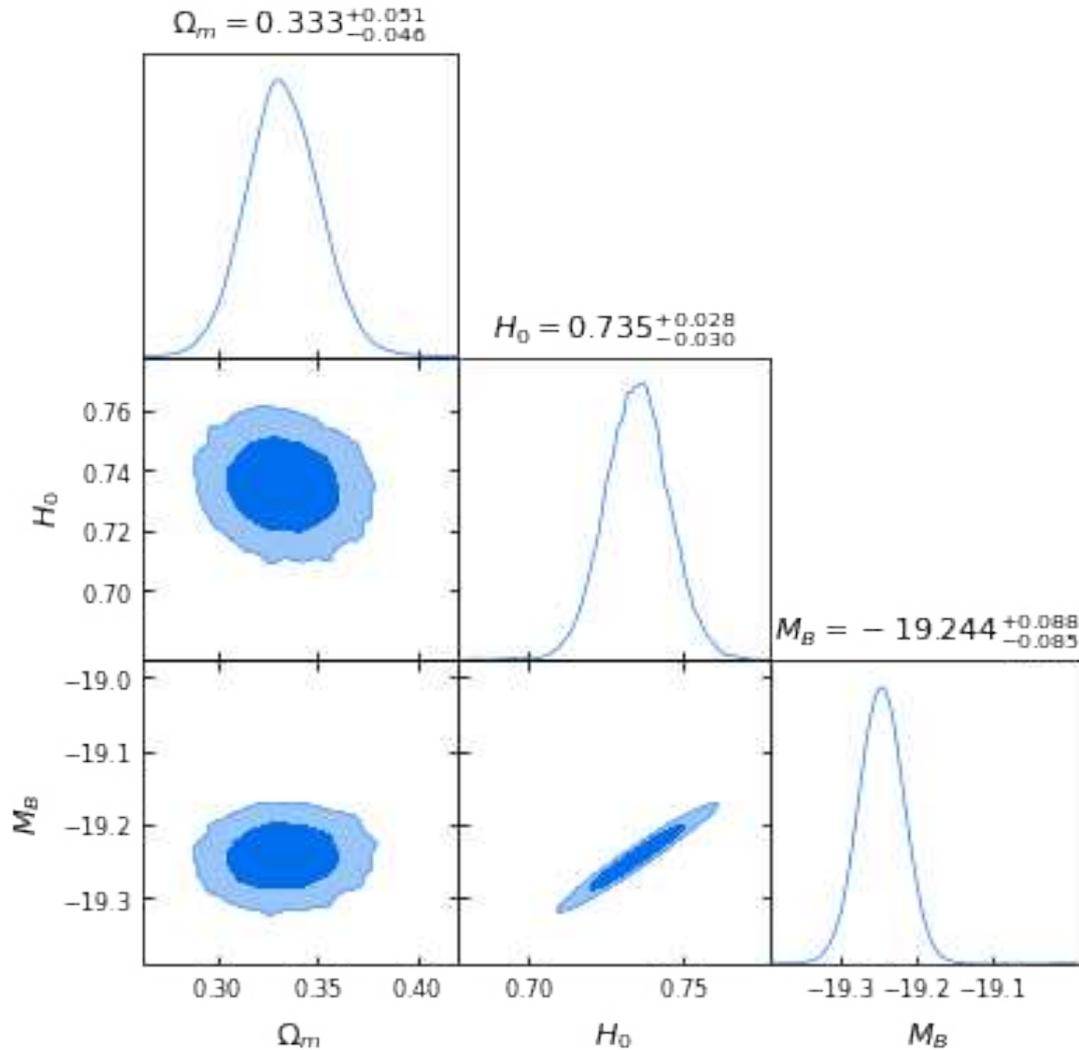


Figura 11 – Parâmetros cosmológicos obtidos com as cadeias de Markov-Chain disponibilizadas em [5] indicando os espaços de parâmetros para a quantidade de matéria, constante de Hubble e Magnitude fiducial da supernova tipo IA. A cadeia considera um universo tipo Λ CDM plano.

realizado com dados de 9 anos, conforme apresentado por [43], produziu um valor de $H_0 = 71,0 \pm 2,2 \text{ Km s}^{-1}\text{Mpc}^{-1}$. Essas estimativas ilustram a evolução nas determinações de H_0 com o aprimoramento dos dados e técnicas de observação e diminuição das incertezas.

As análises mais recentes da RCF foram conduzidas pelo satélite Planck, uma missão da Agência Espacial Europeia. Os dados do Planck, em conjunto com o modelo Λ CDM mínimo contendo seis parâmetros, foram utilizados em 2013 para obter uma estimativa da constante H_0 , resultando em $67,4 \pm 1,4 \text{ Km s}^{-1}\text{Mpc}^{-1}$, com um nível de confiança de 68%. Os resultados do Planck em 2015 mantiveram consistência com valores anteriores,

apresentando uma estimativa de H_0 de $67,27 \pm 0,66 \text{ Km s}^{-1}\text{Mpc}^{-1}$. Na análise mais recente do Planck em 2018, o valor para H_0 foi reportado como $67,27 \pm 0,60 \text{ Km s}^{-1}\text{Mpc}^{-1}$. Esses resultados indicam uma concordância notável entre as análises sucessivas. Entretanto, ao examinarmos os valores obtidos por meio de medições locais utilizando as chamadas "velas padrão", percebemos uma divergência nos valores atribuídos a H_0 . Essa disparidade é conhecida como Tensão de Hubble, um tópico que discutiremos na próxima seção.

5 Tensão de Hubble

A constante H_0 é um dos parâmetros essenciais em cosmologia, pois desempenha um papel crítico ao impor restrições significativas sobre o modelo cosmológico, que procura descrever a evolução integrada do universo. Além disso, essa constante caracteriza a taxa de expansão do universo no presente momento, define o tamanho observável do cosmos, e seu inverso fornece a idade da expansão cósmica [12].

A obtenção de H_0 envolve diversas formas e técnicas, algumas das quais foram apresentadas nas seções anteriores. Em síntese, as principais abordagens incluem as medições locais com o uso das chamadas velas padrão ou o uso da RCF. Na Tabela ??, são apresentados os resultados mais recentes de experimentos e medições da constante de Hubble. Cada entrada na tabela representa um experimento específico, destacando a medida atualizada para a constante de H_0 . Na análise apresentada, cada linha representa um método específico de medição da constante de Hubble, H_0 . Os valores centrais são acompanhados por intervalos de incerteza, indicando a variabilidade associada às respectivas medições.

Se analisarmos as diferenças entre os valores de H_0 , podemos perceber que as estimativas que usam as medições locais cefeidas, Ramo da Gigante Vermelha e supernovas do Tipo Ia são bastante próximas indicando uma concordância razoável entre esses métodos. As medidas usando dados do WMAP e Planck revela uma divergência considerável de aproximadamente 3,73 nos valores da constante de Hubble. Por outro lado, se considerarmos as medições de Planck *versus* Média das Medidas podemos notar uma diferença de aproximadamente 6,77 e são nessas divergências que nasce a Tensão de Hubble.

A partir de dados medidos das últimas observações de grandes experimentos buscamos construir uma representação visual da Tensão de Hubble existente entre as diferentes observações deste parâmetro. Para tanto, inicialmente, foram realizados `imports` de bibliotecas no código para facilitar manipulações e configurações. Os módulos `os` e `sys`

Experimento	H_0 (km s ⁻¹ Mpc ⁻¹)
Cefeidas	$74,03 \pm 1,42$
Ramo da Gigante Vermelha	$69,8^{+0,6}_{-1,6}$
Supernovas Tipo Ia	$74,03 \pm 1,42$
WMAP	$71,03 \pm 2,2$
Planck	$67,27 \pm 0,60$

Tabela 1 – A tabela exhibe diferentes medições de H_0 realizadas por uma variedade de experimentos. Alguns desses estudos analisam objetos astronômicos próximos, como cefeidas e supernovas, enquanto outros se baseiam em dados da RCF obtidos por meio de satélites cosmológicos: WMAP e Planck.

foram incorporados para lidar com operações relacionadas a caminhos de arquivos e interação com o sistema operacional. Adicionalmente, o módulo `matplotlib.pyplot` foi importado para a criação de gráficos, enquanto o módulo de personalização `util` também foi incluído.

```
1 import os
2 import sys
3 sys.path.append(os.path.abspath(os.path.join(os.path.split(os.getcwd())
4 [0], "src")))
5 import matplotlib.pyplot as plt
6 from util import
```

Na sequência, realizamos algumas operações visando estimar o valor H_0 e a análise da tensão em relação à estimativa do Planck. Para tanto, criamos uma sequência linear de 10.000 pontos com intervalos de 50 a 100. E em seguida aplicamos instâncias da classe `Estimate` que representa uma estimativa da constante H_0 para diferentes experimentos. Também há em cada instância o nome da estimativa (`name`), valor médio (`mean`), e desvio padrão (`std`).

```
1 flat = np.linspace(50, 100, num=10000)
2 # create estimate
3 tdc = Estimate(name="HOLiCOW+STRIDES", mean=73.7, std=1.5)#from https://
4 ui.adsabs.harvard.edu/abs/2019arXiv190704869W/abstract
5 planck = Estimate(name="Planck", mean=67.4, std=0.5)#from https://ui.
6 adsabs.harvard.edu/abs/2018arXiv180706209P/abstract
7 des = Estimate(name="DES+BAO+BBN", mean=67.4, std=1.15)#from https://ui.
8 adsabs.harvard.edu/abs/2018MNRAS.480.3879A/abstract
9 boss = Estimate(name="BOSS(Full-Shape)+BAO+BBN", mean=68.6, std=1.1)#
10 update from https://arxiv.org/pdf/2002.04035.pdf
```

As estimativas foram combinadas para formar conjuntos específicos, como aqueles que incluem cefeídeas (`local_ceph`), Gigantes Ramo Horizontal (TRGB - `local_trgb`), MIRAS (`local_miras`), e uma combinação de todas as fontes (`local_all`). Logo após, a função `compute_tension` foi usada para calcular a "tensão" entre as estimativas locais da constante H_0 e a estimativa do Planck.

```
1 # combine estimates
2 local_ceph = combine_estimates([tdc, shoes, masers], flat)
3 local_ceph.name = "with Cepheids"
4 local_trgb = combine_estimates([tdc, cchp, masers], flat)
5 local_trgb.name = "with TRGB"
6 local_miras = combine_estimates([tdc, miras, masers], flat)
7 local_miras.name = "with MIRAS"
8 local_all = combine_estimates([tdc, shoes, cchp, miras, masers, sbf, bTF
9 ], flat)
10 local_all.name = "combining all"
11 # compute the tension wrt Planck
```

```

11 tension_local_ceph = compute_tension(planck, local_ceph)
12 tension_local_trgb = compute_tension(planck, local_trgb)
13 tension_local_miras = compute_tension(planck, local_miras)
14 tension_local_all = compute_tension(planck, local_all)

```

O último passo, consistiu através da utilização da biblioteca `matplotlib` na construção de um gráfico que permitisse a visualização das estimativas da constante H_0 a partir de diferentes fontes e analisando a tensão em relação à estimativa do Planck. Para tanto, definimos as configurações das dimensões da figura, seus rótulos, seus eixos, rótulos e incerteza, esquema de cores e legendas:

```

1 # Define o tamanho da figura:
2 plt.figure(figsize=(8.0, 9.4), dpi=200)
3 plt.subplots_adjust(left=0.01, right=0.99, bottom=0.20, top=0.92)
4 ax = plt.subplot(1, 1, 1)
5 #Define os rotulos dos eixos da figura:
6 plt.xlabel(r"$H_0\ \text{[km}\,s^{-1}\,]\text{Mpc}^{-1}$", fontsize=24)
7 plt.yticks(fontsize=14)
8 plt.xticks(fontsize=20)
9 #Cria listas contendo as estimativas (xs) e posicoes verticais (ys), e
   cores para as fontes de dados:
10 xs = [planck, des, boss, shoes, cchp, miras, tdc, masers, sbf, bTF,
       local_all, local_ceph, local_trgb, local_miras]
11 ys = [17.3, 16.0, 14.8, 13.3, 12.0, 10.7, 9.4, 8.1, 6.8, 5.5, 4, 3, 2,
       1]
12 colors = ['peru', 'grey', 'mediumturquoise', 'royalblue',
13          'crimson', 'purple', 'limegreen', 'goldenrod',
14          'teal', 'darkorange', 'black', 'cornflowerblue',
15          'palevioletred', 'plum']
16
17 #Plotagem das Estimativas e Linhas de Incerteza:
18 for x, y, c in zip(xs, ys, colors):
19     pcs = x.getpcs()
20     plt.scatter(pcs[1], y, s=80, edgecolor=c, c=c,
21               label=x.name)
22     plt.plot([pcs[0], pcs[2]], [y, y], linewidth=2.5,
23             color=c)
24 if x.name not in ['combining all', 'with Cepheids', 'with TRGB', 'with
   MIRAS']:
25     delaytext = r"%%.1f^{+%.1f}_{-%.1f}$" % (pcs[1],
26     pcs[2]-pcs[1], pcs[1]-pcs[0])
27     plt.annotate(delaytext, xy=(pcs[1], y + 0.18),
28                 color = c, horizontalalignment="center", fontsize=17)
29     plt.annotate(x.name, xy=(pcs[1], y-0.4), color = "
   dimgrey", horizontalalignment="center", fontsize=10)
30 else:
31     delaytext = r"%%.1f^{+%.1f}_{-%.1f}$" % (pcs[1], pcs

```

```

[2]-pcs[1], pcs[1]-pcs[0])
28         plt.annotate(delaytext, xy=(pcs[1], y + 0.19), color
    = c, horizontalalignment="center", fontsize=14)
29         plt.annotate(x.name, xy=(pcs[1]+ 2.2, y + 0.19),
    color = "dimgrey", horizontalalignment="center", fontsize=10)
30 # Adiciona uma anotacao principal no grafico indicando que a analise
    compara as estimativas do Planck com o conjunto de dados "Late".
31 xpos = 79.4
32 plt.annotate("Planck vs. Late", xy=(xpos, 4.6), color = "dimgrey",
33             horizontalalignment="center", fontsize=11)
34
35 plt.annotate(r"$\rm{%.1f}\sigma$" % tension_local_all, xy=(xpos, 4.18),
36             color = "black",
37             horizontalalignment="center", fontsize=14)
38 plt.annotate(r"$\rm{%.1f}\sigma$" % tension_local_ceph, xy=(xpos, 3.18),
39             color = "black",
40             horizontalalignment="center", fontsize=14)
41 plt.annotate(r"$\rm{%.1f}\sigma$" % tension_local_trgb, xy=(xpos, 2.18),
42             color = "black",
43             horizontalalignment="center", fontsize=14)
44 plt.annotate(r"$\rm{%.1f}\sigma$" % tension_local_miras, xy=(xpos, 1.18)
45             , color = "black",
46             horizontalalignment="center", fontsize=14)
47 #Ajuste de Layout e Salvamento da Figura:
48 plt.tight_layout()
49 #plt.savefig('H0_tension.pdf', dpi=200)
50 #plt.savefig('H0_tension.png', dpi=200)
51 plt.savefig("fig12_george.png")

```

Ao final, foi obtida na Figura 12 que permite a visualização das estimativas da constante de Hubble provenientes de diferentes fontes, enfatizando a "tensão" com a estimativa do Planck. A Figura 12 apresenta três blocos. No primeiro bloco, na parte superior, são apresentados os experimentos que utilizam a física do universo primordial. No segundo bloco, o central, encontram-se os experimentos que fazem uso da física do universo recente. No terceiro bloco, na parte inferior, são realizadas as combinações entre as medidas.

Cada ponto representa uma estimativa, com linhas indicando as incertezas. É notório que, embora as incertezas dos experimentos tenham diminuído ao longo do tempo, as medidas que utilizam a física do universo recente apresenta uma disparidade com os dados dos experimentos que usa os dados do universo primordial entre os valores apresentados para a constante de H_0 é justamente essa disparidade a chamada **Tensão de Hubble**. Até o momento, não existe uma solução consensual que justifique essa disparidade de valores. No entanto, os cosmólogos indicam que a implementação de novos experimentos e a melhoria contínua das observações com relação aos valores têm o potencial de contribuir para a resolução desse desafio [14].

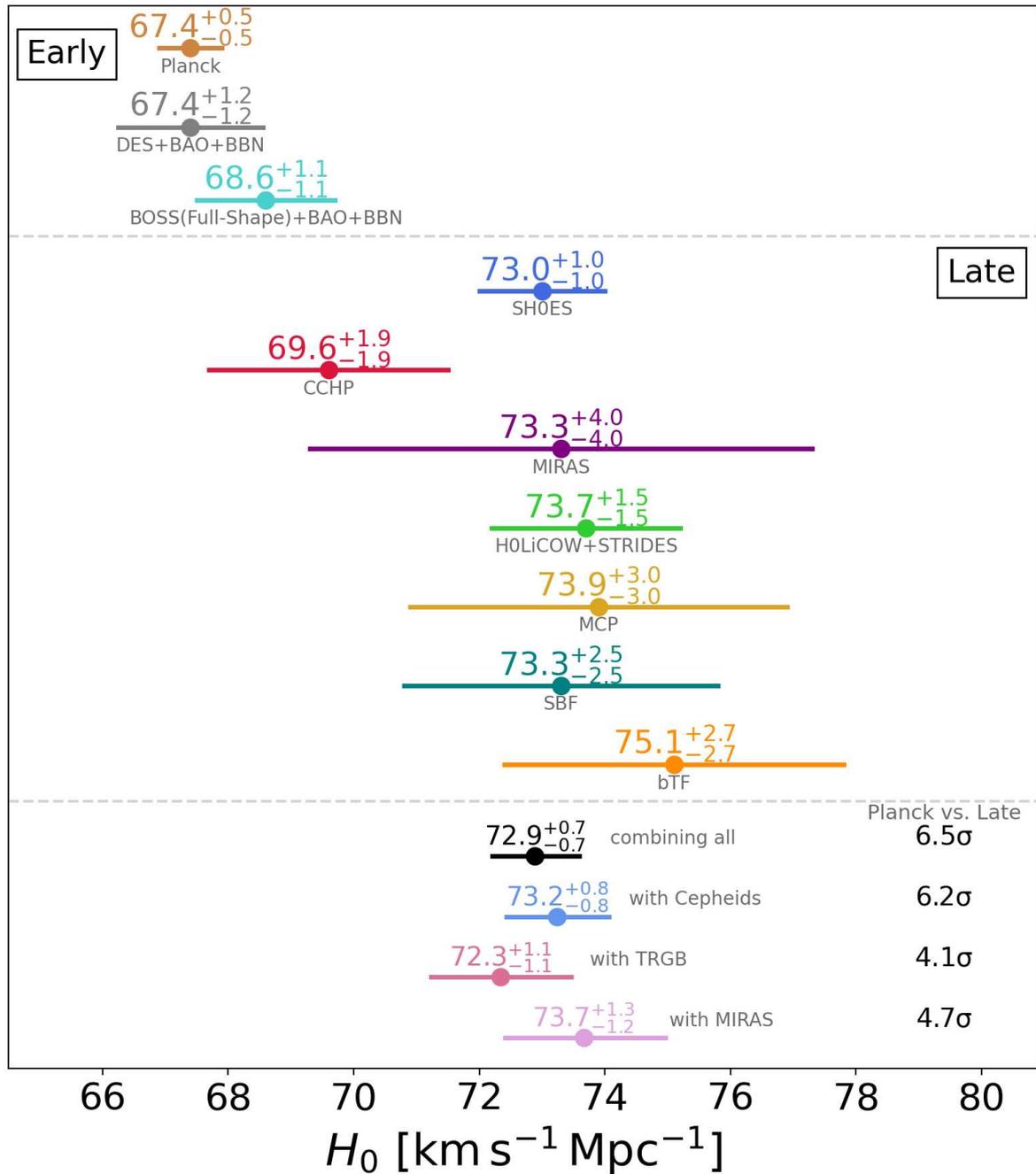
flat – Λ CDM

Figura 12 – A discrepância observada nas diferentes medições da Constante de Hubble H_0 hoje, conhecida como Tensão de Hubble [9].

6 Conclusões

A compreensão das principais técnicas utilizadas dentro da cosmologia para a determinação da constante de Hubble desempenha um papel extremamente importante, pois ela é um indicador cosmológico que define a escala, o tamanho e a idade do universo; é tida como uma das formas mais diretas de quantificar como o universo evolui através do tempo.

Existem várias formas e técnicas que permitem a quantificação do valor da constante de Hubble, mas há duas abordagens principais, consideradas como as maneiras clássicas. A primeira utiliza Velas Padrão, como as estrelas cefeidas, cuja luminosidade conhecida permite calcular distâncias astronômicas. A segunda abordagem baseia-se na análise da Radiação Cósmica de Fundo por meio de satélites como WMAP e Planck, explorando padrões na radiação remanescente do Big Bang para inferir propriedades do universo primordial.

O progresso tecnológico trouxe notáveis melhorias na coleta de dados astronômicos. No entanto, os resultados obtidos para a constante de Hubble revelam uma discrepância entre as técnicas aplicadas localmente e aquelas empregadas no estudo do universo primordial, denominada Tensão de Hubble. A existência de diferenças significativas nos valores obtidos sinaliza desafios para a compreensão da expansão do universo.

Através da Tensão de Hubble e suas implicações, trabalhar com o acesso de dados astronômicos dentro da linguagem Python. Para essa análise, foram empregados pacotes de software como `MATPLOTLIB`, `NUMPY` e `PANDAS` para manipulação e visualização dos dados. Além disso, para análise estatística e modelagem, utilizamos pacotes como `SCIPY` e `SCIKIT-LEARN`. Para acessar e processar dados astronômicos específicos, contamos com o auxílio dos pacotes `ASTROPY` e `ASTROQUERY`. Nesse processo, foi possível compreender as implicações e o papel da constante de Hubble na cosmologia.

Quanto à Tensão de Hubble, podemos afirmar que tem sido objeto de intensa investigação, evidenciados por trabalhos recentes, como exemplo, o projeto SH0ES. No entanto, até o momento, não há um consenso entre as medições da constante de Hubble, as quais podem ser obtidas por diferentes métodos, como o uso de Velas Padrão e a Radiação Cósmica de Fundo.

Existe possibilidade de efeitos sistemáticos desconhecidos, como um fator que tem contribuído para a falta de concordância entre os resultados para a constante de Hubble. Com o avanço contínuo nos equipamentos de investigação e o aumento na quantidade de trabalhos dedicados a esse tema, podemos observar perspectivas promissoras para a resolução da Tensão de Hubble. Assim, é possível pensar que, em um futuro próximo, haverá uma compreensão clara sobre o entendimento desse fenômeno.

Referências

- 1 HUBBLE, E. A relation between distance and radial velocity among extra-galactic nebulae. *Proceedings of the national academy of sciences*, National Acad Sciences, v. 15, n. 3, p. 168–173, 1929.
- 2 WENGER, M. et al. The SIMBAD astronomical database. *Astronomy and Astrophysics Supplement Series*, EDP Sciences, v. 143, n. 1, p. 9–22, apr 2000.
- 3 HOSTIUIJNE, W. et al. The effect of metallicity on cepheid period-luminosity relations from a baade-wesselink analysis of cepheids in the milky way and magellanic clouds. *A&A*, v. 620, p. A99, 2018. Disponível em: <<https://doi.org/10.1051/0004-6361/201833263>>.
- 4 OCHSENBEIN, F.; BAUER, P.; MARCOUT, J. The VizieR database of astronomical catalogues. *Astronomy and Astrophysics Supplement Series*, EDP Sciences, v. 143, n. 1, p. 23–32, apr 2000.
- 5 RIESS, A. G. et al. A comprehensive measurement of the local value of the hubble constant with 1 km ssup-1/sup mpcsup-1/sup uncertainty from the hubble space telescope and the SH0es team. *The Astrophysical Journal Letters*, American Astronomical Society, v. 934, n. 1, p. L7, jul 2022.
- 6 HOSTIUIJNE, J. H. J.; HOOGERWERF, R.; DE ZEEUW, P. T. A Hipparcos study of the Hyades open cluster. Improved colour-absolute magnitude and Hertzsprung-Russell diagrams. v. 367, p. 111–147, fev. 2001.
- 7 BROUT, D. et al. The pantheon analysis: Cosmological constraints. *The Astrophysical Journal*, American Astronomical Society, v. 938, n. 2, p. 110, oct 2022.
- 8 AGHANIM and N. et al. iplanck/i2018 results. *Astronomy & Astrophysics*, EDP Sciences, v. 641, p. A6, sep 2020.
- 9 BONVIN, V.; MILLON, M. *H0LiCOW H0 tension plotting notebook*. Zenodo, 2020. Disponível em: <<https://doi.org/10.5281/zenodo.3635517>>.
- 10 EVOLÇÃO Final das Estrelas. Disponível em: <<http://astro.if.ufrgs.br/estrelas/escola.htm>>.
- 11 SARAIVA, M. d. F. O. *Astronomia e Astrofísica*. [S.l.]: Editora Livraria da Física, 2004.
- 12 RYDEN, B. *Introduction to cosmology*. [S.l.]: Cambridge University Press, 2017.
- 13 MARTINS, R. A. *O Universo: teorias sobre sua origem e evolução. 2. ed.* [S.l.]: Editora Livraria da Física, 2012.
- 14 YUAN, Z. A critical review on hubble tension. MSc. Thesis, 2021.
- 15 ROSENFELD, R. O que é cosmologia? ao olhar o céu em uma noite sem.
- 16 SOARES, D. O universo estático de einstein. *Revista Brasileira de Ensino de Física*, SciELO Brasil, v. 34, 2012.

- 17 HALLIDAY, D.; RESNICK, R.; WALKER, J. *Fundamentos de Física: Gravitação, Ondas E Termodinâmica. Vol. 2*. [S.l.]: Grupo Gen-LTC, 2000.
- 18 HUNTER, J. D. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, IEEE Computer Society, v. 9, n. 03, p. 90–95, 2007.
- 19 HARRIS, C. R. et al. Array programming with NumPy. *Nature*, Springer Science and Business Media LLC, v. 585, n. 7825, p. 357–362, set. 2020. Disponível em: <<https://doi.org/10.1038/s41586-020-2649-2>>.
- 20 MCKINNEY, W. et al. Data structures for statistical computing in python. In: AUSTIN, TX. *Proceedings of the 9th Python in Science Conference*. [S.l.], 2010. v. 445, p. 51–56.
- 21 PRICE-WHELAN, A. M. et al. The astropy project: Building an open-science project and status of the v2. 0 core package. *The Astronomical Journal*, IOP Publishing, v. 156, n. 3, p. 123, 2018.
- 22 GINSBURG, A. et al. astroquery: An Astronomical Web-querying Package in Python. v. 157, p. 98, mar. 2019.
- 23 LEWIS, A. *GetDist: a Python package for analysing Monte Carlo samples*. arXiv, 2019. Disponível em: <<https://arxiv.org/abs/1910.13970>>.
- 24 HOSTI, B.P, t. e. g. e. s. v. 2021. Disponível em: <<https://www.espacotempo.com.br/evolucao-estelar-parte-4-gigantes-e-supergigantes-vermelhas/>>. Acesso em: 12 de Janeiro 2023.
- 25 HOGG, D. W. Distance measures in cosmology. *arXiv preprint astro-ph/9905116*, 1999.
- 26 EXPLANATION of the cosmic distance ladder. 2011. Disponível em: <<https://www.uwa.edu.au/science/-/media/Faculties/Science/Docs/Explanation-of-the-cosmic-distance-ladder.pdf/>>.
- 27 BATALHA, A.; CAMPOS, L. Estrelas cefeidas e rr lyrae.
- 28 VIRTANEN, P. et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, v. 17, p. 261–272, 2020.
- 29 ORTIZ, R. *Evolução Estelar-I*. [S.l.]: Notas de aula de Aperfeiçoamento em Astronomia para, 2014.
- 30 FREEDMAN, W. L. Measurements of the hubble constant: tensions in perspective. *The Astrophysical Journal*, IOP Publishing, v. 919, n. 1, p. 16, 2021.
- 31 ALGARVE. *NOVA MEDIÇÃO DA CONSTANTE DE HUBBLE FAZ CRESCER MISTÉRIO DA EXPANSÃO DO UNIVERSO*. 2019. Disponível em: <https://www.ccvalg.pt/astronomia/noticias/2019/07/19_constante_hubble.html>. Acesso em: 12 de Janeiro 2023.
- 32 SATO, e. *O que são supernovas do tipo Ia?* 2020. Disponível em: <<https://www.institutoprincipia.org/post/o-que-s3A3o-supernovas-do-tipo-ia>>.

- 33 RODRIGUES, G. D. S. Supernovas do tipo ia e suas implicações cosmológicas.
- 34 REIS, R. R.; SIFFERT, B. B. Supernovas do tipo ia e a expansão do universo. *Cadernos de Astronomia*, v. 3, n. 1, p. 21–21, 2022.
- 35 SOUZA, R. *Medindo a energia escura com Supernovas do tipo Ia*. 2000. Disponível em: <<https://www.if.ufrj.br/~ioav/rafael2001.html>>. Acesso em: 11 de Janeiro 2023.
- 36 GUIMARÃES, L. F. de O. A física da radiação cósmica de fundo em micro-ondas. *Cadernos de Astronomia*, v. 4, n. 2, p. 62–88, 2023.
- 37 VILLELA, T.; FERREIRA, I.; WUENSCHÉ, C. A. Cosmologia observacional: a radiação cósmica de fundo em microondas. *Revista USP*, n. 62, p. 104–115, 2004.
- 38 COSMOLOGIA: Espectro de potência da RCFM. Disponível em: <<http://www.das.inpe.br/cosmo/intro-cosmo/espectro-de-potencia.php#:~:text=a%20qual%20expressa%20as%20flutua%C3%A7%C3%B5es,%CE%B1m%20apresentam%20m%C3%A9dia%20nula>>.
- 39 TROTTA, R. Bayes in the sky: Bayesian inference and model selection in cosmology. *Contemporary Physics*, Taylor & Francis, v. 49, n. 2, p. 71–104, 2008.
- 40 PLANCK Legacy Archive. Disponível em: <<https://pla.esac.esa.int/pla/#home>>.
- 41 KOMATSU, E. et al. Five-year wilkinson microwave anisotropy probe* observations: cosmological interpretation. *The Astrophysical Journal Supplement Series*, IOP Publishing, v. 180, n. 2, p. 330, 2009.
- 42 JAROSIK, N. et al. *The Astrophysical Journal Supplement Series*, 192, 14. 2011.
- 43 HINSHAW, G. et al. Nine-year wilkinson microwave anisotropy probe (wmap) observations: cosmological parameter results. *The Astrophysical Journal Supplement Series*, IOP Publishing, v. 208, n. 2, p. 19, 2013.
- 44 A EVOLUÇÃO ESTELAR I. Disponível em: <<https://www.iag.usp.br/siae98/universo/evolucao.htm>>.

APÊNDICE A – Estrelas

A.1 Estrelas

¹ Uma estrela é uma esfera de gás ionizado autogravitante que gera energia em seu centro através de reações de fusão nuclear onde no processo hidrogênio se transforma em hélio e, posteriormente, em elementos mais pesados. As estrelas, sem exceção, nascem, vivem e ... morrem! Sua vida e sua morte dependem de diversos parâmetros, entre eles a composição química e, principalmente, a massa. As estrelas tem massas entre 0,08 e 100 vezes a massa do Sol ($M_{\odot} = 1,9891 \times 10^{30} \text{ kg}$) e as estrelas normais, não colapsadas, apresentam temperaturas efetivas entre 2500K e 30 000K [44, 11].

A.2 Evolução estelar

O nascimento de uma estrela ocorre nas nebulosas, imensas nuvens de gás e poeira. Devido a ação da força da gravidade, as moléculas que compõem essa nuvem vão sendo atraídas umas pelas outras até formar uma proto-estrela, esse processo causa o aumento na temperatura. Quando a temperatura no núcleo se torna suficiente alta (cerca de 8 milhões K) iniciam-se as reações de fusão nuclear e a transformação de Hidrogênio em Hélio no núcleo estelar.

Os passos evolutivos seguintes da estrela vão depender de sua massa inicial e da sua configuração (se ela faz parte de um sistema binário ou múltiplo), na Figura 13 é apresentado um esquema da evolução estelar. Se a estrela iniciar com massa entre 0,8 e 10 M_{\odot} após o consumo do hidrogênio do seu centro ela passa pela fase da gigante, depois supergigante e após ejetar sua massa em uma nebulosa planetária terminará sua vida como uma anã branca, com massa na ordem de 0,8 M_{\odot} e raio 10.000 km.

Se a estrela iniciar sua vida com massa entre 10 e 25 M_{\odot} , após a fase da supergigante ejetará a maior parte de sua massa em uma explosão de supernova terminando sua vida como estrela de nêutrons com temperatura superficial acima de 1 milhão de K, massa de cerca de 1,4 M_{\odot} e raio cerca de 20 km. Caso a estrela possua um campo magnético ela emitirá luz direcionada em um cone em volta dos polos magnéticos.

¹ Este material é um recorte do Capítulo 22 do Livro SARAIVA, Maria de Fátima Oliveira. *Astronomia e Astrofísica*. Editora Livraria da Física, 2004.

Se a estrela iniciar sua vida com massa entre 25 e 100 M_{\odot} , após a fase de supernova restará um buraco negro, com massa na ordem de 6 M_{\odot} e raio do horizonte de cerca de 18 km. O raio do horizonte, ou raio de Schwarzschild [Karl Schwarzschild (1873-1916)], é a distância ao buraco negro dentro da qual nem a luz escapa: $R_{Sch} = 2GM/c^2$, se a estrela iniciar sua vida com massa acima de 100 M_{\odot} , ela ejetará a maior parte de sua massa e depois evoluirá como uma estrela de até 100 M_{\odot} .

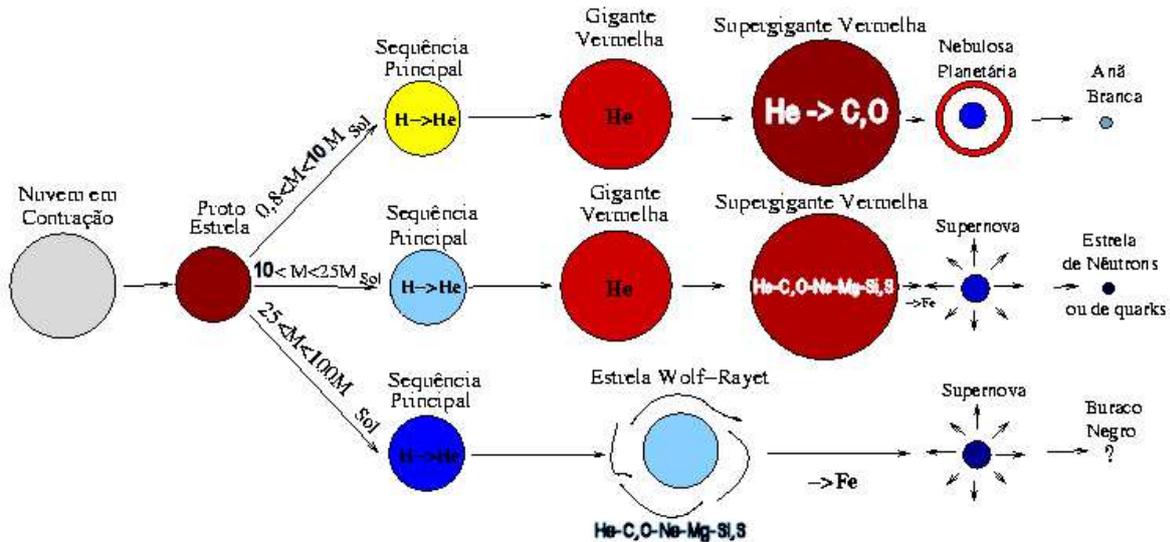


Figura 13 – Esquema de evolução estelar, fora de escala [10].

Os elementos químicos produzidos das reações nucleares que ocorrem no interior das estrelas e posteriormente liberados durante as explosões estelares, como carbono e outros elementos mais tarde colapsam e formam estruturas como planetas, satélites naturais e até mesmo seres vivos.

A.3 Diagrama HR

O Diagrama de Hertzsprung Russell, conhecido como diagrama HR é uma ferramenta útil que nos permite compreender e acompanhar os passos evolutivos da vida das estrelas e está baseado na relação existente entre a luminosidade de uma estrela e sua temperatura efetiva.

A Figura 14 exibe um diagrama HR para um conjunto de estrelas nas proximidades do Sol. A luminosidade (ou magnitude absoluta) como a temperatura superficial de uma estrela, são características determináveis para estrelas de distâncias conhecidas: a primeira pode ser encontrada a partir da magnitude aparente, e a segunda a partir de sua cor ou tipo espectral. A temperatura cresce para a esquerda, e a luminosidade para cima. A maior parte das estrelas

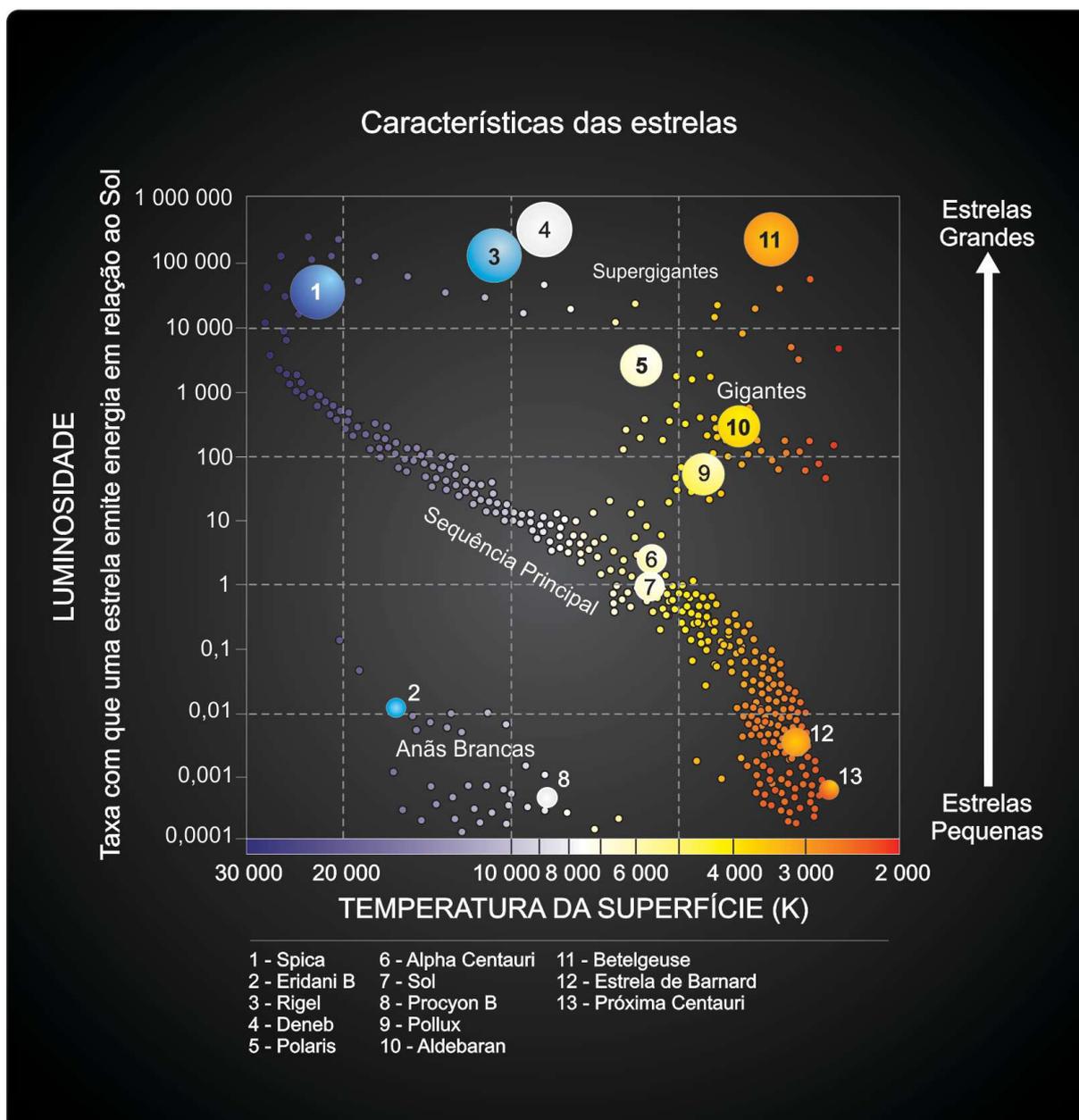


Figura 14 – Representação de um diagrama HR, apresentando a localização de algumas estrelas conhecidas [11].

está alinhada ao longo de uma estreita faixa na diagonal que vai do extremo superior esquerdo (estrelas quentes e muito luminosas), até o extremo inferior direito (estrelas frias e pouco luminosas). Essa faixa é chamada sequência principal, nela as estrelas permanecem a maior parte de sua vida. O fator que determina onde uma estrela se localiza na sequência principal é a sua massa. As estrelas da sequência principal têm, por definição, classe de luminosidade V e são chamadas de anãs. As estrelas que se concentram no canto inferior esquerdo (estrelas quentes e pouco luminosas): são chamadas anãs, as anãs brancas cobrem um intervalo de temperatura e cores que abrange desde as

mais quentes, que são azuis ou brancas, e têm temperatura superficiais de até 200.000K, até as mais frias, que são vermelhas, e têm temperaturas superficiais de apenas 3500 K.

As massas das estrelas em um sistema binário podem ser determinadas usando a Terceira Lei de Kepler e da aplicação notou-se que as massas das estrelas aumentam de baixo para cima ao longo da sequência principal, sendo assim, é possível construir uma relação massa-luminosidade que permite estimar a massa de uma estrela em função do seu tipo espectral.

Para estrelas com massas M grandes, maiores do que $3 M_{\odot}$, a luminosidade é proporcional ao cubo da massa; para massas pequenas, menores do que $0,5 M_{\odot}$, a luminosidade é proporcional potência 2,5 da massa. A luminosidade das estrelas variam entre 0,08 e 100 massas solares e a sua luminosidade variam entre 10^{-4} e 10^{+4} vezes a luminosidade do sol.

Uma das aplicações importantes do diagrama HR é a determinação de distâncias estelares. Se uma estrela tem o tipo espectral G2 e seu espectro que indica que ela se encontra na sequência principal. sua luminosidade pode ser determinada pelo diagrama HR e será em torno de $1L_{\odot}$ ($M=+5$) Conhecendo-se sua magnitude aparente, a sua distância pode ser calculada a partir da relação do módulo da distância:

$$(m - M) = -5 + 5 \text{ Log } d \rightarrow d = 10^{\frac{(m-M+5)}{5}} \quad (\text{A.1})$$

onde: m é a magnitude aparente, M magnitude absoluta e d é a distância em parsecs. Além da classe espectral sozinha é preciso conhecer também a luminosidade da estrela de forma única e também sua classe.

APÊNDICE B – Códigos Python

B.1 Códigos da Figura 1

```

1 import glob
2 from IPython.core.interactiveshell import InteractiveShell
3 from IPython.display import display
4 #-----
5 from adjustText import adjust_text
6 from astroquery.simbad import Simbad
7 import astropy.coordinates as coord
8 import astropy.units as u
9
10 import matplotlib.pyplot as plt
11 import numpy as np
12 import pandas as pd
13 import scipy as sp
14 from sklearn.linear_model import LinearRegression
15 import statsmodels.formula.api as smf
16 from sklearn.linear_model import Ridge
17 from sklearn.preprocessing import PolynomialFeatures, SplineTransformer
18 from sklearn.pipeline import make_pipeline
19 import getdist
20 from getdist import plots as gdplt
21 #-----
22 %matplotlib inline
23 plt.rcParams["figure.figsize"] = (16, 8)
24 InteractiveShell.ast_node_interactivity = "all"
25 s = Simbad(('ra(d;icrs;2000)', 'dec(d;icrs;2000)', 'rv_value', 'distance',
26           'typed_id'))
27 s.add_votable_fields
28 s.remove_votable_fields('coordinates')
29 response = s.query_objects(hubble_data['OBJECT'])
30 df_data = response.to_pandas()[['MAIN_ID', 'RA_d_icrs_2000', '
31     DEC_d_icrs_2000', 'RV_VALUE', 'Distance_distance', 'TYPED_ID']].rename
32     (columns={'RA_d_icrs_2000': 'RAJ2000', 'DEC_d_icrs_2000': 'DEJ2000', '
33     RV_VALUE': 'V_r', 'Distance_distance': 'distance', 'TYPED_ID': 'OBJECT'
34     })
35 df_hubble = pd.merge(hubble_data, df_data, on='OBJECT', how='left')
36 #
37 ra = coord.Longitude(df_hubble.RAJ2000, unit=u.deg).wrap_at(180*u.deg).
38     radian
39 dec = coord.Latitude(df_hubble.DEJ2000, unit=u.deg).radian
40 OBJECTS = df_hubble['OBJECT'].to_numpy()

```

```

35 #Visualizando as galxias no c u.
36 fig = plt.figure(figsize=(16, 9))
37 ax1 = fig.add_subplot(111, projection="mollweide")
38 ax1.set_title('Galaxias Utilizadas por Hubble')
39 ax1.scatter(x=-ra, y=dec, s=12, c='darkblue')
40 texts = []
41 for ra, dec, nome in zip(-ra, dec, OBJECTS):
42     texts.append(plt.text(ra, dec, nome, color='blue', fontsize=10))
43 adjust_text(texts, arrowprops=dict(arrowstyle="-", color='lightgray', lw
44     =0.1))
45 ax1.set_xticklabels(['14h', '16h', '18h', '20h', '22h', '0h', '2h', '4h', '6h',
46     '8h', '10h'])
47 ax1.set_xlabel(r'$\alpha$(h): Ascen o Reta')
48 ax1.set_ylabel(r'$\delta(^{\circ})$: Declina o')
49 ax1.grid(True)
50 plt.savefig("fig01_george.png")
51 plt.tight_layout();

```

B.2 Códigos da Figura 2

```

1 import glob
2 from IPython.core.interactiveshell import InteractiveShell
3 from IPython.display import display
4 #-----
5 from adjustText import adjust_text
6 from astroquery.simbad import Simbad
7 import astropy.coordinates as coord
8 import astropy.units as u
9
10 import matplotlib.pyplot as plt
11 import numpy as np
12 import pandas as pd
13 import scipy as sp
14 from sklearn.linear_model import LinearRegression
15 import statsmodels.formula.api as smf
16 from sklearn.linear_model import Ridge
17 from sklearn.preprocessing import PolynomialFeatures, SplineTransformer
18 from sklearn.pipeline import make_pipeline
19 import getdist
20 from getdist import plots as gdplt
21 #-----
22 %matplotlib inline
23 plt.rcParams["figure.figsize"] = (16, 8)
24 InteractiveShell.ast_node_interactivity = "all"
25 x, y, texto = df_hubble.distance, df_hubble.V_r, df_hubble.OBJECT
26 fig, ax = plt.subplots(figsize=(16,9))

```

```

27 ax.set_title('Diagrama de Hubble atuais')
28 ax.scatter(df_hubble['distance'], df_hubble['V_r'], marker='o', s=15,
           color='blue', label="dados atuais")
29 for a, b, c in zip(x, y, texto):
30     ax.text(a, b, c)
31
32 ax.plot(distance, model_current.predict(distance),
33         label=r'Ajuste Linear H={:.1f} Km/MPc s'.format(model_current.
           coef_[0]),
34         linewidth=2, color='brown')
35 ax.set_xlabel('d (MPc)')
36 ax.set_ylabel('v (Km/s)')
37 plt.legend()
38 plt.grid(True)
39 plt.show();

```

B.3 Códigos da Figura 3

```

1 import glob
2 from IPython.core.interactiveshell import InteractiveShell
3 from IPython.display import display
4 #-----
5 from adjustText import adjust_text
6 from astroquery.simbad import Simbad
7 import astropy.coordinates as coord
8 import astropy.units as u
9
10 import matplotlib.pyplot as plt
11 import numpy as np
12 import pandas as pd
13 import scipy as sp
14 from sklearn.linear_model import LinearRegression
15 import statsmodels.formula.api as smf
16 from sklearn.linear_model import Ridge
17 from sklearn.preprocessing import PolynomialFeatures, SplineTransformer
18 from sklearn.pipeline import make_pipeline
19 import getdist
20 from getdist import plots as gdplt
21 #-----
22 %matplotlib inline
23 plt.rcParams["figure.figsize"] = (16, 8)
24 InteractiveShell.ast_node_interactivity = "all"
25 param_deSitter = { "Om_L" : 1.,
26                   "Om_M" : 0.,
27                   "Om_K" : 0. }
28 param_LCDM = { "Om_L" : 0.7,

```

```

29         "Om_M" : 0.3,
30         "Om_K" : 0. }
31 param_closed = { "Om_L" : 0.,
32                 "Om_M" : 6.,
33                 "Om_K" : -5. }
34 # Precisa primeiro criar o dicionario e popular suas chaves se for fazer
    dicionario aninhado.
35 COSMOS = {}
36 COSMOS["deSitter"] = param_deSitter
37 COSMOS["LCDM"] = param_LCDM
38 COSMOS["closed"] = param_closed
39 HO = 70
40
41 from astropy.cosmology import LambdaCDM
42 from astropy.cosmology import FlatLambdaCDM
43
44 cosmos_astropy = { key: LambdaCDM(HO = HO, Om0 = value["Om_M"], Ode0 =
    value["Om_L"])
45                   for key, value in COSMOS.items()}
46
47 import matplotlib.cm as cm
48
49 zz = np.linspace(0,100,1000)
50 fig, ax = plt.subplots(figsize = (16,9))
51 colors = iter(cm.Set2(np.arange(len(COSMOS))))
52 for key, cosmo in cosmos_astropy.items():
53     c = next(colors)
54     ax.plot(zz, cosmo.angular_diameter_distance(zz), c = c, linestyle
    = "solid", label = r'$D_A$ - {}'.format(key))
55     ax.plot(zz, cosmo.comoving_distance(zz), c = c, linestyle = "dashed"
    , label = r'$D_C$- {}'.format(key))
56     ax.plot(zz, cosmo.luminosity_distance(zz), c = c, linestyle = "
    dotted", label = r'$D_L$- {}'.format(key))
57 ax.set_title("Dist ncias Cosmol gicas")
58 ax.set_xlabel("$z$")
59 ax.set_ylabel("$\mathrm{Dist ncia}$; (Mpc)$")
60 ax.set_yscale("log")
61 ax.set_xscale("log")
62 ax.legend(loc='lower center', bbox_to_anchor=(0.5, -0.2), ncol=3,
    fancybox=True, shadow=True)
63 plt.grid()
64 plt.savefig("fig06_george.png", bbox_inches='tight')
65 plt.show();

```

B.4 Códigos da Figura 4

```

1 import glob
2 from IPython.core.interactiveshell import InteractiveShell
3 from IPython.display import display
4 #-----
5 from adjustText import adjust_text
6 from astroquery.simbad import Simbad
7 import astropy.coordinates as coord
8 import astropy.units as u
9
10 import matplotlib.pyplot as plt
11 import numpy as np
12 import pandas as pd
13 import scipy as sp
14 from sklearn.linear_model import LinearRegression
15 import statsmodels.formula.api as smf
16 from sklearn.linear_model import Ridge
17 from sklearn.preprocessing import PolynomialFeatures, SplineTransformer
18 from sklearn.pipeline import make_pipeline
19 import getdist
20 from getdist import plots as gdplt
21 #-----
22 %matplotlib inline
23 plt.rcParams["figure.figsize"] = (16, 8)
24 InteractiveShell.ast_node_interactivity = "all"
25 # Obtem lista de catalogos de acordo com chave de busca com astroquery
26 from astroquery.vizier import Vizier
27 catalog_list = Vizier.find_catalogs('Gieren W. , Storm J., Konorski P',
28     max_catalogs=1000)
29
30 # Obtem cat logo espec fico de cefeidas na pequena nuvem de magalh es
31 SMC_cepheids = list(catalog_list.keys())[2]
32 Vizier.ROW_LIMIT = -1
33 catalog = Vizier.get_catalogs(SMC_cepheids)
34 # Converte tabela de cat logo para pandas dataframe
35 cepheids = catalog[0].to_pandas()
36 lightcurves = catalog[1].to_pandas()
37
38 def lightcurve(nn):
39     # seleciona estrela da lista nica
40     star = cepheids.ID.unique()[nn]
41     # Calcula per odo
42     PP = 10 ** (cepheids[cepheids.ID == star].logP.values)
43     # Dataframe para guardar informa es de curva de luz
44     df = lightcurves[lightcurves.ID == star].copy()
45     df["phase"] = (lightcurves._tab2_2 - 2450000)/PP - np.floor((
lightcurves._tab2_2 - 2450000)/PP)

```

```

46 # Preparando para fazer dados para dois per odos
47 _df = df.copy()
48 _df.phase = _df.phase + 1
49 df = pd.concat([df, _df])
50 # Ajuste de Spline
51 X_train = df.phase.values[:, np.newaxis]
52 X_plot = np.linspace(0, 2, 100)[:, np.newaxis]
53 y_train = df.Kmag.values
54 # B-spline with 8 + 3 - 1 = 6 basis functions
55 model = make_pipeline(SplineTransformer(n_knots=8, degree=3), Ridge(
alpha=1e-3))
56 model.fit(X_train, y_train)
57 y_plot = model.predict(X_plot)
58 return df.sort_values(by="phase"), X_plot.reshape(-1), y_plot
59
60 # Gráfico da curva de luz para as 4 primeiras estrelas
61 fig, ax = plt.subplots(ncols=2, nrows=2)
62 df, XX, YY = lightcurve(0)
63 ax[0, 0].scatter( df.phase, df.Kmag, label=df.ID.values[0], marker = "o"
)
64 ax[0, 0].plot( XX, YY, color="black", label="B-Spline interpolation")
65 ax[0, 0].legend()
66 df, XX, YY = lightcurve(1)
67 ax[1, 0].scatter( df.phase, df.Kmag, label=df.ID.values[0], marker = "o"
)
68 ax[1, 0].plot( XX, YY, color="black", label="B-Spline interpolation")
69 ax[1, 0].legend()
70 df, XX, YY = lightcurve(2)
71 ax[0, 1].scatter( df.phase, df.Kmag, label=df.ID.values[0], marker = "o"
)
72 ax[0, 1].plot( XX, YY, color="black", label="B-Spline interpolation")
73 ax[0, 1].legend()
74 df, XX, YY = lightcurve(3)
75 ax[1, 1].scatter( df.phase, df.Kmag, label=df.ID.values[0], marker = "o"
)
76 ax[1, 1].plot( XX, YY, color="black", label="B-Spline interpolation")
77 ax[1, 1].legend()
78 plt.savefig("fig03_george.png")
79 plt.show();

```

B.5 Códigos da Figura 5

```

1 import glob
2 from IPython.core.interactiveshell import InteractiveShell
3 from IPython.display import display
4 #-----

```

```

5 from adjustText import adjust_text
6 from astroquery.simbad import Simbad
7 import astropy.coordinates as coord
8 import astropy.units as u
9
10 import matplotlib.pyplot as plt
11 import numpy as np
12 import pandas as pd
13 import scipy as sp
14 from sklearn.linear_model import LinearRegression
15 import statsmodels.formula.api as smf
16 from sklearn.linear_model import Ridge
17 from sklearn.preprocessing import PolynomialFeatures, SplineTransformer
18 from sklearn.pipeline import make_pipeline
19 import getdist
20 from getdist import plots as gdplt
21 #-----
22 %matplotlib inline
23 plt.rcParams["figure.figsize"] = (16, 8)
24 InteractiveShell.ast_node_interactivity = "all"
25 import glob
26 from IPython.core.interactiveshell import InteractiveShell
27 from IPython.display import display
28 #-----
29 import matplotlib.pyplot as plt
30 import numpy as np
31 import pandas as pd
32 import scipy as sp
33 from sklearn.linear_model import LinearRegression
34 #-----
35 %matplotlib inline
36 plt.rcParams["figure.figsize"] = (16, 8)
37 InteractiveShell.ast_node_interactivity = "all"
38
39 data = pd.read_table("../data/raw/R22_orig19_NIR.wm31.out", header=None,
40                     sep="\s+", on_bad_lines='skip', skiprows=2)
41 data.columns = ["Host", "ra", "dec", "ID", "period", "V-I", "sigma_V", "
42                 H", "sigma_H", "metal", "HST"]
43
44 data["mw"] = data["H"] - 0.34*data["V-I"]
45 data["logP"] = np.log10(data["period"])
46 #data["sigma_mw"] = np.sqrt(data["sigma_H"]**2 + (0.34 * data["V-I"])**2
47     * ((0.02/0.34)**2 + (data["V-I"]/data["sigma_V"])**2))
48 data["sigma_mw"] = np.sqrt(data["sigma_H"]**2)
49
50 def fit_leavitt(data):
51     X = data[["logP", "metal"]]

```

```

49     y = data["mw"]
50     linear_regression = LinearRegression()
51     linear_regression.fit(X, y)
52     bw, zw = linear_regression.coef_
53     zpw = linear_regression.intercept_
54     logp_min, logp_max = data["logP"].min(), data["logP"].max()
55     logp_range = np.linspace(logp_min, logp_max, 100)
56     metal = data.metal.mean()*np.ones(100)
57     predict = pd.DataFrame({"logP": logp_range, "metal": metal})
58     YY = linear_regression.predict(predict)
59     return YY, logp_range, bw, zw, zpw
60
61     data = data[data.Host.isin(["LMC", "M31", "N4258", "N5584"])]
62
63     fig, ax = plt.subplots(ncols=4, nrows=1, figsize=(16, 4))
64     ii = 0
65     for host, df in data.groupby("Host"):
66         YY, logp, bw, zw, zpw = fit_leavitt(df)
67         kk = ii % 4
68         ax[kk].errorbar(df["logP"], df["mw"], df["sigma_mw"], label=host,
69                        fmt=".", alpha=0.6, linewidth=.3)
70         ax[kk].plot(logp, YY, color="red")
71         ax[kk].legend()
72         ii = ii + 1
73         ax[kk].invert_yaxis()
74     ax[0].set_ylabel(r"$m^W$")
75     fig.supxlabel(r'$Log\; P$')
76     plt.savefig("fig04_george.png")
77     plt.show();

```

B.6 Códigos da Figura 6

```

1 from astroquery.vizier import Vizier
2 from color_system import cs_hdtv, planck
3 cs = cs_hdtv
4 lam = np.arange(380., 781., 5)
5
6 vv = Vizier(columns=["HIP", "Vmag", "Plx", "B-V", "SpType", "Period"],
7               column_filters={"Vmag": "-5.0..8.0"})
8 vv.ROW_LIMIT = -1
9 hipparcos = vv.get_catalogs('I/239/hip_main')[0].to_pandas()
10 #varstars = vv.get_catalogs('I/239/hip_va_1')[0].to_pandas()
11 # limpando dataframe de colunas vazias e nan
12 df = hipparcos.applymap(lambda x: np.nan if isinstance(x, str) and x.
13                          isspace() else x)

```

```

13 df["Period"] = df["Period"].fillna(0)
14 df = df.dropna()
15 df = df[df.Plx > 0] # Algumas paralaxes são negativas!
16
17 df["Vmag"] = df["Vmag"].astype(float)
18 df["Plx"] = df["Plx"].astype(float)
19 df["B-V"] = df["B-V"].astype(float)
20 # Nova coluna com magnitude absoluta
21 df['M_V'] = df['Vmag'] + 5 * np.log10(df['Plx']/100.)
22 # Nova coluna com temperatura
23 def TT(BV):
24     """F rmula de Ballesteros"""
25     result = 4600 * ( 1 / (0.92 * BV + 1.7) + 1 / (0.92 * BV + 0.62))
26     return result
27 df["T"] = TT(df["B-V"])
28
29 df["color"] = df["T"].apply(lambda row: cs.spec_to_rgb(planck(lam, row),
30     out_fmt='html'))
31
32 df["Sp"] = df["SpType"].apply(lambda row: row[0:2])
33 df["Lum"] = df["SpType"].apply(lambda row: row[2:] if len(row)>=2 else
34     None )
35
36 Lum_classes = pd.DataFrame({"Lum": ["0", "Ia", "Iab", "Ib", "II", "III",
37     "IV", "V", "VI", "VII"],
38     "Tam": [10, 9, 8, 7, 6, 5, 4, 3, 2, 1]})
39
40 # Inclui luminosidade
41 df = df.merge(Lum_classes, left_on="Lum", right_on="Lum", how="left")
42 # Descarta estrelas sem luminosidade
43 df = df.dropna()
44
45 def chop(num):
46     power = int(10 ** np.floor(np.log10(num)))
47     result = int(np.round(num / power, 1) * power)
48     return result
49
50 fig, ax = plt.subplots(figsize=(8,10))
51 ax.set_facecolor("black")
52 ax.title.set_fontsize(26)
53 ax.set_xlabel(' ndice  de Cor B-V')
54 ax.set_ylabel('Magnitude Absoluta')
55
56 ax.scatter(df['B-V'], df['M_V'], s=df["Tam"]**2, edgecolors='none', c=df
57     ["color"])
58 # Cefeidas

```

```

56 ax.scatter(cefeidas['B-V'], cefeidas['M_V'], s=5 * cefeidas["Tam"]**2,
    edgecolors='none', marker = "*", c="red")
57
58 ax.tick_params(axis='both', labelsz=14)
59 ax.set_xlim(-0.5, 2.5)
60 ax.set_ylim(15, -12)
61 ax.grid()
62 ax.set_title('Diagrama Hertzsprung-Russel \n Cat logo Hipparcos')
63
64 ax2 = ax.twinx()
65 ax2.set_xticks( ax.get_xticks() )
66 ax2.set_xbound(ax.get_xbound())
67 ax2.set_xticklabels([chop(TT(xx)) for xx in ax.get_xticks()])
68 ax2.set_xlabel("Temperatura (K)")
69 plt.savefig("fig05_george.png")
70 plt.show();

```

B.7 Códigos da Figura 7

```

1
2 URL_SN = "https://raw.githubusercontent.com/PantheonPlusSHOES/
    DataRelease/main/Pantheon%2B_Data/4_DISTANCES_AND_COVAR/Pantheon%2
    BSHOES.dat"
3 IDSURVEY = {1:'SDSS', 4:'SNLS', 5:'CSP', 10:'DES', 15:'PS1MD', 18:'CNIA0
    .02', 50:'LOWZ/JRK07', 51:'LOSS1',
4             56:'SOUSA', 57:'LOSS2', 61:'CFA1', 62:'CFA2', 63:'CFA3S',
5             64:'CFA3K', 65:'CFA4p2', 66:'CFA4p3',
6             100:'HST', 101:'SNAP', 106:'CANDELS', 150:'FOUND'}
6 df = pd.read_table(URL_SN, sep="\s+")
7
8 zz = np.linspace(0.001, 3, 100)
9 mus = []
10 for key, cosmo in cosmos_astropy.items():
11     mu = 5 * np.log10(cosmo.luminosity_distance(zz).value) + 25
12     mus.append(mu)
13
14     cosmos_astropy.keys()
15
16     import matplotlib.ticker as ticker
17 labels = ["De-Sitter", r"$\Lambda$CDM", "Fechado"]
18 fig, ax = plt.subplots(figsize = (16,6))
19 for name, _df in df.groupby("IDSURVEY"):
20     ax.errorbar(x = _df.zHD, y = _df.MU_SHOES, yerr = _df.
21                 MU_SHOES_ERR_DIAG, label = IDSURVEY[name], fmt = ".")
22 for ii, mu in enumerate(mus):
    ax.plot(zz, mu, label=labels[ii])

```

```

23 ax.set_title("Supernovas")
24 ax.set_xlabel("$z$")
25 ax.set_ylabel("$\mu$")
26 ax.legend(loc='lower center', bbox_to_anchor=(0.5, -0.3), ncol=10,
           fancybox=True, shadow=True)
27 ax.set_xscale("log")
28 ax.xaxis.set_major_formatter(ticker.ScalarFormatter())
29 #ax.xaxis.set_minor_formatter(ticker.ScalarFormatter())
30 ax.ticklabel_format(style='plain', axis='x', useOffset=False)
31 plt.grid()
32 plt.savefig("fig07_george.png", bbox_inches='tight')
33 plt.show();

```

B.8 Códigos da Figura 8

```

1 pars_fiducial = camb.CAMBparams();
2 # Carrega os par metros
3 pars_fiducial.set_cosmology(H0=H0, ombh2=ombh2, omch2=omch2,
4                             mnu=mnu, omk=omk, tau=tau);
5 pars_fiducial.InitPower.set_params(As=As, ns=ns);
6 pars_fiducial.set_for_lmax(3000, lens_potential_accuracy=2);
7 # calculate results for these parameters
8 results_fiducial = camb.get_results(pars_fiducial);
9 powers_fiducial = results_fiducial.get_cmb_power_spectra(pars_fiducial,
10                                                         spectra=['total'], CMB_unit='muK');
11 Dll_fiducial=powers_fiducial['total'];
12 ell = np.arange(Dll_fiducial.shape[0]);
13 # Dois conjuntos de dados para o primeiro painel
14 ell_binned = target_data[2]['1']
15 Dll_binned = target_data[2]['best_fit']
16 # Erros n o sim tricos exigem formatar a lista para o errorplot.
17 Dll_binned_err = [target_data[2]['-DL'], target_data[2]['+DL']]
18 # Dados para o painel
19 D_interp = sp.interpolate.interp1d(ell[2:], Dll_fiducial[2:,0])
20 delta = (Dll_binned - D_interp(ell_binned))/D_interp(ell_binned)
21 #-----
22 fig = plt.figure(figsize=(16,6))
23 gs = gridspec.GridSpec(4, 1, hspace = 0)
24 ax1 = fig.add_subplot(gs[:3, 0])
25 ax1.errorbar(ell_binned, Dll_binned, yerr=Dll_binned_err, fmt='k.',
26             markersize=2, ecolor='red', label='best fit')
27 ax1.plot(ell, Dll_fiducial[:,0], label='Theory')
28 ax1.set_xlabel(r"$\ell$")
29 ax1.set_ylabel(r"$\ell(\ell+1)C_\ell / 2\pi \left[ \mu \mathrm{K}^2 \right]$")
30 ax1.set_xlim([20,2600])

```

```

29 ax1.set_xticks([30, 500, 1000, 1500, 2000, 2500])
30 ax1.label_outer()
31 ax1.secondary_yaxis("right")
32 plt.grid()
33 leg = plt.legend(frameon=True)
34 leg.get_frame().set_edgecolor("white")
35 leg.get_frame().set_alpha(.8)
36 ax2 = fig.add_subplot(gs[3, 0], sharex=ax1)
37 ax2.scatter(ell_binned, delta, marker='|', s=10, c='red')
38 ax2.plot(ell_binned, delta, color='blue')
39 ax2.secondary_yaxis("right")
40 ax2.set_ylabel(r'$\frac{\Delta D_1}{D_1}$')
41 plt.grid()
42 plt.savefig("fig08_george.png")
43 plt.show();

```

B.9 Códigos da Figura 9

```

1 factor = ell * (ell + 1.) / 2. / np.pi
2 Cll_fiducial = Dll_fiducial[2:,0] /factor[2:]
3 hp.mollview(hp.synfast(Cll_fiducial, nside = 512))
4 # Dimensions:
5 # - LL: the size of the patch in the x and y dimensions (in radians)
6 LL = 64. * np.pi/180 # ngulo
7 NN = 512 #pixels
8 map_flat = nmt.synfast_flat(NN, NN, LL, LL, np.array([Cll_fiducial]),
9 [0])
9 plt.figure()
10 plt.imshow(map_flat[0],
11 extent=[0, np.rad2deg(LL), 0, np.rad2deg(LL)], interpolation=
12 'nearest', origin='lower')
12 plt.colorbar();
13 def get_cll(cosmology = None, size = 64, pixels = 512):
14 """Obtem espectro de potencia para visualizar efeitos cosmológicos
15 do conteúdo de matéria do universo."""
16 LL = np.radians(size)
17 H0 = cosmology["H0"]
18 hh = H0/100
19 cambPar = camb.CAMBparams()
20 cambPar.set_cosmology(H0 = H0,
21 omch2 = cosmology["Om_M"]*pow(hh,2),
22 omk = cosmology["Om_K"],
23 ombh2 = cosmology["Om_Bh2"])
24 cambPar.set_for_lmax(1200, lens_potential_accuracy=0)
25 results = camb.get_results(cambPar)

```

```

25     powers = results.get_cmb_power_spectra(cambPar, spectra=['total'],
CMB_unit='muK')
26     Dll=powers['total']
27     ell = np.arange(Dll.shape[0])
28     factor = ell * (ell + 1.) / 2. / np.pi
29     Cl1 = Dll[2:,0] /factor[2:]
30     map_flat = nmt.synfast_flat(pixels, pixels, LL, LL, np.array([Cl1]),
[0])
31     return Dll[:,0], map_flat
32 float(LCDM_params.loc['omegam'].value)
33 param_planck = { "H0": H0,
34                 "Om_L" : float(LCDM_params.loc['omegal'].value),
35                 "Om_M" : float(LCDM_params.loc['omegam'].value),
36                 "Om_K" : float(LCDM_params.loc['omegak'].value),
37                 "Om_Bh2" : float(LCDM_params.loc['omegab2'].value)}
38 param_DE = { "H0": H0,
39             "Om_L" : float(LCDM_params.loc['omegal'].value)+0.3,
40             "Om_M" : float(LCDM_params.loc['omegam'].value)-0.3,
41             "Om_K" : float(LCDM_params.loc['omegak'].value),
42             "Om_Bh2" : float(LCDM_params.loc['omegab2'].value)}
43 param_MATTER = { "H0": H0,
44                 "Om_L" : float(LCDM_params.loc['omegal'].value)-0.6,
45                 "Om_M" : float(LCDM_params.loc['omegam'].value)+0.6,
46                 "Om_K" : float(LCDM_params.loc['omegak'].value),
47                 "Om_Bh2" : float(LCDM_params.loc['omegab2'].value)}
48 param_BARYON = { "H0": H0,
49                 "Om_L" : float(LCDM_params.loc['omegal'].value),
50                 "Om_M" : float(LCDM_params.loc['omegam'].value),
51                 "Om_K" : float(LCDM_params.loc['omegak'].value),
52                 "Om_Bh2" : float(LCDM_params.loc['omegab2'].value)*4}
53 nomes = ["planck2018",
54          r"$\Omega_\Lambda = \Omega_\Lambda^{\mathrm{planck}} + 0.3$",
55          r"$\Omega_M = \Omega_M^{\mathrm{planck}} + 0.6$",
56          r"$\Omega_{bh}^2 = 4; (\Omega_{bh}^2)^{\mathrm{planck}}$"]
57
58 cosmologias = [param_planck, param_DE, param_MATTER, param_BARYON]
59 universos = [get_cl1(cosmology=cosmologia) for cosmologia in cosmologias
]
60 fig, ax = plt.subplots(nrows = 2, ncols = 4, figsize = (16,9))
61 cmap=cm.get_cmap('viridis')
62 #normalizer=colors.Normalize(-500,500)
63 normalizer=colors.SymLogNorm(linthresh = 10, vmin = -500, vmax = 500,
base = 10)
64
65 im=cm.ScalarMappable(norm=normalizer, cmap = cmap)
66 for ii, universo in enumerate(universos):
67     Dll = universo[0]

```

```

68     ell = np.arange(0,D11.shape[0])
69     ax[0,ii].plot(ell[2:], D11[2:], label = nomes[ii])
70     ax[0,ii].grid()
71     ax[0,ii].legend(loc = "upper center")
72     ax[0,ii].set_ylim([0,10000])
73     ax[1,ii].imshow(universo[1][0], interpolation='nearest', origin='
lower',
74                     cmap = cmap, norm = normalizer)
75 fig.colorbar(im, ax=ax.ravel().tolist(), orientation="horizontal")
76 plt.savefig("fig09_george.png")
77 plt.show();

```

B.10 Códigos da Figura 10

```

1 samples1 = np.loadtxt("../data/raw/plikHM_TT/base_plikHM_TT_1.txt")
2 samples2 = np.loadtxt("../data/raw/plikHM_TT/base_plikHM_TT_2.txt")
3 samples3 = np.loadtxt("../data/raw/plikHM_TT/base_plikHM_TT_3.txt")
4 samples4 = np.loadtxt("../data/raw/plikHM_TT/base_plikHM_TT_4.txt")
5 column_names_raw = np.loadtxt("../data/raw/plikHM_TT/base_plikHM_TT.
paramnames", dtype=str, usecols=[0])
6 column_names = [x.replace("b'",',').replace("'",',') for x in
column_names_raw]
7 samples1 = pd.DataFrame(samples1[:,2:], columns=column_names)
8 samples2 = pd.DataFrame(samples2[:,2:], columns=column_names)
9 samples3 = pd.DataFrame(samples3[:,2:], columns=column_names)
10 samples4 = pd.DataFrame(samples4[:,2:], columns=column_names)
11 samples = pd.concat([samples1, samples2, samples3, samples4])
12
13 use_params = ["omegab2", "omegam*", "age*", "ns", "logA", "tau", "H0*"]
14 labels = [ r"\Omega_b h^2", r"\Omega_m", r"t_0", r"n_s", r"\log A",
r"\tau", r"H_0"]
15 df_planck = samples[use_params]
16 df_planck.columns = ["omegab2", "omegam", "age", "ns", "logA", "tau", "
H0"]
17 gsamples = getdist.MCSamples(samples=df_planck.to_numpy(), names=
use_params, labels=labels)
18
19 g = gdplt.get_subplot_plotter()
20 g.settings.figure_legend_frame = False
21 g.settings.alpha_filled_add=0.4
22 g.settings.title_limit_fontsize = 14
23 g.triangle_plot(gsamples, use_params,
24                 filled=True,
25                 legend_loc='upper right',
26                 title_limit=3
27                 )

```

```
28 plt.savefig("fig10_george.png")
29 plt.show();
```

B.11 Códigos da Figura 11

```
1
2 [language=Python]
3 samples = np.loadtxt("../data/raw/Pantheon+SH0ES_FlatLambdaCDM.txt")
4 params = ["omegam", "H0", "M", "prior", "post"]
5 use_params = ["omegam", "H0", "M"]
6 labels = [ r"\Omega_m", r"H_0", r"M_B"]
7 df = pd.DataFrame(samples)
8 df.columns = params
9 df_union = df[use_params]
10
11 gsamples = getdist.MCSamples(samples=df_union.to_numpy(), names=
    use_params, labels=labels)
12 g = gdplt.get_subplot_plotter()
13 g.settings.figure_legend_frame = False
14 g.settings.alpha_filled_add=0.4
15 g.settings.title_limit_fontsize = 14
16 g.triangle_plot(gsamples, use_params,
17     filled=True,
18     legend_loc='upper right',
19     title_limit=3
20     )
21 plt.savefig("fig11_george.png")
22 plt.show();
```

B.12 Códigos da Figura 12

```
1 import os
2 import sys
3 sys.path.append(os.path.abspath(os.path.join(os.path.split(os.getcwd())
    [0], "src")))
4 import matplotlib.pyplot as plt
5 from util import *
6
7 flat = np.linspace(50, 100, num=10000)
8
9 # create estimate
10 tdc = Estimate(name="HOLiCOW+STRIDES", mean=73.7, std=1.5)#from https://
    ui.adsabs.harvard.edu/abs/2019arXiv190704869W/abstract
11 planck = Estimate(name="Planck", mean=67.4, std=0.5)#from https://ui.
    adsabs.harvard.edu/abs/2018arXiv180706209P/abstract
```

```

12 des = Estimate(name="DES+BAO+BBN", mean=67.4, std=1.15)#from https://ui.
    adsabs.harvard.edu/abs/2018MNRAS.480.3879A/abstract
13 boss = Estimate(name="BOSS(Full-Shape)+BAO+BBN", mean=68.6, std=1.1)#
    update from https://arxiv.org/pdf/2002.04035.pdf
14
15
16 shoes = Estimate(name="SHOES", mean=73.0, std=1.0)# from https://ui.
    adsabs.harvard.edu/abs/2021arXiv211204510R/abstract
17 cchp = Estimate(name="CCHP", mean=69.6, std=1.9)# from "https://ui.
    adsabs.harvard.edu/abs/2019ApJ...882...34F/abstract"
18 miras = Estimate(name="MIRAS", mean=73.3, std=4.0)# from https://ui.
    adsabs.harvard.edu/abs/2020ApJ...889....5H/abstract
19 masers = Estimate(name="MCP", mean=73.9, std=3.0) #update from https://
    ui.adsabs.harvard.edu/abs/2020ApJ...891L...1P/abstract
20 sbf = Estimate(name="SBF", mean=73.3, std=2.5) # https://ui.adsabs.
    harvard.edu/abs/2021arXiv210102221B/abstract
21 bTF = Estimate(name="bTF", mean=75.1, std=2.7) #https://ui.adsabs.
    harvard.edu/abs/2020arXiv200608615S/abstract
22
23 # combine estimates
24 local_ceph = combine_estimates([tdc, shoes, masers], flat)
25 local_ceph.name = "with Cepheids"
26 local_trgb = combine_estimates([tdc, cchp, masers], flat)
27 local_trgb.name = "with TRGB"
28 local_miras = combine_estimates([tdc, miras, masers], flat)
29 local_miras.name = "with MIRAS"
30 local_all = combine_estimates([tdc, shoes, cchp, miras, masers, sbf, bTF
    ], flat)
31 local_all.name = "combining all"
32
33
34 # compute the tension wrt Planck
35 tension_local_ceph = compute_tension(planck, local_ceph)
36 tension_local_trgb = compute_tension(planck, local_trgb)
37 tension_local_miras = compute_tension(planck, local_miras)
38 tension_local_all = compute_tension(planck, local_all)
39
40 # Special fine-tuned design, adapted for Verde, Treu and Riess 2019
41
42 plt.figure(figsize=(8.0, 9.4), dpi=200)
43 plt.subplots_adjust(left=0.01, right=0.99, bottom=0.20, top=0.92)
44
45 ax = plt.subplot(1, 1, 1)
46 plt.xlabel(r"$H_0 \text{ [km, s}^{-1}\text{, Mpc}^{-1}]$", fontsize=24)
47 plt.yticks(fontsize=14)
48 plt.xticks(fontsize=20)
49

```

```

50 xs = [planck, des, boss, shoes, cchp, miras, tdc, masers, sbf, bTF,
        local_all, local_ceph, local_trgb, local_miras]
51 ys = [17.3, 16.0, 14.8, 13.3, 12.0, 10.7, 9.4, 8.1, 6.8, 5.5, 4, 3, 2,
        1]
52
53 colors = ['peru', 'grey', 'mediumturquoise', 'royalblue', 'crimson', '
           purple', 'limegreen',
54           'goldenrod', 'teal', 'darkorange', 'black', 'cornflowerblue',
           'palevioletred', 'plum']
55
56 plt.axis([64.5, 80.9, 0.5, 18.0]) # 75.6
57 plt.gca().yaxis.set_major_locator(plt.NullLocator())
58 plt.title(r'$\rm{flat-\Lambda CDM}$', fontsize=24)
59
60 for x, y, c in zip(xs, ys, colors):
61     pcs = x.getpcs()
62     plt.scatter(pcs[1], y, s=80, edgecolor=c, c=c, label=x.name)
63     plt.plot([pcs[0], pcs[2]], [y, y], linewidth=2.5, color=c)
64
65     if x.name not in ['combining all', 'with Cepheids', 'with TRGB', '
with MIRAS']:
66         delaytext = r"$%.1f^{+%.1f}_{-%.1f}$" % (pcs[1], pcs[2]-pcs[1],
        pcs[1]-pcs[0])
67         plt.annotate(delaytext, xy=(pcs[1], y + 0.18), color = c,
        horizontalalignment="center", fontsize=17)
68         plt.annotate(x.name, xy=(pcs[1], y-0.4), color = "dingrey",
        horizontalalignment="center", fontsize=10)
69
70     else:
71         delaytext = r"$%.1f^{+%.1f}_{-%.1f}$" % (pcs[1], pcs[2]-pcs[1],
        pcs[1]-pcs[0])
72         plt.annotate(delaytext, xy=(pcs[1], y + 0.19), color = c,
        horizontalalignment="center", fontsize=14)
73         plt.annotate(x.name, xy=(pcs[1]+ 2.2, y + 0.19), color = "
        dingrey", horizontalalignment="center", fontsize=10)
74
75 # dashed lines to split early, late and combinations
76 plt.axhline(14.2, color="darkgrey", ls='--', alpha=0.5)
77 plt.axhline(4.9, color="darkgrey", ls='--', alpha=0.5)
78
79 # subpanels labels
80 bbox_props = dict(boxstyle="square", lw=1, fc="white")
81 plt.annotate("Early", xy=(66.07, 17.2), color = "black",
82             horizontalalignment="right", verticalalignment="center",
            fontsize=17, bbox=bbox_props)
83 plt.annotate("Late", xy=(80.55, 13.6), color = "black",
84             horizontalalignment="right", verticalalignment="center",

```

```
    fontsize=17, bbox=bbox_props)
85
86
87 # printing the tension with Planck
88 xpos = 79.4
89 plt.annotate("Planck vs. Late", xy=(xpos, 4.6), color = "dimgrey",
90             horizontalalignment="center", fontsize=11)
91
92 plt.annotate(r"$\rm{%.1f}\sigma$" % tension_local_all, xy=(xpos, 4.18),
93             color = "black",
94             horizontalalignment="center", fontsize=14)
95 plt.annotate(r"$\rm{%.1f}\sigma$" % tension_local_ceph, xy=(xpos, 3.18),
96             color = "black",
97             horizontalalignment="center", fontsize=14)
98 plt.annotate(r"$\rm{%.1f}\sigma$" % tension_local_trgb, xy=(xpos, 2.18),
99             color = "black",
100            horizontalalignment="center", fontsize=14)
101 plt.annotate(r"$\rm{%.1f}\sigma$" % tension_local_miras, xy=(xpos, 1.18)
102            , color = "black",
103            horizontalalignment="center", fontsize=14)
104 plt.tight_layout()
105 #plt.savefig('H0_tension.pdf', dpi=200)
106 #plt.savefig('H0_tension.png', dpi=200)
107 plt.savefig("fig12_george.png")
```