



UNIVERSIDADE  
ESTADUAL DE LONDRINA

---

RENAN ALVES DE OLIVEIRA

**TESTES DE ISOTROPIA UTILIZANDO  
VETORES DE MULTIPOLO EM PEQUENAS ESCALAS**

---

Londrina  
2018

RENAN ALVES DE OLIVEIRA

**TESTES DE ISOTROPIA UTILIZANDO  
VETORES DE MULTIPOLLO EM PEQUENAS ESCALAS**

Dissertação de mestrado apresentada ao Departamento de Física da Universidade Estadual de Londrina, como requisito parcial à obtenção do título de Mestre.

Orientador: Prof. Dr. Thiago dos Santos Pereira

Londrina

2018

Alves de Oliveira, Renan Alves de Oliveira.

Testes de Isotropia Utilizando Vetores de Multipolo em Pequenas Escalas / Renan Alves de Oliveira. - Londrina, 2018.

78 f. : il.

Orientador: Thiago do Santos Pereira.

Dissertação (Mestrado em Física) - Universidade Estadual de Londrina, Centro de Ciências Exatas, Programa de Pós-Graduação em Física, 2018.

Inclui bibliografia.

1. Radiação Cósmica de Fundo - Tese. 2. Vetores de Multipolo - Tese. 3. Isotropia - Tese. 4. Princípio Cosmológico - Tese. I. dos Santos Pereira, Thiago dos Santos. II. Universidade Estadual de Londrina. Centro de Ciências Exatas. Programa de Pós-Graduação em Física. III. Título.

RENAN ALVES DE OLIVEIRA

**TESTES DE ISOTROPIA UTILIZANDO  
VETORES DE MULTIPOLO EM PEQUENAS ESCALAS**

Dissertação de mestrado apresentada ao Departamento de Física da Universidade Estadual de Londrina, como requisito parcial à obtenção do título de Mestre.

**BANCA EXAMINADORA**

---

Orientador: Prof. Dr. Thiago dos Santos Pereira  
Universidade Estadual de Londrina - UEL

---

Dr<sup>a</sup>. Camila Paiva Novaes  
Observatório Nacional - ON

---

Prof<sup>a</sup>. Dr<sup>a</sup>. Paula Fernanda Bienzobas  
Universidade Estadual de Londrina - UEL

Londrina, 14 de Setembro de 2018.

*Dedico este trabalho a todos aqueles que estão dispostos a não terem medo do desconhecido.*

# Agradecimentos

À minha mãe e à minha irmã que sempre me apoiaram em todos os momentos durante o meu mestrado, inclusive nas madrugadas em que fiquei trabalhando na programação dos códigos e realizando simulações.

Aos meus amigos da graduação e da pós-graduação, em especial aos meus colegas da Sala 25, onde compartilhamos diversas ideias não só relacionadas à física, mas também sobre nós mesmos, sendo estas responsáveis para o amadurecimento pessoal.

Aos meus amigos fora da universidade, sejam eles do LoL, quanto aos amigos de Maringá. Em especial, ao meu companheiro que sempre me apoiou em momentos difíceis e que me fez acreditar em mim mesmo, não só aprendendo com os erros, mas também relembrar dos acertos.

Aos meus professores da pós-graduação, sendo estes responsáveis não só pela minha formação profissional, mas também no crescimento pessoal. Agradeço também à professora vizinha de sala pelas inúmeras conversas que tivemos.

Agradeço profundamente ao meu orientador por ter acreditado em mim durante esses sete anos de trabalho (cinco de graduação e dois de mestrado). Por ter me ajudado a explorar não só o universo físico, mas também o computacional, onde certamente abrirá novas perspectivas e oportunidades para mim no futuro. Obrigado por tudo! Além disso, gostaria de agradecer imensamente ao apoio que tive durante todo o meu projeto de pesquisa desenvolvido em parceria com a Universidade Federal do Rio de Janeiro.

Ao Programa de Pós-Graduação em Física por me dar oportunidade dar continuidade aos meus estudos, à CAPES pelo auxílio financeiro e à UEL como sendo um anexo da minha casa por todos esses anos.

*“A imaginação muitas vezes nos leva a mundos que nunca sequer existiram. Mas sem ela, não vamos a lugar nenhum.”*

*Carl Sagan.*

ALVES DE OLIVEIRA, Renan. **Testes de isotropia utilizando vetores de multipolo em pequenas escalas**. 2018. 78 p. Dissertação de Mestrado (Mestrado em Física) – Universidade Estadual de Londrina, Londrina, 2018.

## Resumo

A detecção das flutuações de temperatura da Radiação Cósmica de Fundo só foi possível devido ao lançamento sucessivo de sondas espaciais equipadas com instrumentos mais sensíveis. A partir dessas medidas, diversas concepções cosmológicas puderam ser testadas. Uma delas é o Princípio Cosmológico onde afirma que o universo é espacialmente isotrópico em escalas cosmológicas. O objetivo deste trabalho foi testar a isotropia do universo usando os dados obtidos pelo Satélite Planck das flutuações de temperatura, sendo estes convertidos em vetores de multipolo. As vantagens em se utilizar esse formalismo são à independência desses vetores a qualquer modelo físico e a um sistema de coordenadas. O teste de isotropia utilizado consistiu em simular mapas cosmológicos mascarados e com ruído instrumental isotrópico com a finalidade de se obter um perfil da distribuição dos vetores de multipolo até  $\ell = 1500$ . Para uma comparação dos dados observados com os simulados, foram realizados testes de  $\chi^2$ , obtendo-se valores- $p$  em várias escalas de interesse cosmológico.

**Palavras-chave:** Radiação Cósmica de Fundo. Vetores de Multipolo. Isotropia. Princípio Cosmológico.

ALVES DE OLIVEIRA, Renan. **Isotropy tests using multipole vectors in small scales.** 2018. 78 p. Master Dissertation (Master in Physics) – Universidade Estadual de Londrina, Londrina, 2018.

## Abstract

The detection of temperature fluctuations of the Cosmic Background Radiation was only possible due to the successive launching of space probes equipped with more sensitive instruments. From these measurements, several cosmological conceptions could be tested. One is the Cosmological Principle which states that the universe is spatially isotropic on cosmological scales. The objective of this work was to test the isotropy of the universe using the data obtained by the Planck Satellite from temperature fluctuations, these being converted into multipole vectors. The advantages of using this formalism are the independence of these vectors to any physical model and to a coordinate system. The isotropic test consisted of simulating masked cosmological maps and instrumental isotropic noise in order to obtain a distribution profile of the multipole vectors up to  $\ell = 1500$ . For a comparison of the observed data with the simulated,  $\chi^2$  tests were performed, obtaining  $p$ -values at various scales of cosmological interest.

**Keywords:** Cosmic Microwave Background. Multipole Vectors. Isotropy. Cosmological Principle.

# Lista de ilustrações

Figura 1	– Espectro angular de potências obtidos pelo satélite Planck publicado em 2018. Estão representados no eixo horizontal os multipolos $\ell$ , enquanto que no eixo vertical a amplitude de $\mathcal{D}_\ell \equiv \mathcal{C}_\ell^{\ell(\ell+1)}/4\pi$ . Os pontos vermelhos com suas respectivas barras de erro são os dados observados e a linha contínua azul é o espectro angular de potências teórico obtido através do melhor ajuste do modelo $\Lambda$ CDM. . . . .	29
Figura 2	– Comparação entre as resoluções angulares das observações realizadas pelos satélites COBE, WMAP e Planck. As regiões alaranjadas e azuis representam as flutuações positivas e negativas de temperatura da RCF com amplitude da ordem de $10^{-5}$ K . . . . .	30
Figura 3	– Mapas com diferentes pixelizações gerados no HEALPix. Cada pixel apresenta uma determinada amplitude e a projeção mais comum utilizada é a de Mollweide. As coordenadas dos centros dos mapas são em $\theta = \pi/2$ e $\phi = 0$ ( $\phi$ cresce do centro para à esquerda). . . . .	30
Figura 4	– Efeitos astrofísicos observados pelo satélite que são removidos dos mapas obtidos. . . . .	32
Figura 5	– Máscara UT78. Essa máscara possui uma região de corte maior no centro, onde há grande contribuição do centro da Via Láctea. A fração do céu não mascarada é da ordem de 78%. . . . .	32
Figura 6	– Comparação de intensidades dos espectros de potências do ruído instrumental de cada <i>pipeline</i> do Planck. A linha preta representa o melhor ajuste do modelo $\Lambda$ CDM aos dados da RCF. As linhas acima à linha preta representam o melhor ajuste mais o ruído instrumental de cada <i>pipeline</i> . . . . .	37
Figura 7	– Diferença entre mapas com a aplicação de máscara sem e com apodizagem. . . . .	38
Figura 8	– Tempo gasto para obter vetores de multipolo usando diferentes metodologias em um computador com processador Intel(R) Core(TM) i5-600CPU @ 2.70 GHz. Os triângulos verdes, quadrados azuis e pontos vermelhos são referentes à [1, 2], [3] e [4] respectivamente. Acima de $\ell \gtrsim 900$ , o tempo gasto pelo MPSolve aumenta devido ao mal condicionamento dos polinômios já que o software aumenta a precisão para contornar o problema de condicionamento. . . . .	41
Figura 9	– Distribuição dos vetores de multipolo localizados no hemisfério norte para o multipolo $\ell = 10$ provenientes de 5000 MIG. A linha tracejada preta representa (4.5). A forma da distribuição é simétrica ao incluir as antípodas. . . . .	42

Figura 10 – Máscara sintética com uma faixa equatorial de $30^\circ$ de largura com simetria azimutal. . . . .	42
Figura 11 – Efeito da máscara equatorial para a distribuição de $5 \times 10^5$ vetores de multipolo situados no hemisfério norte. Os dados foram separados uniformemente, sendo 80 <i>bins</i> primeira linha e 10 <i>bins</i> na segunda linha. 43	43
Figura 12 – Projeção ortográfica dos vetores de multipolo de $\ell = 100$ obtidos de 5000 MIG mascarados com a máscara da Figura 10. Os $5 \times 10^7$ vetores de multipolo são representados por pontos na esfera, dando a impressão de um contínuo. . . . .	44
Figura 13 – Fluxograma de obtenção do teste de $\chi^2$ de normalidade para cada multipolo até $\ell_{máx} = 1500$ . Como os vetores de multipolo possuem dois graus de liberdade, $\eta$ e $\varphi$ , têm-se duas colunas de resultados sendo elas geradas à partir das distribuições dessas coordenadas. . . . .	45
Figura 14 – Resultado do teste de $\chi^2$ de normalidade para $2 \leq \ell \leq 1500$ para as simulações de controle e todas as <i>pipelines</i> do satélite Planck. As curvas e a regiões verdes representam respectivamente a média e a variância para cada multipolo, provenientes dos dados simulados. Os pontos vermelhos e azuis são os $\chi^2_\ell$ para cada <i>pipeline</i> . Já os marcadores em cruz são os resultados do teste de $\chi^2$ para vetores de multipolo com uma distribuição diferente da dos dados simulados. . . . .	46

# Lista de tabelas

Tabela 1 – Resolução angular para diversos valores de NSIDE. A resolução angular compreende o comprimento de arco na esfera de raio unitário. . . . .	31
Tabela 2 – $\chi^2 [\chi_\ell^2]$ sendo uma combinação dos pontos vermelhos e azuis em comparação com as curvas verdes da Figura 14 e seus valores $-p$ associados para diversos intervalos de interesse. . . . .	47
Tabela 3 – Mesma análise da Tabela 2, porém com dados sem a máscara UT78, representados pelos marcadores roxos em formato de cruz. . . . .	47



# Lista de abreviaturas e siglas

RCF	Radiação C3smica de Fundo
PC	Princ3pio Cosmol3gico
COBE	<i>Cosmic Microwave Background Explorer</i>
WMAP	<i>Wilkinson Microwave Anisotropy Probe</i>
MIG	Mapas Isotr3picos e Gaussianos



# Sumário

<b>1</b>	<b>INTRODUÇÃO</b> . . . . .	<b>25</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b> . . . . .	<b>27</b>
2.1	A Radiação Cós mica de Fundo . . . . .	27
2.2	A Missão Planck . . . . .	29
2.2.1	HEALPix . . . . .	30
2.2.2	Componentes Astrofísicos e Máscaras . . . . .	31
<b>3</b>	<b>VETORES DE MULTIPOLO</b> . . . . .	<b>33</b>
<b>4</b>	<b>MATERIAIS E MÉTODOS</b> . . . . .	<b>37</b>
4.1	Simulação de Mapas . . . . .	37
4.2	Extração dos Vetores de Multipolo . . . . .	38
4.3	Distribuição dos Vetores de Multipolo . . . . .	39
4.3.1	Testes Estatísticos . . . . .	39
<b>5</b>	<b>RESULTADOS E DISCUSSÃO</b> . . . . .	<b>41</b>
5.1	Comparação entre Métodos de Extração dos Vetores de Multipolo . . . . .	41
5.2	Distribuição dos Vetores de Multipolo . . . . .	42
5.2.1	Aplicação de máscaras . . . . .	43
5.3	Teste de $\chi^2$ de Normalidade para o Planck . . . . .	44
<b>6</b>	<b>CONCLUSÃO</b> . . . . .	<b>49</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>51</b>
	<b>APÊNDICES</b> . . . . .	<b>55</b>
	<b>APÊNDICE A – VETORES DE MULTIPOLO E TESTE DE <math>\chi^2</math></b> . . . . .	<b>57</b>
A.1	Independência dos $\mathcal{C}_\ell$ nos Vetores de Multipolo . . . . .	57
A.2	Teste de $\chi^2$ de Normalidade . . . . .	57
A.3	Valores- $p$ . . . . .	59
	<b>APÊNDICE B – CÓDIGOS</b> . . . . .	<b>61</b>



# 1 Introdução

Ao longo da história da humanidade, muitas questões relacionadas ao universo foram formuladas sendo algumas respondidas e outras permanecendo ainda em aberto. Uma dessas questões diz respeito ao centro do universo, sendo que por muito tempo acreditou-se que a Terra estava nele. O modelo de universo geocêntrico foi predominante desde os gregos até o século 16 A.C.. Entretanto, existiram astrônomos como Aristarco de Samos (230 A.C.) questionando se a Terra realmente estava no centro do universo. Com o advento do renascimento, o surgimento do cálculo e de novas ferramentas observacionais como o telescópio, a ideia geocêntrica foi substituída pelo heliocentrismo, sendo o Sol o centro do universo. Devido à mudança de pensamento em relação ao centro do universo, surge o Princípio Copernicano<sup>1</sup> afirmando a não existência de um centro no universo.

Por conta dos avanços da astronomia no fim do século XIX, com a descoberta de outros objetos fora do Sistema Solar, tais como galáxias e seus aglomerados, a formulação da Teoria de Relatividade Geral de Albert Einstein em 1915 e, posteriormente, a descoberta da expansão do universo através das observações de *redshift* de galáxias em 1929 por Edwin Powell Hubble, deu-se origem à cosmologia moderna. A cosmologia ganhou de fato *status* de ciência devido à possibilidade de desenvolver diversos modelos matemáticos sendo esses confrontados com as observações.

O modelo cosmológico mais aceito hoje leva em consideração a constante cosmológica  $\Lambda$  e a matéria escura fria, ou, *Cold Dark Matter (CDM)* [5]. Neste modelo, o universo é dominado por 75% de energia escura sendo esta responsável pela sua expansão. Os outros 25% é composto pela matéria escura, que seria responsável por explicar diversos efeitos astrofísicos como a curva de rotação de galáxias. O termo "escura" está relacionado à nossa ignorância em respeito à forma e às características dessas quantidades. Mesmo considerando esses ingredientes misteriosos, há quase 20 anos foi observado que o universo está em expansão acelerada, e é o modelo  $\Lambda$ CDM que ganha maior destaque ao ser convalidado com as observações astrofísicas de *remdshifts* de supernovas do tipo 1a [6].

Um grande marco para a cosmologia moderna foi a detecção da Radiação Cósmica de Fundo (RCF) em micro-ondas, sendo esta importantíssima tanto para a determinação de parâmetros cosmológicos [7], quanto para a caracterização de aspectos globais do universo. Uma generalização do Princípio Copernicano é o Princípio Cosmológico (PC) [8]. O PC afirma que em escalas cosmológicas ( $\gtrsim 100 \text{ Mpc}$ )<sup>2</sup> o universo é espacialmente homogêneo e isotrópico. Homogêneo pois não existe nenhuma região privilegiada, sendo que um observador obterá as mesmas características sobre o universo sendo este situado na Terra ou nos confins do universo. Isotrópico pois o universo é o mesmo em todas as direções,

<sup>1</sup> Em homenagem à Nicolau Copérnico devido as suas investigações em relação ao modelo heliocêntrico.

<sup>2</sup>  $1 \text{ Mpc} \sim 3 \times 10^{22} \text{ m}$ .

não havendo nenhuma direção privilegiada. Uma das formas de testar o PC por exemplo é analisar as propriedades da RCF ou determinar a distribuição espacial de galáxias, como por exemplo realizada pelo *Sloan Digital Sky Survey* [9] e o *2dF Galaxy Redshift Survey* [10].

Estamos interessados neste trabalho em utilizar os dados da RCF para testar a hipótese de isotropia do PC com o auxílio de simulações e testes estatísticos usando uma ferramenta matemática conhecida na literatura como vetores de multipolos. Realizamos uma brevíssima introdução sobre essa radiação no Capítulo 2, apresentando alguns conceitos matemáticos e o tratamento numérico desses dados. O leitor interessado em um tratamento completo poderá consultar a referência [11]. No Capítulo 3, apresentamos uma pequena revisão sobre uma nova forma de análise para testar o PC utilizando os vetores de multipolo, sendo exposto alguns problemas computacionais encontrados e também soluções para contornar esses problemas. Adiante, no Capítulo 4, apresentamos uma metodologia sobre o procedimento de análise. Os resultados deste trabalho estão exibidos e discutidos no Capítulo 5, sendo o trabalho concluído no Capítulo 6.

## 2 Fundamentação Teórica

### 2.1 A Radiação Cósmica de Fundo

Interessados no aprimoramento de um equipamento ultrasensível na faixa das micro-ondas, os físicos americanos Arno Allan Penzias e Robert Woodrow Wilson verificaram a existência de um ruído captado em todas as direções do céu em que o radiotelescópio era apontado. A origem deste ruído foi comprovada mais tarde em 1964 como sendo um dos resquícios do universo primordial, sendo esta radiação eletromagnética chamada de RCF [12].

Os primeiros elementos químicos foram formados aproximadamente nos três primeiros minutos após o Big Bang [13], sendo o universo constituído por um plasma composto por várias partículas como os fótons, os elétrons e os prótons. Deste período até o surgimento dos primeiros átomos de hidrogênio estáveis, fótons interagiram constantemente com os elétrons via espalhamento Compton. Existia também uma competição entre o potencial gravitacional da matéria e a pressão da radiação dos fótons. Essa competição deu origem as regiões de sub e sobredensidade, o que equivale hoje às regiões quentes e frias do campo de temperatura da RCF. Com a expansão do universo primordial, o livre caminho médio de interação dos fótons foi aumentando, fazendo com que esses fótons passassem a circular livremente por todas as regiões do universo desde aproximadamente 300 mil anos após o *Big Bang*, formando assim a RCF.

Desde a descoberta desta radiação, os astrônomos determinaram diversas características do universo além do espectro visível, explorando-o em micro-ondas. O espectro da RCF foi medido posteriormente com maior precisão pelo satélite *Cosmic Microwave Background Explorer* (COBE) [14] em 1989, apresentando o comportamento de uma radiação de corpo negro de intensidade

$$I(\nu) = \frac{2h\nu^3}{c^2} e^{1-\frac{h\nu}{kT}}, \quad (2.1)$$

sendo  $\nu$  a frequência eletromagnética,  $h = 6,63 \times 10^{-34} \text{ Js}$  a constante de Planck,  $c = 299792458 \text{ m s}^{-1}$  a velocidade da luz,  $k = 1,38 \times 10^{-23} \text{ JK}^{-1}$  a constante de Boltzmann. A temperatura média da RCF foi medida sendo aproximadamente  $T_0 \sim 2,73 \text{ K}$ . Posteriormente, desvios desta temperatura média foram observados pela sonda *Wilkinson Microwave Anisotropy Probe* (WMAP) [15] sendo eles da ordem de  $\delta T/T_0 \equiv (T - T_0)/T_0 \sim 10^{-5} \text{ K}$ .

Uma vez que os fótons da RCF percorrem aproximadamente a mesma distância até chegar aos instrumentos de medida, as flutuações de temperatura se comportam como um função na esfera, e pode portanto ser expandida em harmônicos esféricos como:

$$\frac{\delta T}{T_0}(\theta, \phi) = \sum_{\ell=1}^{\infty} f_{\ell} = \sum_{\ell=1}^{\infty} \sum_{m=-\ell}^{\ell} a_{\ell m} Y_{\ell m}(\theta, \phi), \quad (2.2)$$

sendo  $f_\ell$  uma função multipolar descrevendo a contribuição de cada multipolo  $\ell$ . Essa decomposição assume implicitamente um sistema de coordenadas para determinar as funções de base  $Y_{\ell m}$ . No caso da eq. (2.2), esse é o sistema de coordenadas esféricas usual, onde  $\hat{\mathbf{r}} = (\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta)$ . Para cada multipolo existe uma escala angular correspondente dada por

$$\alpha_\ell \equiv \frac{\pi}{\ell}. \quad (2.3)$$

Finalmente, os coeficientes  $a_{\ell m}$  da expansão são chamados de momentos de multipolo, sendo eles coeficientes complexos. Os momentos de multipolo são interpretados como sendo realizações de um *ensemble* resultante de flutuações aleatórias de temperatura gerados por um campo escalar real aleatório. Como (2.2) é uma função real, segue que

$$a_{\ell m}^* = (-1)^m a_{\ell, -m}, \quad (2.4)$$

onde usamos  $Y_{\ell m}^*(\theta, \phi) = (-1)^m Y_{\ell, -m}$ . Utilizando a ortonormalidade da base de harmônicos esféricos, os momentos de multipolo são dados por

$$a_{\ell m} = \int d^3\hat{\mathbf{r}} \frac{\delta T}{T_0}(\theta, \phi) Y_{\ell m}^*(\theta, \phi). \quad (2.5)$$

A quantidade  $\delta T/T_0$  pode ser calculada teoricamente dado um modelo cosmológico, porém este resultado não será demonstrado neste trabalho. Para mais detalhes sobre os processos físicos relacionados a formação das anisotropias da RCF, veja o Capítulo 8 de [5]. Devido ao PC, as flutuações de temperatura devem ser homogêneas e isotrópicas. Uma forma de testar o PC é usando a correlação entre as flutuações de temperatura em dois pontos diferentes, sendo que esta função de correlação deve ser invariante por translação espacial e pela troca de coordenadas  $\hat{\mathbf{r}}_1$  e  $\hat{\mathbf{r}}_2$ . Para que o universo seja isotrópico, esta função deve depender apenas dos ângulos entre  $\hat{\mathbf{r}}_1$  e  $\hat{\mathbf{r}}_2$  [16]. Os mesmos argumentos são válidos para a função de correlação dos momentos de multipolo. Como consequência direta das hipóteses de homogeneidade e isotropia, segue que [16]

$$\langle a_{\ell m} a_{\ell' m'}^* \rangle = \mathcal{C}_\ell \delta_{\ell\ell'} \delta_{mm'}, \quad (2.6)$$

sendo  $\langle \rangle$  o valor esperado em vários ensembles, e  $\mathcal{C}_\ell$  é chamado de espectro angular de potências. Esse espectro descreve a variância dos multipolos para determinada escala cosmológica. Esta quantidade pode ser calculada teoricamente e também quantificada através das observações conforme a Figura 1. Entretanto, como temos somente um universo, é conveniente definir um estimador teórico para este espectro sendo definido como

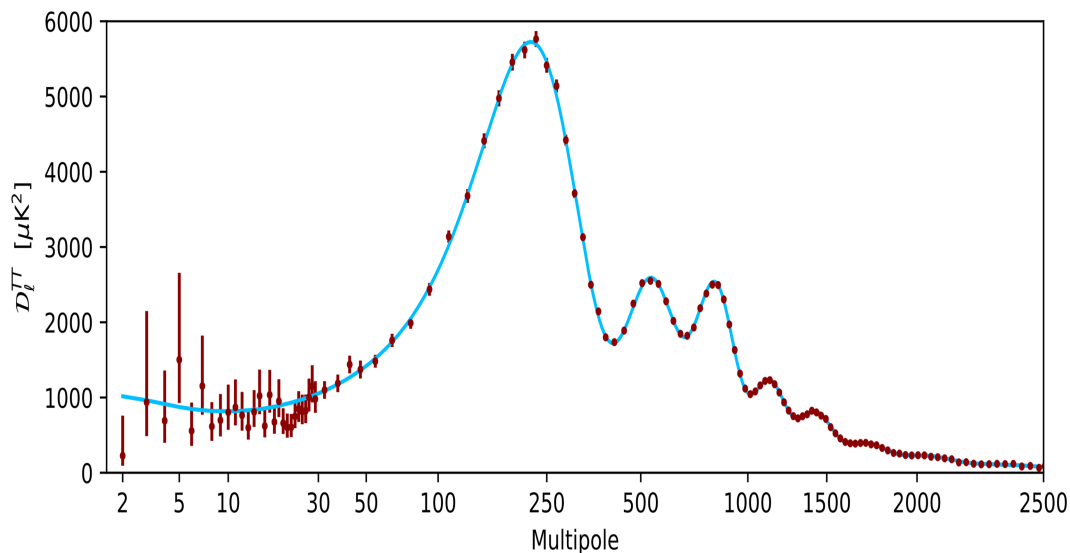
$$\hat{\mathcal{C}}_\ell \equiv \frac{1}{2\ell + 1} \sum_{m=-\ell}^{\ell} |a_{\ell m}|^2, \quad (2.7)$$

tal que a média deste estimador é

$$\langle \hat{\mathcal{C}}_\ell \rangle = \frac{1}{2\ell + 1} \sum_{m=-\ell}^{\ell} \langle |a_{\ell m}|^2 \rangle = \mathcal{C}_\ell, \quad (2.8)$$

onde na última igualdade foi utilizada a Eq. (2.6).

Figura 1 – Espectro angular de potências obtidos pelo satélite Planck publicado em 2018. Estão representados no eixo horizontal os multipolos  $\ell$ , enquanto que no eixo vertical a amplitude de  $\mathcal{D}_\ell \equiv \mathcal{C}_\ell^{\ell(\ell+1)}/4\pi$ . Os pontos vermelhos com suas respectivas barras de erro são os dados observados e a linha contínua azul é o espectro angular de potências teórico obtido através do melhor ajuste do modelo  $\Lambda$ CDM.



**Fonte:** Colaboração Planck/ESA.

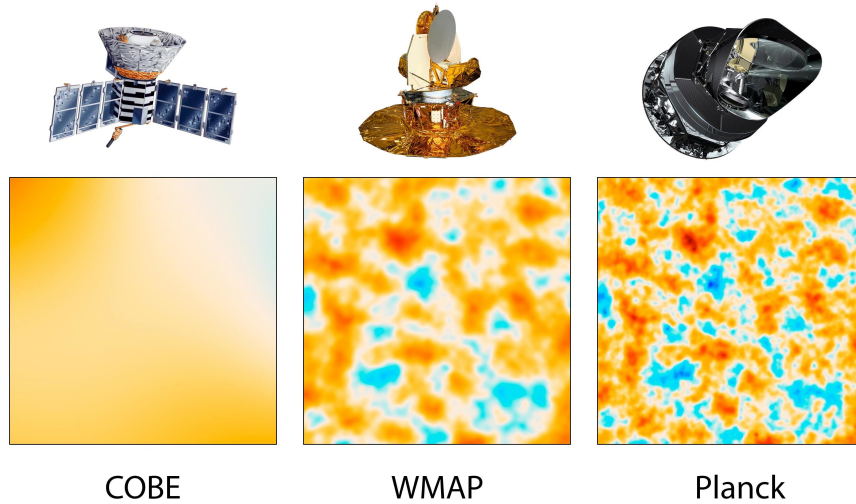
## 2.2 A Missão Planck

Lançado em 14 de Maio de 2009, o satélite Planck foi responsável por obter as informações mais precisas que caracterizam a RCF, conforme a Figura 2. Seus objetivos foram desde medidas em alta resolução da intensidade e polarização da RCF, efeitos de lentes gravitacionais até a criação de um catálogo de galáxias [17]. A missão teve término no dia 23 de Outubro de 2013 e foi financiada pela Agência Espacial Europeia (ESA) em parceria com a Agência Espacial Norte Americana (NASA). Para uma leitura mais profunda a respeito da missão com detalhes desde a criação e a capacidade de precisão do satélite, a missão Planck disponibilizou um livro [18] contendo todas essas informações.

O satélite foi equipado com instrumentos capazes de medir a temperatura do universo em diversas frequências do espectro eletromagnético, sendo um interferômetro sensível a baixas frequências e outro em altas frequências, ambos abrangendo um intervalo entre 30 à 857 GHz. Além disso, o satélite incluiu um sistema ativo e passivo de resfriamento para até 0,1 °C acima do zero absoluto. Antes de entrarmos nos detalhes da obtenção dos dados disponíveis ao público através do *Planck Legacy Archive*<sup>1</sup>, efetuaremos uma digressão a respeito do tratamento de dados utilizando pixels com o software HEALPix.

<sup>1</sup> <http://pla.esac.esa.int/pla/>.

Figura 2 – Comparação entre as resoluções angulares das observações realizadas pelos satélites COBE, WMAP e Planck. As regiões alaranjadas e azuis representam as flutuações positivas e negativas de temperatura da RCF com amplitude da ordem de  $10^{-5}$  K .

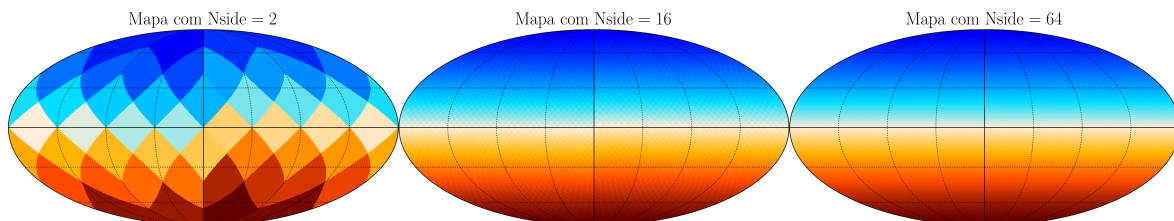


Fonte: NASA/JPL-Caltech/ESA.

### 2.2.1 HEALPix

O *Hierarchical Equal Area Iso Latitude Pixelation* (HEALPix) é um programa capaz de discretizar funções na esfera e convertê-las de forma eficiente em pixels de áreas iguais na esfera [19, 20]. A manipulação de cada pixel é facilitada devido a sua forma de indexação, sendo possível a conversão de suas posições em coordenadas esféricas ou cartesianas. Além disso, o software é capaz de simular mapas<sup>2</sup> cosmológicos da RCF como, por exemplo, mostrado na Figura 3. Um dos objetivos e aplicações deste pacote foi a análise de dados provenientes de satélites precedentes ao Planck, como o COBE e o WMAP. O satélite Planck realizou diversas medidas ao longo dos seus quatro anos de funcionamento, obtendo mapas das flutuações de temperatura em diversas resoluções angulares.

Figura 3 – Mapas com diferentes pixelizações gerados no HEALPix. Cada pixel apresenta uma determinada amplitude e a projeção mais comum utilizada é a de Mollweide. As coordenadas dos centros dos mapas são em  $\theta = \pi/2$  e  $\phi = 0$  ( $\phi$  cresce do centro para à esquerda).



Fonte: O próprio autor.

<sup>2</sup> Neste trabalho, os mapas estão associados ao campo bidimensional das flutuações de temperatura da RCF.

Tabela 1 – Resolução angular para diversos valores de NSIDE. A resolução angular compreende o comprimento de arco na esfera de raio unitário.

NSIDE	NPIX	$\Theta$
1	12	58,6°
2	48	29,3°
4	192	14,7°
8	768	7,33°
16	3072	3,66°
32	12288	1,83°
64	49152	55,0'
128	196608	27,5'
256	786432	13,7'
512	3145728	6,87'
1024	12582912	3,44'

**Fonte:** Fragmento da Tabela 1 de [20].

No esquema de pixelização do HEALPix, um mapa possui uma resolução angular dada pelo parâmetro NSIDE apresentados na Tabela 1, sendo a resolução angular  $\Theta$  para diversos valores de NSIDE e número de pixels NPIX correspondentes.

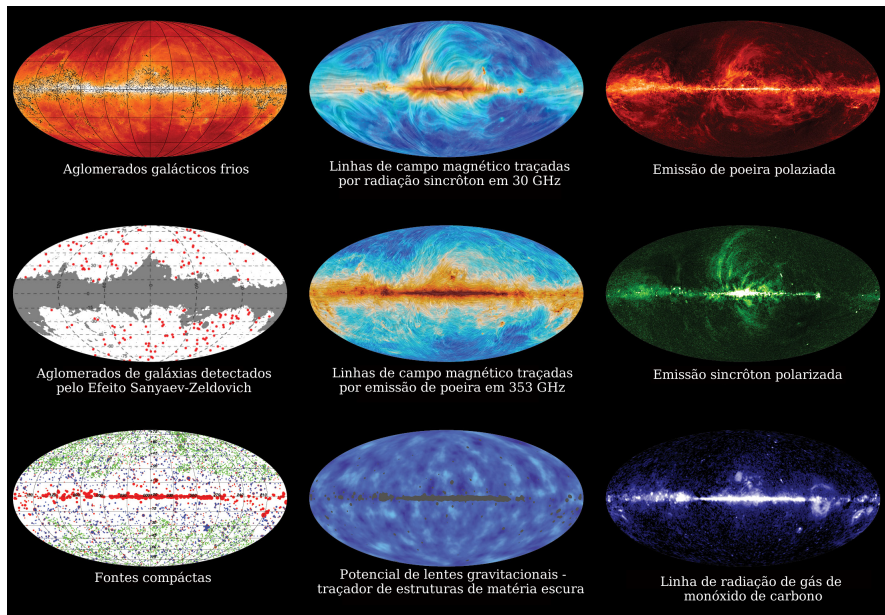
### 2.2.2 Componentes Astrofísicos e Máscaras

As observações cosmológicas estão sujeitas a contaminações de fontes astrofísicas provenientes do Sistema Solar, da Via Láctea e de outras regiões do universo em diversos comprimentos de onda. Para que apenas sinais cosmológicos sejam analisados, é necessário realizar a remoção de diversos componentes astrofísicos.

Para a remoção dos sinais apresentados na Figura 4, o Planck conta com quatro “*pipelines*” denominadas: **Commander**, **NILC**, **SEVEM** e **SMICA**. Cada *pipeline* utiliza diferentes métodos [21] de compressão dos mapas em várias frequências obtidos da missão, atribuindo a cada um desses mapas pesos diferentes, totalizando quatro mapas finais de intensidade da temperatura da RCF. Com a separação de componentes astrofísicos, foi possível determinar o ruído experimental ao longo da missão, sendo este um importante item para análises com multipolos acima de  $\ell > 1000$ , uma vez que o ruído experimental é anisotrópico. Retornaremos a esta discussão no Capítulo 4.

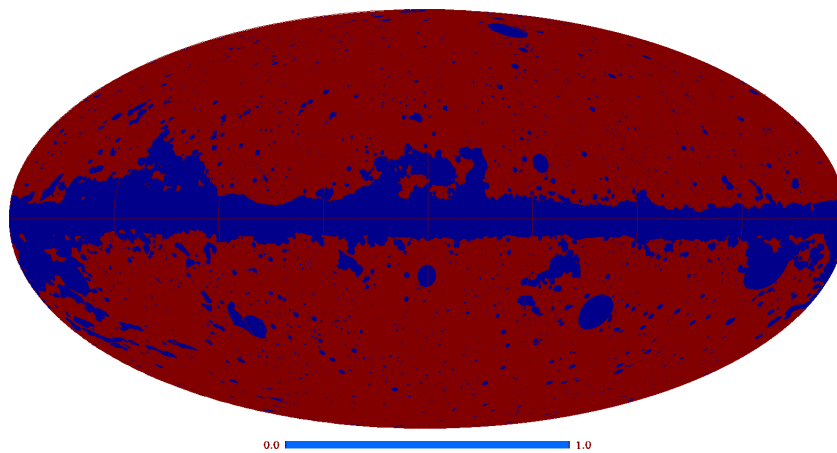
Existem regiões inacessíveis ao satélite devido a diversos fatores, dos quais podemos citar (i) limitações experimentais (por exemplo, a frequência observada); (ii) posicionamento do satélite em relação a galáxia; e (iii) fontes pontuais emissoras de radiação. Utiliza-se então uma máscara, responsável por eliminar tais regiões de maior contaminação denominada máscara UT78 (Figura 5), usada em análises de todas as *pipelines*.

Figura 4 – Efeitos astrofísicos observados pelo satélite que são removidos dos mapas obtidos.



Fonte: Adaptado de <http://sci.esa.int/jump.cfm?oid=56323>, cortesia ESA e a colaboração Planck.

Figura 5 – Máscara UT78. Essa máscara possui uma região de corte maior no centro, onde há grande contribuição do centro da Via Láctea. A fração do céu não mascarada é da ordem de 78%.



Fonte: NASA/JPL-Caltech/ESA.

### 3 Vetores de Multipolo

A missão COBE, pioneira em obter mapas da RCF e estudar sobre o seu espectro de corpo negro, foi capaz de detectar subestruturas na RCF com uma precisão de  $7^\circ$ . Estudos aprofundados relacionados à isotropia do universo foram possíveis devido ao satélite WMAP em 2001 (veja a Figura 2), evidenciando a existência de processos físicos ocorridos durante a formação da RCF. As informações obtidas por esses satélites foram comprimidas em pixels sendo posteriormente convertidas em funções na esfera, juntamente com seus coeficientes multipolares.

Devido ao avanço em relação às medidas das anisotropias da RCF, diferentes técnicas para o tratamento de dados foram propostas. Os autores da referência [3] apresentaram um método de testar a independência estatística dos momentos de multipolo utilizando uma técnica batizada como vetores de multipolo. Para cada momento de multipolo, existem  $2\ell$  vetores unitários correspondentes, sendo  $\ell$  vetores as antípodas dos outros  $\ell$ . Utilizando dados do satélite WMAP, estes autores determinaram os vetores de multipolo para multipolos do intervalo  $2 \leq \ell \leq 8$  e realizaram diversos testes, entre eles envolvendo a soma de produtos internos e externos entre esses vetores. Das combinações testadas, envolvendo produtos internos e externos, a única inconsistente com a isotropia estatística foi a área orientada, sendo esta definida como o alinhamento (produto interno) do produto externo entre os vetores de multipolo, porém nada conclusivo.

Mais tarde no mesmo ano (2004), Weeks [4] salientou que os vetores de multipolo eram conhecidos muito antes. No “*Tratado sobre eletricidade e magnetismo*” de 1873, Maxwell descreveu uma representação dos termos da expansão multipolar como derivadas direcionais da função  $1/r$ , onde  $r = \sqrt{x^2 + y^2 + z^2}$  no espaço. Um exemplo de eletromagnetismo reportado por Weeks foi o de um sistema composto por duas cargas elétricas (i.e., um dipolo) na qual as direções de atração entre cada carga caracterizou-se como sendo os vetores de multipolos. Assim, para  $\ell = 1$  obtêm-se dois vetores de multipolo, sendo apenas um independente. Uma das vantagens em utilizar esta descrição é que as direções das derivadas direcionais rotacionam como vetores, e mais tarde foi provado que elas são únicas e independentes de qualquer sistema de coordenadas (teorema de Sylvester) [4]. Isso entra em contraste com os momentos de multipolo definidos na eq. (2.5), os quais dependem de um sistema de coordenadas. Por esta razão, os vetores de multipolo são uma ferramenta mais robusta para se testar a hipótese de isotropia do universo, pois eles garantem que nenhum referencial espúrio seja introduzido na análise.

Uma demonstração do teorema de Sylvester foi apresentada por [1], utilizando a álgebra de momento angular quântico, transformada integral e integração gaussiana. A prova consiste em descrever uma função qualquer na esfera  $f = f(\theta, \phi)$  como a projeção na base coordenada de um auto-estado quântico de spin inteiro rotacionado para uma direção

arbitrária. Com o estado rotacionado, determina-se os vetores de multipolo em função dos momentos de multipolo  $a_{\ell m}$  introduzidos na eq. (2.5). Pouco mais tarde, [22] calculou analiticamente a função de correlação de dois pontos para vetores de multipolo de qualquer multipolo. Além de determinar a correlação entre esses vetores, foi possível caracterizar algumas propriedades estatísticas, como por exemplo o fato dos vetores de multipolo obedecerem uma distribuição estatística uniforme no contexto do modelo cosmológico padrão. Ao obter os vetores de multipolo usando os dados do WMAP e calcular a função de correlação de dois pontos para diversos multipolos, os autores de [23] constataram que para  $\ell < 20$ , não existe nenhum desvio de isotropia entre os vetores de multipolo obtidos.

Os autores de [2] utilizaram o algoritmo proposto por [1], onde os vetores de multipolo são raízes de um polinômio com coeficientes complexos proporcionais aos coeficientes multipolares  $a_{\ell m}$ . A metodologia utilizada foi diferente da proposta por [3], onde os vetores são obtidos realizando iterações entre momentos de multipolo. A forma do polinômio para a extração dos vetores de multipolo explorada por [1] e [2] com aplicações em cosmologia, era conhecida inicialmente por [24, 25] em estudos relacionados a polinômios aleatórios com algumas aplicações em mecânica quântica caótica de ênfase na mecânica estatística.

A distribuição estatística e a função de correlação de dois pontos das raízes de polinômios com coeficientes aleatórios caracterizados por média e variância foram investigadas por [24, 25]. Um dos casos estudados foi um polinômio da mesma forma algébrica daquele explorado em [1, 2], em que uma conclusão importante deste trabalho foi que, no caso em que os coeficientes multipolares  $a_{\ell m}$  resultam do modelo cosmológico padrão, as raízes do polinômio se distribuem uniformemente na esfera unitária. A função de densidade de probabilidade para as raízes complexas projetadas na esfera unitária foi decomposta em duas funções, usando o sistema de coordenadas esféricas,  $P(\theta)$  e  $P(\phi)$ , dadas por

$$P(\theta) = \frac{\sin \theta}{2}, \quad (3.1)$$

e

$$P(\phi) = \frac{1}{2\pi}. \quad (3.2)$$

Entretanto, [2] reforçou que obter com precisão das raízes de um polinômio seria um problema inviável para multipolos altos, visto que numericamente os polinômios podem ser mal condicionados<sup>1</sup>.

Um algoritmo para a obtenção de raízes de polinômios foi descrito e implementado na referência [26] e é conhecido como `MPSolve`. Um dos objetivos deste programa foi obter com precisão arbitrária as raízes de polinômios onde a amplitude de seus coeficientes fossem elevadas, e em casos de mal condicionamento, aumentar a precisão até obter uma

<sup>1</sup> Polinômios mal condicionados são aqueles em que pequenas variações de seus coeficientes acarretam em uma mudança drástica em suas raízes.

solução adequada. Os autores realizaram uma comparação de performance e precisão para raízes de diversos polinômios utilizando o `Wolfram Mathematica` e o `Maple`, dando destaque ao `MPSolve` em performance e estabilidade. Um dos grandes resultados obtidos foi a obtenção das raízes de um polinômio da ordem 2097151. Uma reestruturação do `MPSolve` com a possibilidade de se obter as raízes utilizando múltiplos núcleos de um processador foi realizada por [27].

Numericamente, nem sempre ao aplicar uma raiz  $\zeta$  em um polinômio  $f$  resulta em  $f(\zeta) = 0$  devido a problemas de arredondamento nos coeficientes do polinômio e mal condicionamento do mesmo e/ou suas raízes. Para certificar-se de que as raízes  $\zeta$  extraídas de um polinômio  $f$  são corretas, [28] implementou um algoritmo chamado `alphaCertified` baseado na Teoria  $\alpha$  de Smale [29], sendo necessário o cálculo de parâmetros envolvendo o polinômio, suas derivadas e raízes.

Posteriormente, diversos testes envolvendo a certificação de raízes de polinômios aleatórios foram estudados por [30], uma vez que esses polinômios estão sujeitos a ruídos em seus coeficientes. Os autores mencionaram diversos métodos utilizados na obtenção das raízes, tais como o método de matrizes companheiras [31] (usado pelo `MatLab`) ou o método Alberth [32] (usado pelo `MPSolve`). Para comparar os dois métodos, foram determinadas as raízes de

$$f(x) = \prod_{j=1}^{10} (x - j), \quad (3.3)$$

usando o `MatLab` e o `MPSolve`. A validade das raízes foram checadas pelo programa `alphaCertified`, sendo que somente o `MPSolve` foi capaz de determiná-las com precisão. Um outro ponto mencionado foi que o `MPSolve` utiliza um outro mecanismo de certificação de raízes, que é necessário para aumentar a precisão numérica quando necessário para evitar problemas de condicionamento.



## 4 Materiais e Métodos

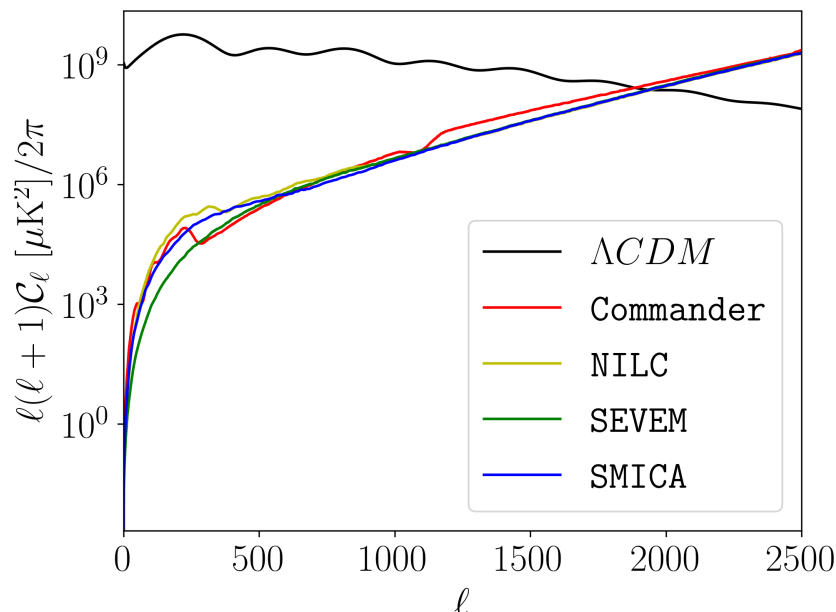
### 4.1 Simulação de Mapas

Para a realização de testes de isotropia do universo com vetores de multipolo em pequenas escalas angulares, foi necessário construir um modelo contendo simulações Monte Carlo de Mapas Isotrópicos e Gaussianos (MIG) da RCF. Para a geração de mapas cosmológicos utilizou-se a função SYNFAST do HEALPix, sendo necessário fornecer a variância dos multipolos, dada pelo espectro angular de potências  $\mathcal{C}_\ell$ , a resolução angular do mapa (NSIDE), o momento de multipolo máximo  $\ell_{máx}$  e uma semente para gerar números aleatórios. Esta última opção foi utilizada para garantir a reprodutibilidade das simulações, caso necessário.

O mesmo procedimento foi aplicado para a simulação de mapas ruído instrumental isotrópico, porém utilizando o espectro de potência dos ruídos de cada *pipeline* do Planck, conforme a Figura 6. A rigor, estes ruídos são anisotrópicos, com intensidade que ultrapassa a da RCF em multipolos maiores de  $\ell$  1750. Para a geração de mapas de ruído instrumental anisotrópico seria necessário levar em consideração aspectos astrofísicos e experimentais do satélite e, portanto, não realizado neste trabalho.

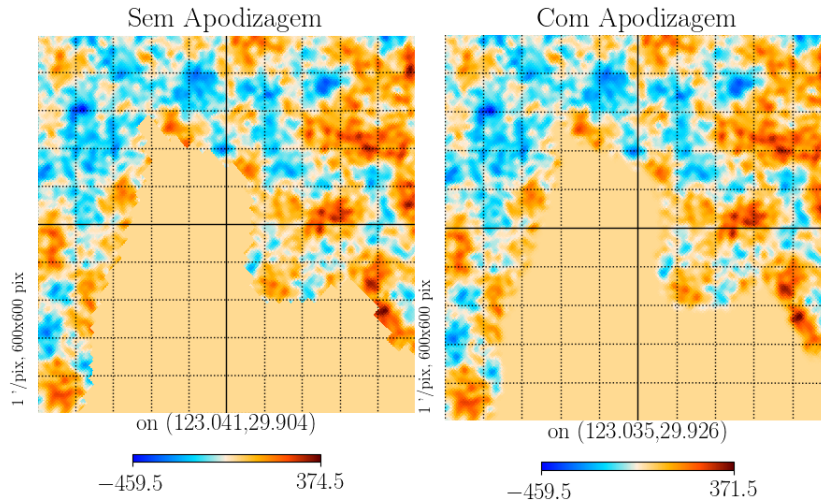
Além de mapas da RCF com ruídos isotrópicos, foi utilizada a máscara UT78

Figura 6 – Comparação de intensidades dos espectros de potências do ruído instrumental de cada *pipeline* do Planck. A linha preta representa o melhor ajuste do modelo  $\Lambda$ CDM aos dados da RCF. As linhas acima à linha preta representam o melhor ajuste mais o ruído instrumental de cada *pipeline*.



Fonte: Adaptado de [21].

Figura 7 – Diferença entre mapas com a aplicação de máscara sem e com apodizagem.



**Fonte:** O próprio autor.

para a remoção de sinais não cosmológicos. Esta máscara é anisotrópica com uma forma aproximadamente azimutal, cobrindo uma boa porção do disco galáctico, conforme a Figura 5. A máscara possui diversos pixels com valores iguais a um ou zero, de forma que a transição entre pixels vizinhos com valores diferentes não é suave. Para que as regiões de transição na máscara sejam suaves, utilizou-se o procedimento de apodizagem, que consiste em aplicar um feixe Gaussiano de 5' de abertura à máscara UT78, o que é feito pela função `SMOOTHING` do software `HEALPix`. Essa suavização é necessária para aliviar efeitos numéricos tais como o fenômeno de Gibbs [33], o qual é evidenciado em regiões de grande contraste de pixels. Para regiões de pixels menores que zero, troca-se os valores para zero e nas regiões de valores maiores que um, para um. Na Figura 7 é possível identificar o efeito de apodizagem em um mapa com a máscara sem e com a apodização.

## 4.2 Extração dos Vetores de Multipolo

Com a construção de MIG com ruídos e a aplicação da máscara UT78 apodizada, obtêm-se os momentos de multipolo  $a_{\ell m}$  de cada um destes mapas com o auxílio da função `MAP2ALM`, sendo necessário especificar o multipolo máximo. O resultado dessa operação é um conjunto de coeficientes multipolares para todos os multipolos de interesse, o que nos permite extrair os vetores de multipolo para cada multipolo usando as raízes do seguinte polinômio [2, 23]

$$\mathcal{P}_\ell(\zeta) = \sum_{m=-\ell}^{\ell} \sqrt{\binom{2\ell}{\ell+m}} a_{\ell m} \zeta^{\ell+m}. \quad (4.1)$$

As raízes de (4.1) foram determinadas pelo software `MPSolve`, sendo necessário implementar um *script* em `python` (Apêndice B) responsável por calcular os coeficientes de (4.1). Para as análises, as raízes foram convertidas em coordenadas angulares na esfera

unitária usando a projeção estereográfica:

$$\zeta = \Re(\zeta) + i\Im(\zeta) = |\zeta| e^{i \arg \zeta} = \cot\left(\frac{\theta}{2}\right) e^{i\phi}, \quad (4.2)$$

sendo  $\theta$  e  $\phi$  ângulos polares definidos nos intervalos  $\theta \in [0, \pi]$  e  $\phi \in [0, 2\pi]$ .

### 4.3 Distribuição dos Vetores de Multipolo

A distribuição de probabilidade dos vetores de multipolo para MIG são dadas pelas funções de densidade de probabilidade nas equações (3.1) e (3.2). Porém, como em  $\theta$ , a densidade de probabilidade não é uniforme (devido à simetria esférica) e que as equações (3.1) e (3.2) não estão normalizadas separadamente, para todas as análises, utilizou-se a seguinte troca de variáveis:

$$\eta \equiv 1 - \cos \theta \quad (4.3)$$

e

$$\varphi \equiv \frac{\phi}{2\pi}, \quad (4.4)$$

sendo

$$P(\eta, \varphi) = \begin{cases} 1 & \eta, \varphi \in [0, 1] \\ 0 & \eta, \varphi \notin [0, 1] \end{cases}. \quad (4.5)$$

As distribuições foram obtidas escolhendo vetores de multipolo situados no hemisfério norte sendo  $\theta \in [0, \pi/2]$ , e conseqüentemente  $P(\theta) = \sin \theta$ . A utilização de apenas um hemisfério para as análises da distribuição é arbitrária, uma vez que  $\ell$  vetores de multipolo são antípodas dos outros  $\ell$ .

#### 4.3.1 Testes Estatísticos

Os vetores de multipolo não dependem da variância das variáveis aleatórias  $a_{\ell m}$ , como demonstramos no Apêndice A.1. É importante enfatizar que esta é uma propriedade crucial para a nossa análise, uma vez que estamos interessados em testes que são independentes do espectro angular de temperatura da RCF. Para testar a hipótese de que os vetores de multipolo também são independentes multipolo a multipolo e que a distribuição das coordenadas destes vetores é dada pela eq. (4.5), utilizou-se o teste de  $\chi^2$  de normalidade para  $\eta$  e  $\varphi$ :

$$\chi_\ell^2 = \frac{1}{\nu - 1} \sum_{i,j=1}^{\nu-1} (f_i - \mu_i) \Sigma_{ij}^{-1} (f_j - \mu_j), \quad (4.6)$$

sendo que para cada simulação, os dados pertencentes ao intervalo  $[0, 1]$  foram subdivididos em  $\nu$  graus de liberdade<sup>1</sup>, com frequência  $f$  com valor esperado  $\mu$  e matriz de covariância

<sup>1</sup> Graus de liberdade também são chamados de categorias, ou binagens (em inglês *bins*).

$\Sigma$ , ambos obtidos através de um conjunto de simulações. Para mais detalhes sobre a implementação do teste de  $\chi^2$ , a obtenção da média  $\mu$  e da matriz de covariância  $\Sigma$  entre as categorias, veja o Apêndice A.2.

Os dados utilizados para o teste de  $\chi^2$  de normalidade foram vetores de multipolo obtidos através de *novos* MIG (mapas de controle) com a adição de ruídos de cada *pipeline* e a máscara UT78 apodizada. Foi necessário realizar novas simulações de controle devido a problemas de estimação da matriz de covariância  $\Sigma$  baseado em simulações. Além disso, distribuição dos vetores de multipolo deixa de ser uniforme após a aplicação da máscara UT78.

Os valores médios dos  $\chi_\ell^2$  dos mapas de controle  $\langle \chi_\ell^2 \rangle$  serão utilizados para a comparação com os  $\chi_\ell^2$  obtidos das *pipelines* do Planck. Para realizar a comparação das simulações com os mapas reais, foi necessário fazer um  $\chi^2$  do  $\chi_\ell^2$ :

$$\chi^2 [\chi_\ell^2] = \sum_{i=1}^{\ell_{\text{máx}}} \frac{(\chi_i^2 - \langle \chi_i^2 \rangle)^2}{\sigma_i^2}, \quad (4.7)$$

onde foi necessário assumir que os multipolos são descorrelacionados (matriz de covariância diagonal),  $\chi_i^2$  são os valores de  $\chi_\ell^2$  de cada *pipeline* do Planck,  $\langle \chi_i^2 \rangle$  e  $\sigma_i^2$  a média e a variância dos  $\chi_\ell^2$  calculados a partir de *novas* simulações. Para este último teste, investigou-se os  $\chi^2$  em diversas escalas cosmológicas tanto para  $\eta$  e  $\varphi$ , obtendo-se diversos valores- $p$  (Apêndice A.3).

## 5 Resultados e Discussão

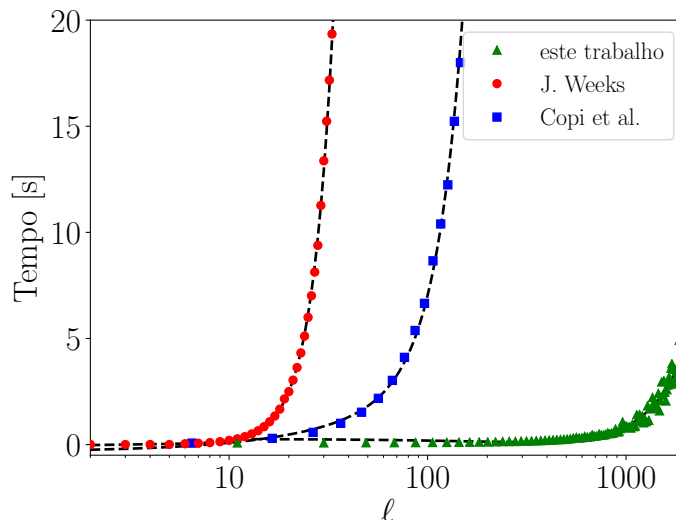
### 5.1 Comparação entre Métodos de Extração dos Vetores de Multipolo

O método utilizado neste trabalho para a obtenção dos vetores de multipolo como raízes de (4.1) foi compatível com a obtenção dos mesmos vetores de multipolo dos mapas abaixo:

- Mapa *Lagrange Internal Linear Combination* (LILC) do primeiro ano de observações do WMAP [34], apresentado por [4];
- Mapa LILC do primeiro ano [34] para  $2 \leq \ell \leq 5$ , Mapa LILC do primeiro ano para  $2 \leq \ell \leq 5$  corrigido por [35] e Mapa LILC do terceiro ano [15] para  $2 \leq \ell \leq 10$  todos provenientes das observações do WMAP investigados por [2];
- Todas as quatro *pipelines* do Planck, no intervalo de  $2 \leq \ell \leq 50$ , analisados por [36].

Um teste de comparação de performance para a obtenção dos vetores de multipolo para diversos multipolos utilizando os métodos propostos por [3, 4, 2] está apresentado na Figura 8. O diferencial deste trabalho em relação a [2] foi a utilização do `MPSolve` ao invés da função `NSolve` utilizada pelo `Wolfram Mathematica`.

Figura 8 – Tempo gasto para obter vetores de multipolo usando diferentes metodologias em um computador com processador Intel(R) Core(TM) i5-600CPU @ 2.70 GHz. Os triângulos verdes, quadrados azuis e pontos vermelhos são referentes à [1, 2], [3] e [4] respectivamente. Acima de  $\ell \gtrsim 900$ , o tempo gasto pelo `MPSolve` aumenta devido ao mal condicionamento dos polinômios já que o software aumenta a precisão para contornar o problema de condicionamento.



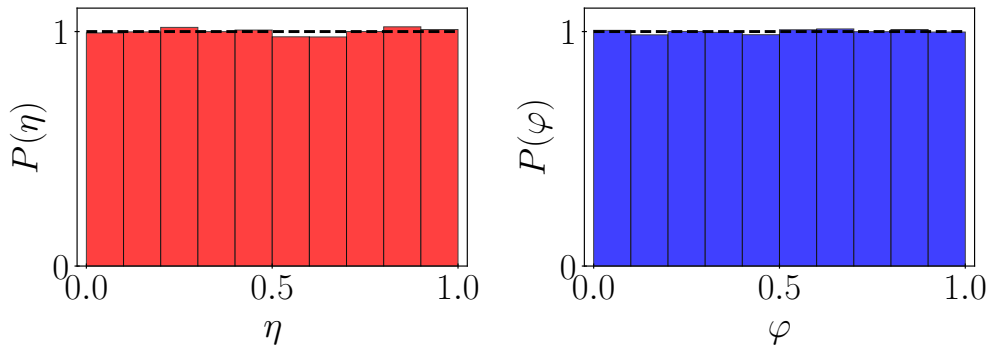
Fonte: O próprio autor.

Uma ressalva em relação a Figura 8 é que os códigos apresentados por [3], [4] e o deste trabalho foram implementados respectivamente em `python`; `Wolfram Mathematica`; e uma mistura entre `python` e `C`. Foi possível extrair todos os 9005992000 vetores de multipolo de 8000 mapas cosmológicos até o multipolo máximo  $\ell_{m\acute{a}x} = 1500$  em aproximadamente um mês usando um servidor equipado com um processador Intel(R) Xeon(R) CPU E5-2650 @ 2.00 GHz e 32 GB de RAM.

## 5.2 Distribuição dos Vetores de Multipolo

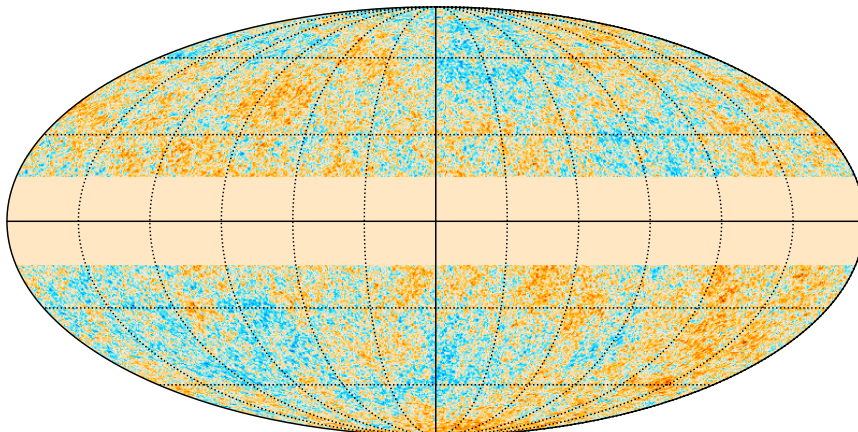
Com a finalidade de verificar se os vetores de multipolo estão uniformemente distribuídos conforme [24, 25], foram simulados 5000 MIG, obtendo-se os vetores de multipolo apenas para o multipolo  $\ell = 10$ , sendo apresentadas as distribuições das coordenadas  $\eta$  e  $\varphi$  na Figura 9 (de acordo com a troca de variáveis em (4.3) e (4.4), respectivamente).

Figura 9 – Distribuição dos vetores de multipolo localizados no hemisfério norte para o multipolo  $\ell = 10$  provenientes de 5000 MIG. A linha tracejada preta representa (4.5). A forma da distribuição é simétrica ao incluir as antípodas.



Fonte: O próprio autor.

Figura 10 – Máscara sintética com uma faixa equatorial de 30° de largura com simetria azimutal.

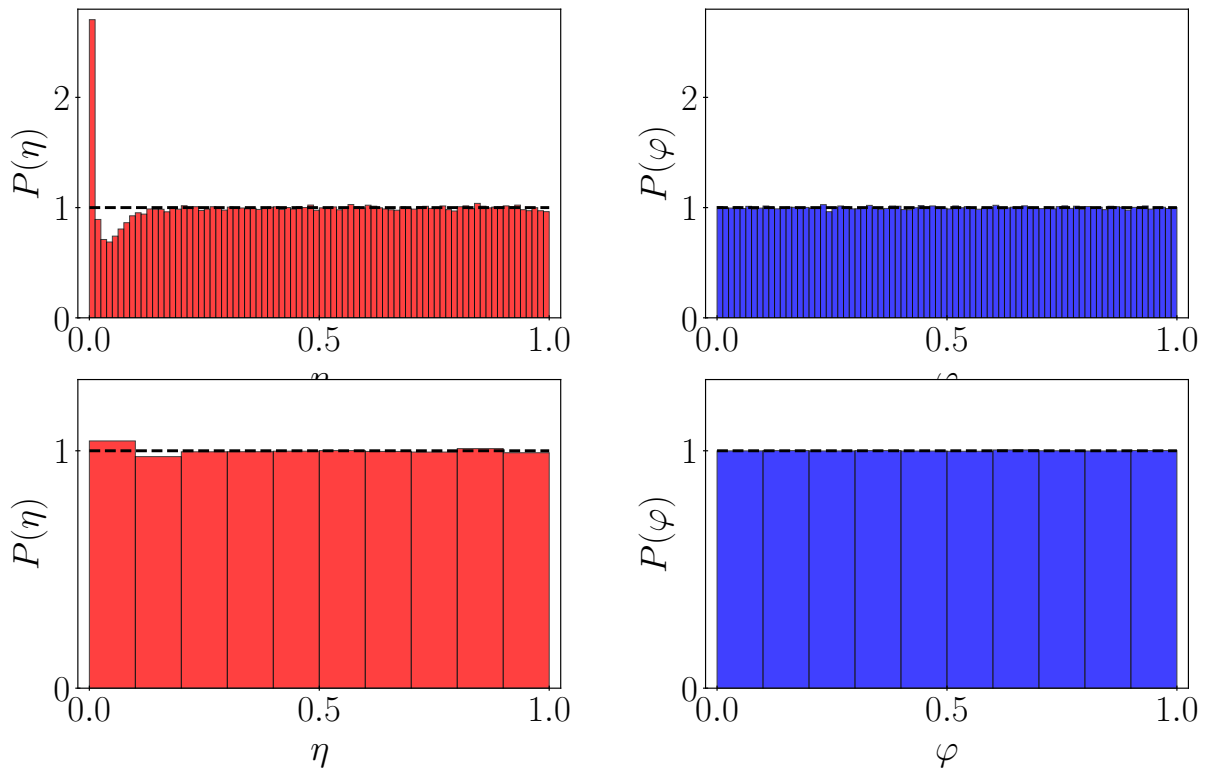


Fonte: O próprio autor.

### 5.2.1 Aplicação de máscaras

Em virtude dos fatores relacionados a contaminações mencionadas no Capítulo 2, a aplicação da máscara é fundamental para análises de efeitos cosmológicos na RCF. Entretanto, máscaras quebram a isotropia de mapas, sendo responsável por influenciar a distribuição dos vetores de multipolo. Como vetores de multipolo são raízes de um polinômio de ordem arbitrária, a interpretação do efeito da máscara nesses vetores não é direta. Para examinar melhor o seu efeito em vetores de multipolo, aplicou-se uma máscara equatorial simétrica (Figura 10) em 5000 MIG, obtendo-se os vetores de multipolo correspondentes a  $\ell = 100$ , obtendo histogramas para as distribuições de  $\eta$  e  $\varphi$  subdividindo os dados em diferentes categorias.

Figura 11 – Efeito da máscara equatorial para a distribuição de  $5 \times 10^5$  vetores de multipolo situados no hemisfério norte. Os dados foram separados uniformemente, sendo 80 *bins* primeira linha e 10 *bins* na segunda linha.

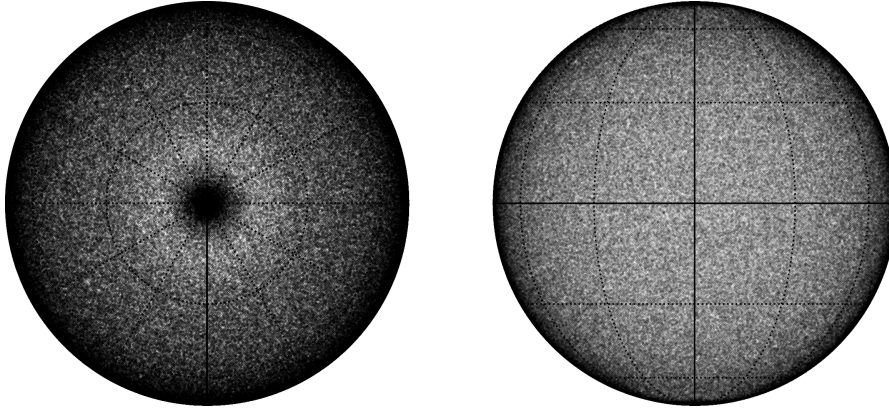


**Fonte:** O próprio autor.

Intuitivamente esperaríamos que a máscara afetasse mais os vetores de multipolo localizados no equador. Nesse sentido o histograma da Figura 11 é um tanto inesperado. Porém, como argumentamos anteriormente, é difícil intuir o impacto que uma característica dos mapas terá nos vetores de multipolo. Além deste efeito, dependendo de como os dados são categorizados, a assinatura da máscara pode acabar sendo perdida (por exemplo, no teste de  $\chi^2$ ), já que o efeito é mais evidente aumentando as subdivisões conforme a primeira coluna da Figura 11. Uma forma de analisar o efeito da máscara, complementar

aos histogramas dos vetores de multipolo, é através de uma projeção destes vetores na esfera:

Figura 12 – Projeção ortográfica dos vetores de multipolo de  $\ell = 100$  obtidos de 5000 MIG mascarados com a máscara da Figura 10. Os  $5 \times 10^7$  vetores de multipolo são representados por pontos na esfera, dando a impressão de um contínuo.



**Fonte:** O próprio autor.

Das projeções ortográficas apresentadas na Figura 12, é evidente a concentração de vetores de multipolo no polo norte (projeção da esquerda de 12). Já na região da máscara, a distribuição dos vetores de multipolo continua sendo uniforme, tornando o efeito não intuitivo.

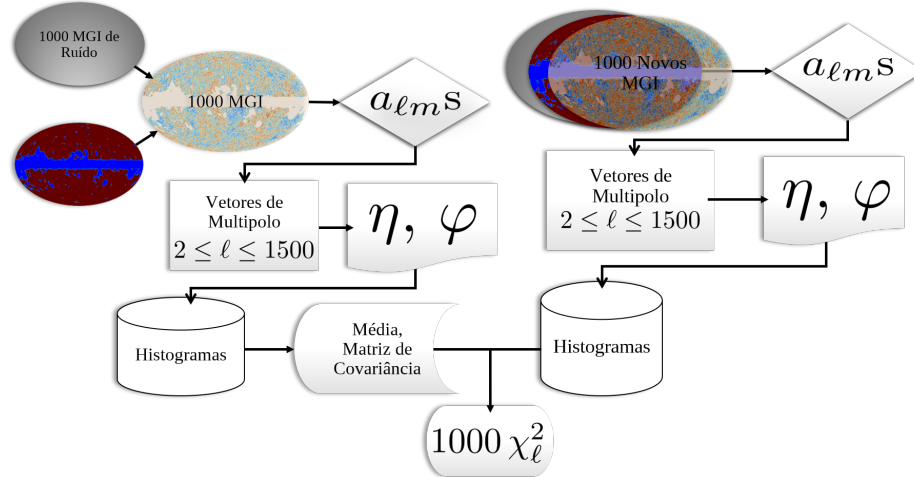
### 5.3 Teste de $\chi^2$ de Normalidade para o Planck

O procedimento para a efetuação do teste de  $\chi^2$  de normalidade envolveu simulações necessárias para estimar a média  $\mu$  e matriz de covariância entre os *bins* de cada multipolo. Um fluxograma apresentado na Figura 13 exemplifica o processo desde a confecção dos mapas até os dados de  $\chi_\ell^2$  para cada multipolo.

A Figura 14 apresenta o teste do  $\chi^2$  usando os vetores de multipolo, sendo esses independentes do espectro angular de potências  $\mathcal{C}_\ell$ . Em outras palavras, o teste representado nesta figura é completamente independente do resultado da Figura 1 e representa o primeiro teste nulo de homogeneidade e isotropia nas escalas angulares analisadas. A variância ( $\mathcal{C}_{\ell s}$ ) é dependente do modelo cosmológico adotado, sendo que neste trabalho para as simulações foi utilizada a variância do melhor ajuste do modelo  $\Lambda$ CDM com seus parâmetros fixados [7] através de dados observados pelo satélite Planck da RCF em conjunto com outros métodos fora do escopo deste trabalho.

Os resultados de  $\chi_\ell^2$  de cada *pipeline* estão apresentados em ordem alfabética em cada linha da Figura 14. A linha e a região verde representam o valor médio de  $\chi_\ell^2$  e um desvio padrão das 1000 simulações de controle.

Figura 13 – Fluxograma de obtenção do teste de  $\chi^2$  de normalidade para cada multipolo até  $\ell_{máx} = 1500$ . Como os vetores de multipolo possuem dois graus de liberdade,  $\eta$  e  $\varphi$ , têm-se duas colunas de resultados sendo elas geradas à partir das distribuições dessas coordenadas.



**Fonte:** O próprio autor.

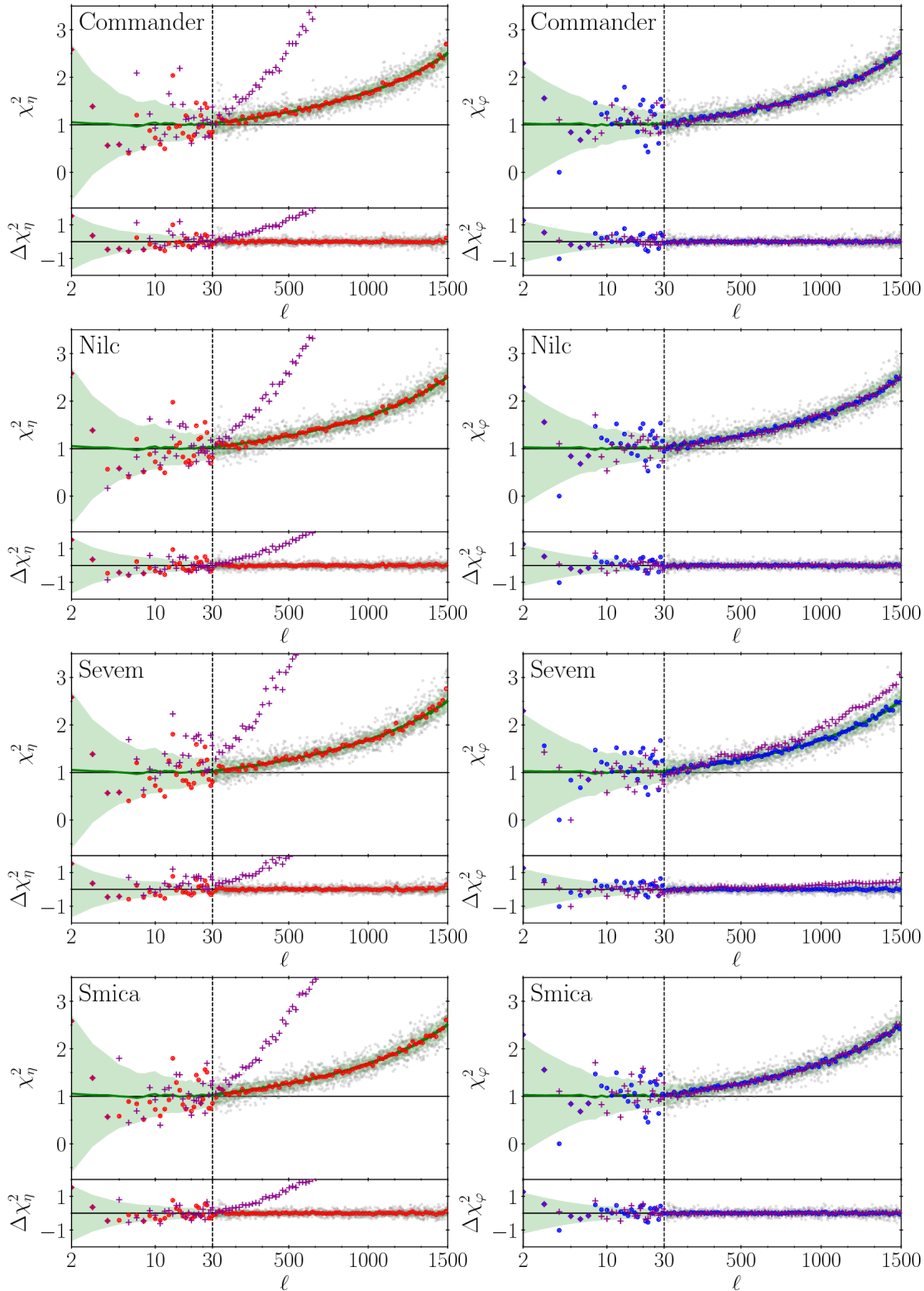
Os pontos vermelhos e azuis representam respectivamente  $\chi^2_\ell$  de  $\eta$  e  $\varphi$  para cada *pipeline*. Nas regiões para  $\ell \geq 30$ , foi realizada uma binagem de 70 pontos para melhor visualização e comparação com a linha verde (estimativa teórica). Entretanto, todos os pontos sem essa compressão estão apresentados em cinza claro, sendo esses os mesmos pontos utilizados para construir os pontos vermelhos.

Além dos dois painéis superiores de cada linha, os painéis inferiores apresentam valores de  $\Delta\chi^2$ , a diferença entre os dados e a média das simulações, para melhor visualização em relação à dispersão e localização de tendências. Para todas as *pipelines*, os pontos flutuaram de forma homogênea em torno de zero.

A fim de checar a consistência do teste, foram simulados quatro MIG com a adição de ruídos isotrópicos de cada *pipeline*, a partir dos quais os valores de  $\eta$  e  $\varphi$  estimados apresentaram uma distribuição uniforme, conforme as equações (4.3) e (4.4). A distribuição dos vetores de multipolo desses mapas foi diferente dos mapas de controle uma vez que não foi utilizado a máscara UT78 responsável por quebrar a isotropia dos mapas. Os pontos roxos em formato de cruz são os resultados de testes de  $\chi^2_\ell$  utilizando a média  $\mu$  e a covariância  $\Sigma$  de mapas não mascarados. Como esperado, os resultados foram desvios dos valores teóricos atribuídos pela curva verde, sugerindo uma distribuição diferente das simulações utilizadas.

A rigor, devido à forma de  $\chi^2_\ell$ , os valores deveriam flutuar em torno de um. Porém, isso não ocorre em consequência dos problemas associados à inversão da matriz de covariância. Uma solução, conforme o Apêndice A.2, seria aumentar o número de simulações para a estimativa da matriz de covariância. Entretanto, isso não foi um problema, pois ambos os resultados, das *pipelines* e das simulações, apresentaram o mesmo comportamento.

Figura 14 – Resultado do teste de  $\chi^2$  de normalidade para  $2 \leq \ell \leq 1500$  para as simulações de controle e todas as *pipelines* do satélite Planck. As curvas e a regiões verdes representam respectivamente a média e a variância para cada multipolo, provenientes dos dados simulados. Os pontos vermelhos e azuis são os  $\chi_\ell^2$  para cada *pipeline*. Já os marcadores em cruz são os resultados do teste de  $\chi^2$  para vetores de multipolo com uma distribuição diferente da dos dados simulados.



Fonte: O próprio autor.

Tabela 2 –  $\chi^2 [\chi_\ell^2]$  sendo uma combinação dos pontos vermelhos e azuis em comparação com as curvas verdes da Figura 14 e seus valores- $p$  associados para diversos intervalos de interesse.

Intervalos	Commander		NILC		SEVEM		SMICA	
	$\chi^2$ -total	valor- $p$	$\chi^2$ -total	valor- $p$	$\chi^2$ -total	valor- $p$	$\chi^2$ -total	valor- $p$
Largas Escalas	1,095	0,288	0,952	0,580	1,009	0,456	0,929	0,630
Escalas do WMAP	0,986	0,634	0,999	0,509	1,012	0,380	0,953	0,877
Escalas do Planck	1,003	0,457	1,003	0,450	1,062	0,009	1,002	0,471

**Fonte:** O próprio autor.

Com exceção de alguns multipolos, todas as *pipelines* do Planck obtiveram seus  $\chi_\ell^2$  dentro do esperado em um desvio padrão das simulações. Para quantificar melhor estes resultados, foram calculados  $\chi^2$  usando a eq. (4.7) para  $\eta$  e  $\varphi$ , cuja a combinação dos resultados (A.10), permitiu a estimativa de valores- $p$  (Apêndice A.3) em várias escalas cosmológicas mostradas na Tabela 2. Por completeza, calculou-se também os valores- $p$  para os dados sem a máscara (marcadores roxos), apresentadas na Tabela 3, sendo que apenas em grandes escalas, os dados são marginalmente compatíveis. Os intervalos referentes às escalas cosmológicas são:

$$\text{Largas Escalas: } 2 \leq \ell \leq 30,$$

$$\text{Escalas do WMAP: } 2 \leq \ell \leq 600,$$

$$\text{Escalas do Planck: } 2 \leq \ell \leq 1500.$$

Um valor- $p$  muito baixo foi obtido para a *pipeline* SEVEM conforme à Tabela 2, sugerindo a não normalidade da distribuição dos vetores de multipolo ao serem comparados com os dados simulados (valor- $p < 0,05$ ). Este resultado corrobora com [36] sugerindo a não utilização desta *pipeline* para análises cosmológicas. Talvez seja necessário uma revisão de como foi feita a separação dos componentes astrofísicos desta *pipeline*.

Tabela 3 – Mesma análise da Tabela 2, porém com dados sem a máscara UT78, representados pelos marcadores roxos em formato de cruz.

Intervalos	Commander		NILC		SEVEM		SMICA	
	$\chi^2$ -total	valor- $p$	$\chi^2$ -total	$p$ -value	$\chi^2$ -total	valor- $p$	$\chi^2$ -total	valor- $p$
Largas Escalas	0,839	0,804	0,583	0,995	1,364	0,034	0,767	0,904
Escalas do WMAP	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
Escalas do Planck	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0

**Fonte:** O próprio autor.



## 6 Conclusão

Mesmo o PC sendo razoável com a presença de catálogos de galáxias, ele ainda precisa ser testado usando outras observações disponíveis aos cosmólogos. Neste trabalho, pela primeira vez, a RCF foi usada para testar o PC usando o formalismo dos vetores de multipolo em escalas sub-angulares, e de maneira completamente independente da física contida no espectro angular de temperatura.

A realização deste teste em pequenas escalas cosmológicas foi possível graças ao avanço de técnicas numéricas para contornar problemas de mal condicionamento de polinômios, sendo que o método apresentado foi o mais eficiente. Isso abriu uma oportunidade para a utilização de um *ensemble* muito maior de simulações, obtendo estatísticas mais confiáveis.

A utilização dos vetores de multipolo também é uma ferramenta motivada geometricamente e mais vantajosa em relação aos momentos de multipolo para testes de isotropia da RCF, uma vez que esses vetores são invariantes por troca de sistema de coordenadas e independem do espectro angular de potências.

O resultado mais importante desde trabalho foi que mesmo em pequenas escalas cosmológicas (multipolos acima de  $\ell > 1000$ ), os mapas de temperatura da RCF de todas as *pipelines* do Planck apresentaram o mesmo comportamento das simulações de controle realizadas sendo uma outra forma de validação do PC.



# Referências

- [1] DENNIS, M. R. Canonical representation of spherical functions: Sylvester's theorem, maxwell's multipoles and majorana's sphere. v. 37, n. 40, p. 9487–9500. ISSN 0305-4470, 1361-6447. Disponível em: <<http://arxiv.org/abs/math-ph/0408046>>.
- [2] HELLING, R. C.; SCHUPP, P.; TESILEANU, T. CMB statistical anisotropy, multipole vectors and the influence of the dipole. v. 74, n. 6. ISSN 1550-7998, 1550-2368. Disponível em: <<http://arxiv.org/abs/astro-ph/0603594>>.
- [3] COPI, C. J.; HUTERER, D.; STARKMAN, G. D. Multipole vectors: A new representation of the CMB sky and evidence for statistical anisotropy or non-gaussianity at  $2 \leq \ell \leq 8$ . *Physical Review D*, American Physical Society (APS), v. 70, n. 4, aug 2004. ISSN 1550-7998, 1550-2368. Disponível em: <<https://link.aps.org/doi/10.1103/PhysRevD.70.043515>>.
- [4] WEEKS, J. R. Maxwell's multipole vectors and the CMB. Disponível em: <<http://arxiv.org/abs/astro-ph/0412231>>.
- [5] DODELSON, S. *Modern Cosmology*. Elsevier Science, 2003. ISBN 9780122191411. Disponível em: <<https://books.google.com.br/books?id=3oPRxdXJexcC>>.
- [6] RIESS, A. G. et al. Observational evidence from supernovae for an accelerating universe and a cosmological constant. *Astron. J.*, v. 116, p. 1009–1038, 1998.
- [7] AGHANIM, N. et al. Planck 2018 results. VI. Cosmological parameters. 2018.
- [8] MUKHANOV, V. *Physical foundations of cosmology*. [S.l.]: Cambridge university press, 2005.
- [9] KENT, S. M. Sloan digital sky survey. *Astrophysics and Space Science*, Springer, v. 217, n. 1-2, p. 27–30, 1994.
- [10] COLLESS, M. et al. The 2df galaxy redshift survey: spectra and redshifts. *Monthly Notices of the Royal Astronomical Society*, Blackwell Science Ltd Oxford, UK, v. 328, n. 4, p. 1039–1063, 2001.
- [11] DURRER, R. *The Cosmic Microwave Background*. [S.l.]: Cambridge University Press, 2008.
- [12] PENZIAS, A. A.; WILSON, R. W. A measurement of excess antenna temperature at 4080 mc/s. *The Astrophysical Journal*, v. 142, p. 419–421, 1965.

- [13] WEINBERG, S.; TODD, R. *The first three minutes*. [S.l.]: Blackstone Audio, Incorporated, 2005.
- [14] SMOOT, G. F. et al. Structure in the COBE differential microwave radiometer first-year maps. *The Astrophysical Journal*, v. 396, p. L1–L5, 1992.
- [15] HINSHAW, G. et al. Three-year Wilkinson Microwave Anisotropy Probe (WMAP) observations: temperature analysis. *Astrophys. J. Suppl.*, v. 170, p. 288, 2007.
- [16] ABRAMO, L. R.; PEREIRA, T. S. Testing gaussianity, homogeneity and isotropy with the cosmic microwave background. *Adv. Astron.*, v. 2010, p. 378203, 2010.
- [17] ADE, P. A. R. et al. Planck 2015 results. XXVII. The Second Planck Catalogue of Sunyaev-Zeldovich Sources. *Astron. Astrophys.*, v. 594, p. A27, 2016.
- [18] TAUBER, J. et al. The Scientific programme of Planck. 2006.
- [19] THE HEALPix Primer. Disponível em: <<http://healpix.sourceforge.net>>.
- [20] GORSKI, K. M. et al. HEALPix – a framework for high resolution discretization, and fast analysis of data distributed on the sphere. *The Astrophysical Journal*, IOP Publishing, v. 622, n. 2, p. 759–771, apr 2005. ISSN 0004-637X, 1538-4357. Disponível em: <<http://arxiv.org/abs/astro-ph/0409513>>.
- [21] COLLABORATION, P. et al. Planck 2015 results. IX. diffuse component separation: CMB maps. *Astronomy & Astrophysics*, EDP Sciences, v. 594, p. A9, sep 2016. ISSN 0004-6361, 1432-0746. Disponível em: <<http://arxiv.org/abs/1502.05956>>.
- [22] DENNIS, M. R. Correlations between maxwell’s multipoles for gaussian random functions on the sphere. *Journal of Physics A: Mathematical and General*, IOP Publishing, v. 38, n. 8, p. 1653–1658, feb 2005. ISSN 0305-4470, 1361-6447. Disponível em: <<http://arxiv.org/abs/math-ph/0410004>>.
- [23] DENNIS, M. R.; LAND, K. Probability density of the multipole vectors for a gaussian cosmic microwave background. v. 383, n. 2, p. 424–434. ISSN 00358711. Disponível em: <<http://arxiv.org/abs/0704.3657>>.
- [24] BOGOMOLNY, E.; BOHIGAS, O.; LEBOEUF, P. Distribution of roots of random polynomials. *Physical Review Letters*, American Physical Society (APS), v. 68, n. 18, p. 2726–2729, 1992. ISSN 0031-9007. Disponível em: <<https://link.aps.org/doi/10.1103/PhysRevLett.68.2726>>.
- [25] BOGOMOLNY, E.; BOHIGAS, O.; LEBOEUF, P. Quantum chaotic dynamics and random polynomials. *Journal of Statistical Physics*, Springer Nature, v. 85, n. 5, p. 639–679, 1996. Disponível em: <<http://link.springer.com/10.1007/BF02199359>>.

- [26] BINI, D. A.; FIORENTINO, G. Design, analysis, and implementation of a multiprecision polynomial rootfinder. *Numerical Algorithms*, Springer Nature, v. 23, n. 2, p. 49, 2000.
- [27] ROBOL, L. *A rootfinding algorithm for polynomials*. Tesi di Laurea. Disponível em: <<https://leonardo.robol.it/files/robol-thesis.pdf>>.
- [28] HAUENSTEIN, J. D.; SOTTILE, F. *Algorithm XXX: alphaCertified: Certifying Solutions to Polynomial Systems*. 2011.
- [29] SMALE, S. *Newton's method estimates from data at one point. The merging of disciplines: new directions in pure, applied, and computational mathematics (Laramie, Wyo., 1985), 185–196*. [S.l.]: Springer, New York, 1986.
- [30] CLEVELAND, J. et al. Certified counting of roots of random univariate polynomials. *arXiv preprint arXiv:1412.1717*, 2014.
- [31] HORN, R. A.; JOHNSON, C. R. *Matrix analysis*. Cup. [S.l.]: Cambridge University Press, 1990. ISBN 9780521386326,0521386322.
- [32] ABERTH, O. Iteration methods for finding all zeros of a polynomial simultaneously. *Mathematics of computation*, v. 27, n. 122, p. 339–344, 1973.
- [33] HEWITT, E.; HEWITT, R. E. The gibbs-wilbraham phenomenon: An episode in fourier analysis. *Archive for History of Exact Sciences*, v. 21, n. 2, p. 129–160, Jun 1979. ISSN 1432-0657. Disponível em: <<https://doi.org/10.1007/BF00330404>>.
- [34] ERIKSEN, H. K. et al. On foreground removal from the wilkinson microwave anisotropy probe data by an internal linear combination method: Limitations and implications. *The Astrophysical Journal*, v. 612, n. 2, p. 633, 2004. Disponível em: <<http://stacks.iop.org/0004-637X/612/i=2/a=633>>.
- [35] TEGMARK, M.; OLIVEIRA-COSTA, A. de; HAMILTON, A. A high resolution foreground cleaned CMB map from WMAP. *Phys. Rev.*, D68, p. 123523, 2003.
- [36] PINKWART, M.; SCHWARZ, D. J. Test of the cosmological principle by means of CMB multipole vectors for  $l \leq 50$ . Disponível em: <<http://arxiv.org/abs/1803.07473>>.



# Apêndices



# APÊNDICE A – Vetores de Multipolo e Teste de $\chi^2$

## A.1 Independência dos $\mathcal{C}_\ell$ nos Vetores de Multipolo

Seja duas variáveis aleatórias  $X$  e  $Z$  onde<sup>1</sup>  $X \sim \mathcal{N}(\mu, \sigma^2)$  e  $Z \sim \mathcal{N}(0, 1)$  de forma que a relação entre essas variáveis aleatórias é dada por

$$Z = \frac{(X - \mu)}{\sigma}, \quad (\text{A.1})$$

uma vez que

$$\langle Z \rangle = \frac{\langle X - \mu \rangle}{\sigma} = 0, \quad (\text{A.2})$$

e

$$\sigma_Z^2 = \langle Z^2 \rangle - \langle Z \rangle^2 = \frac{1}{\sigma^2} (\langle X^2 \rangle - 2\mu^2 + \mu^2) = \frac{1}{\sigma^2} (\langle X^2 \rangle - \langle X \rangle^2) = 1. \quad \blacksquare \quad (\text{A.3})$$

No contexto cosmológico, os momentos de multipolo têm  $\mu = \langle a_{\ell m} \rangle = 0$  e  $\sigma^2 = \langle |a_{\ell m}|^2 \rangle = \mathcal{C}_\ell$ . Ao substituirmos a variância dos momentos de multipolo  $\sigma^2 \rightarrow \sigma_Z^2$ , basta utilizarmos (A.1) de forma que

$$Z = \frac{\sigma_Z}{\sigma} X \implies \tilde{a}_{\ell m} = \sqrt{\frac{\tilde{\mathcal{C}}_\ell}{\mathcal{C}_\ell}} a_{\ell m}, \quad (\text{A.4})$$

onde  $\sigma_Z^2 \equiv \tilde{\mathcal{C}}_\ell$ . Como o termo  $\sqrt{\tilde{\mathcal{C}}_\ell/\mathcal{C}_\ell}$  de (A.4) é global, ao substituirmos em (4.1),

$$\sqrt{\frac{\tilde{\mathcal{C}}_\ell}{\mathcal{C}_\ell}} \sum_{m=-\ell}^{\ell} \sqrt{\binom{2\ell}{\ell+m}} \tilde{a}_{\ell m} \zeta^{\ell+m} = 0, \quad (\text{A.5})$$

as raízes de (A.5) permanecem inalteradas, portanto vetores de multipolo são independentes da variância.

## A.2 Teste de $\chi^2$ de Normalidade

A definição do teste de  $\chi^2$  de normalidade é dada por (4.6), sendo um dos requisitos separar arbitrariamente os dados em categorias para obter a frequência  $f$  de cada um, sendo necessário usar apenas vetores de multipolo no hemisfério norte devido à simetria antipodal.

<sup>1</sup> A notação utilizada para descrever uma variável aleatória  $Y$  com uma distribuição normal  $\mathcal{N}$  com média  $\mu$  e variância  $\sigma^2$  é  $Y \sim \mathcal{N}(\mu, \sigma^2)$ . Não confundir com  $Y$  aproximadamente à uma distribuição normal.

**Exemplo:**

Para o multipolo  $\ell = 10$ , têm-se 10 vetores de multipolo situados no hemisfério norte, cada um com coordenadas  $\eta$  e  $\varphi$ . Os intervalos de  $\eta$  e  $\varphi$  são subdivididos hipoteticamente em 10 categorias, sendo agrupadas as coordenadas dos 10 vetores de multipolo em cada uma delas. A frequência  $f$  de cada bin é o número de vezes as coordenadas de um vetor de multipolo é separada em uma categoria sobre o número total de vetores de multipolo.

A categorização dos ângulos dos vetores de multipolo ( $\theta$  e  $\phi$ ) é uniforme, tendo o mesmo tamanho para cada subintervalo. Devido à forma de  $\eta$  em (4.3), para garantir que a categorização ainda seja uniforme em  $\theta$ , é necessário definir o tamanho de cada categoria de  $\eta$  sendo

$$\Delta\eta = \sin\theta\Delta\theta, \quad (\text{A.6})$$

tal que o tamanho de cada categoria não seja uniforme em  $\eta$ . Para  $\phi$  e  $\varphi$  nada muda pois  $\Delta\varphi = \Delta\phi$ .

Para estimar a frequência teórica  $\mu$ , foi necessário construir um modelo para cada *pipeline* contendo 1000 simulações e fazer a média em cada *bin*. Esse procedimento não analítico é uma alternativa razoável uma vez que a isotropia dos vetores de multipolo é quebrada devido à máscara UT78 em MIG conforme a seção 4.1 do Capítulo 4. A matriz de covariância teórica  $\Sigma$  é definida como

$$\Sigma \equiv \frac{1}{N-1} \sum_{i,j=1}^{\nu-1} (f_i - \mu_i)(f_j - \mu_j), \quad (\text{A.7})$$

sendo a média de  $N$  simulações contendo  $N$  matrizes de covariância construídas através das  $\nu - 1$  categorias (*bins* ou graus de liberdade). Usar  $\nu - 1$  categorias é necessário pois  $[0, 1]$  é um intervalo compacto (inclui as bordas). Seguindo o exemplo acima, se subdividirmos o intervalo em 10 subintervalos, a última subdivisão por exemplo será uma combinação das outras, uma vez que se nenhuma coordenada do vetor de multipolo cair nas outras 9 categorias, com certeza irá cair na décima. Os problemas em estimar a matriz de covariância teórica numericamente é que seria necessário  $N \rightarrow \infty$ , e também foram utilizadas 1000 simulações para estimar  $\Sigma$ , sendo que devido a um teorema de álgebra linear, só é possível inverter  $\Sigma$  se os subintervalos forem até de 999 *bins*, necessário para o cálculo de  $\chi_\ell^2$ .

Em consequência aos complicadores para a estimação de  $\Sigma$ , as subdivisões multipolo à multipolo para obter  $\mu$  de  $\eta$  e  $\varphi$  no  $[0, 1]$  obedeceram à seguinte regra: para  $\ell \leq 30$  (grandes escalas),  $\ell$  categorias e para  $\ell \geq 30$ , separar em  $^{570}/_{1470}(\ell - 30) + 30$  tal que para  $\ell = 1500$ , o número de categorias seja 600 *bins*. A escolha de 600 ao invés de 999 *bins* foi estratégica, já que para multipolos altos o efeito da máscara ainda é capturado, e a inversão da matriz de covariância é numericamente mais comportada.

### A.3 Valores- $p$

Uma forma de quantificar o quão provável uma hipótese é em relação à hipótese nula é utilizando o valor- $p$ , sendo este definido como

$$p \equiv 1 - \int_0^v P(x) dx, \quad (\text{A.8})$$

sendo  $P(x)$  a distribuição dos dados e  $v$  o valor da distribuição calculada através de dados. O valor- $p$  fornece a probabilidade de que a hipótese nula seja verdadeira, porém não necessariamente de que os dados são provenientes desta hipótese. Comumente, rejeita-se a hipótese nula caso valor- $p < 0,05$ .

A distribuição dos dados obtidos das simulações conforme a Figura 14 é uma distribuição de  $\chi^2$  sendo esta uma função dependente do número de graus de liberdade  $\nu$

$$P(\nu, x) = \frac{x^{\frac{\nu}{2}-1} e^{-\frac{x}{2}}}{2^{\frac{\nu}{2}} \Gamma\left(\frac{\nu}{2}\right)}. \quad (\text{A.9})$$

Para o efetuar o cálculo de (A.8) contido nas Tabelas 2 e 3, foi necessário calcular (4.7) para cada escala analisada, ou seja, intervalo de multipolo. Como as variáveis  $\eta$  e  $\varphi$  são independentes, para realizar a combinação delas e obter apenas um valor- $p$  foi necessário somar as contribuições de (4.7) para cada variável, sendo

$$v \equiv \chi_\eta^2 [\chi_\ell^2] + \chi_\varphi^2 [\chi_\ell^2]. \quad (\text{A.10})$$



## APÊNDICE B – Códigos

Todos os códigos usados neste trabalho foram desenvolvidos usando a linguagem Python 3.6 (<https://anaconda.org>). Além disso, foi utilizado o software MPSolve (<https://github.com/robol/MPSolve>).

- Pacotes básicos:

```
import warnings
warnings.filterwarnings('ignore')
import healpy as hp
import numpy as np
import gc
import gmpy2
import os
from gmpy2 import sqrt, bincoef
from numba import jit, i8, f8
from scipy.misc import factorial
from scipy.optimize import minimize
from scipy.special import gamma, binom
```

- Vetores de Multipolo:

```
def pol_mpsolve(alms, l, sample):
    """
    Generates *.pol file with instructions for MPSolve.

    Args:
        alms (complex array): multipole moments from a map.
        l (int): multipole.
        sample (int): additional number that is useful for
            parallel computing. Assures no confusion ]
            while reading *.pol file.

    Returns:
        *.pol file.
    """
    os.makedirs('tmp', mode = 0o777, exist_ok = True)
```

```

lmax = hp.sphtfunc.Alm.getlmax(len(alms))

header = ['! We will evaluate the roots for l = {}'.format(l),
         ' ',
         'Monomial;',
         'Degree={}'.format(2*l),
         'Complex;',
         'FloatingPoint;',
         ' ']

# Instructions for MPSolve:
m_array = np.arange(- l, l + 1, 1) # m values.
neg_m = m_array[0:l] # Negative m values.
pos_m = m_array[l:] # Positive m values.

binom_term = [sqrt(bincoef(2*l, l + i)) for i in range(- l, l + 1)]

index = hp.sphtfunc.Alm.getidx(lmax, l, abs(m_array))
alm_ell_real = np.real(alms[index])
alm_ell_imag = np.imag(alms[index])

neg_rea_m_term = [ (- 1)**int(i) for i in neg_m]
neg_imag_m_term = [ - (- 1)**int(i) for i in neg_m]

rea_sign = np.concatenate((neg_rea_m_term, np.ones(l + 1)))
imag_sign = np.concatenate((neg_imag_m_term, np.ones(l + 1)))

rea = rea_sign*alm_ell_real*binom_term # Real terms.
ima = imag_sign*alm_ell_imag*binom_term # Imaginary terms.

del m_array, neg_m, pos_m, binom_term, index
del alm_ell_real, alm_ell_imag, neg_rea_m_term, neg_imag_m_term
del rea_sign, imag_sign

complex_ = rea + 1j*ima # Get complex terms.

polycoefs = ['{0:..ME}'.format(complex_[i]).\
             replace('(', '').replace(')', '\
             for i in range(2*l + 1)]

```

---

```

del complex_, rea, ima

file_to_save = open('tmp/coeffs_mpsolve_{:04d}_{}.pol'.\
                    format(l, sample), 'w')

for item in header + polycoeffs:
    file_to_save.write("%s\n" % item)

del file_to_save, polycoeffs

def m_vectors(alms, l, sample, all_mpsvs = None):
    """
    Get multipole vectors in spherical coordinates for a multipole.

    Args:
        alms (complex array): multipole moments from a map.
        l (int): multipole.
        sample (int): additional number that is useful for parallel
            computing. Assures no confusion on
            while reading *.pol file.
        all_mpsvs (bool): if True, returns all multipole vectors.
            If False, returns only north hemisphere
            multipole vectors.

    Returns:
        Float array [theta, phi] in radians.
    """
    # Generate polynomials:
    pol_mpsolve(alms, l, sample)

    # Call MPSolve:
    os.system('mpsolve -Ob tmp/coeffs_mpsolve_{:04d}_{}.pol >
              tmp/roots_mpsolve_{:04d}_{}.txt'\
              .format(l, sample, l, sample))

    # Get roots, and write multipole vectors in spherical coordinates:
    roots_mpsolve = np.loadtxt('tmp/roots_mpsolve_{:04d}_{}.txt'.\
                               format(l, sample))

```

```

roots_mpsolve = np.vectorize(complex)(roots_mpsolve[...,0],\
                                   roots_mpsolve[...,1])

phi = np.angle(roots_mpsolve)
r = np.absolute(roots_mpsolve)
theta = 2*np.arctan(1/r)

del roots_mpsolve, r

m_vec = np.vstack((theta, phi)).T

del phi, theta

os.remove('tmp/coeffs_mpsolve_{:04d}_{}.pol'.format(l, sample))
os.remove('tmp/roots_mpsolve_{:04d}_{}.txt'.format(l, sample))

if all_mpvs == True:
    return m_vec
    del m_vec

else:
    return m_vec[m_vec[:, 0] < np.pi/2]
    del m_vec

def many_m_vectors(alms, lmin, lmax, sample, all_mpvs = None):
    """
    Get multiple multipole vectors in spherical coordinates.

    Args:
        alms (complex array): multipole moments from a map.
        lmin (int): minimum multipole.
        lmax (int): maximum multipole.
        sample (int): additional number that is useful for parallel
            computing. Assures no confusion
            while reading *.pol file.
        all_mpvs (bool): if True, returns all multipole vectors.
            If False, returns only north hemisphere mus.

    Returns:

```

---

```

        Float array [theta, phi] in radians.
    """
    return np.concatenate([m_vectors(alms, i, sample, all_mpv) \
                           for i in range(lmin, lmax + 1)])

def extract_mvs(start, end, alms, mvs_out_folder, mvs_out):
    """
    Get multiple multipole vectors for severam monte carlo simulations.

    Args:
        start (int): initial simulation.
        end (int): final simulation.
        alms (string): name of multipole moments for an specific
            simulation.
        mvs_out_folder (string): name of multipole vector folder that
            they will be stored.
        mvs_out (string): name of multipole vectors that will be saved.

    Returns:
        Nothing. All files will be saved during inner loops.
    """
    for i in range(start, end + 1):
        alm = np.load(alms.format(i))
        os.makedirs(mvs_out_folder.format(i), mode = 0o777, exist_ok = True)
        os.chdir(mvs_out_folder.format(i))
        for j in range(2, 1500 + 1):
            mvs = m_vectors(alm, j, i, True)
            np.save(mvs_out.format(i, j), mvs)
            del mvs
        del alm

```

- Funções variadas:

```

def get_multipole(alms, l):
    """
    Get specific multipole.

    Args:
        alms (complex array): multipole moments from a map.
        l (int): multipole.

```

*Returns:*

*Complex array.*

"""

```
ells, ms = hp.Alm.getlm(hp.Alm.getlmax(len(alms)))
```

```
return alms[np.where(ells == 1)]
```

```
def to_cart(theta_phi_array):
```

"""

*Converts spherical to cartesian.*

*Args:*

*theta\_phi\_array (float array): array containing in radians  
[theta, phi].*

*Returns:*

*Float array.*

"""

```
x = np.sin(theta_phi_array[:, 0])*np.cos(theta_phi_array[:, 1])
```

```
y = np.sin(theta_phi_array[:, 0])*np.sin(theta_phi_array[:, 1])
```

```
z = np.cos(theta_phi_array[:, 0])
```

```
return np.vstack((x, y, z)).T
```

```
def get_mvs_l(mpvs, l, lmin, lmax, all_mpvs = None):
```

"""

*Extract specific multipole vectors.*

*Only works for array containing sequences of 2\*l multipole vectors  
and multipole vectors on spherical coordinates.*

*Args:*

*mpvs (float array): array containing multipole  
vectors [theta, phi].*

*l (int): multipole.*

*lmin (int): minimum multipole.*

*lmax (int): maximum multipole.*

*all\_mpvs (bool): if True, returns all multipole vectors.*

*If False, returns only north hemisphere  
multipole vectors.*

*Returns:*

*Float array [theta, phi] in radians.*

```

"""
indexes_ = np.repeat(np.arange(lmin, lmax + 1, 1), \
                    2*np.arange(lmin, lmax + 1, 1))
mpvs_ = np.vstack((indexes_, np.array(mpvs).T)).T
result_0 = mpvs_[mpvs_[:, 0] == 1][:, 1:]

if all_mpvs == True:
    return result_0
else:
    return result_0[result_0[:, 0] < np.pi/2]

```

- Teste de certificação  $\alpha$ :

```

def roots_testing(alms, l, sample):
    """
    Get roots from the polynomial we obtain multipole vectors.

    Args:
        alms (complex array): multipole moments from a map.
        l (int): multipole.
        sample (int): additional number that is useful for
            parallel computing. Assures no confusion
            while reading *.pol file.

    Returns:
        Complex array containing 2*l roots.
    """
    # Polynomials for MPSolve:
    pol_mpsolve(alms, l, sample)

    # Call MPSolve:
    os.system('mpsolve -Ob tmp/coeffs_mpsolve_{:04d}_{}.pol >
              tmp/roots_mpsolve_{:04d}_{}.txt'\
              .format(l, sample, l, sample))

    # Arrange roots:
    roots_mpsolve = np.loadtxt('tmp/roots_mpsolve_{:04d}_{}.txt'.\
                               format(l, sample))

```

```

roots_mpsolve = roots_mpsolve[... ,0] + 1j*roots_mpsolve[... ,1]
roots_mpsolve = [gmpy2.mpc(i) for i in roots_mpsolve]
return roots_mpsolve

def alpha_polycoefs(alms, l):
    """
    Write polynomial coefficients to test roots from previous function.

    Args:
        alms (complex array): multipole moments from a map.
        l (int): multipole.

    Returns:
        Complex array.
    """
    lmax = hp.sphtfunc.Alm.getlmax(len(alms)) # Get lmax given an array.

    m_array = np.arange(- l, l + 1, 1) # m values.
    neg_m = m_array[0:l] # Negative m values.
    pos_m = m_array[l:] # Positive m values.

    binom_term = [sqrt(bincoef(2*l, l + i)) for i in range(- l, l + 1)]

    index = hp.sphtfunc.Alm.getidx(lmax, l, abs(m_array))

    alm_ell_real = np.real(alms[index])
    alm_ell_imag = np.imag(alms[index])

    neg_rea_m_term = [ (- 1)**int(i) for i in neg_m]
    neg_imag_m_term = [ - (- 1)**int(i) for i in neg_m]

    rea_sign = np.concatenate((neg_rea_m_term, np.ones(l + 1)))
    imag_sign = np.concatenate((neg_imag_m_term, np.ones(l + 1)))

    rea = rea_sign*alm_ell_real*binom_term # Real terms.
    ima = imag_sign*alm_ell_imag*binom_term # Imaginary terms.
    complex_ = rea + 1j*ima # Get complex terms.

    polycoefs = [complex_[i] for i in range(2*l + 1)]

```

---

```

return polycoefs[::-1]

def alpha_beta_func(alms, l, root):
    """
    Auxiliary function for this test.

    Args:
        alms (complex array): multipole moments from a map.
        l (int): multipole.
        root (complex array): roots for testing.

    Returns:
        Float number.
    """
    poly = np.poly1d(alpha_polycoefs(alms, l))
    der = np.polyder(poly, m = 1) # First derivative of polynomial.
    return abs(poly(root)/der(root))

def alpha_gamma_func(alms, l, root):
    """
    Auxiliary function for this test.

    Args:
        alms (complex array): multipole moments from a map.
        l (int): multipole.
        root (complex array): roots for testing.

    Returns:
        Float number.
    """
    poly = np.poly1d(alpha_polycoefs(alms, l))
    test = [] # Maximum value.
    for k in range(2, 2*l + 1):
        term1 = np.polyder(poly, m = k)
        term2 = np.polyder(poly, m = 1)
        test.append(abs(term1(root)/(term2(root)*\
                        gmpy2.factorial(k)))*(1/(k - 1)))
    test = np.max(np.array(test), axis = 0)
    return test

```

```

def alpha_test(alms, l, root):
    """
    Alpha certified test. This test is based in Smale's alpha theory.

    Args:
        alms (complex array): multipole moments from a map.
        l (int): multipole.
        root(complex array): roots for testing.

    Returns:
        Float number.
    """
    return alpha_beta_func(alms, l, root)*\
           alpha_gamma_func(alms, l, root)

```

- Teste de  $\chi^2$ :

```

def chi2_theory(theta, phi, n_sim, n_bins, test_type):
    """
    Evaluates the reduced chi-squared test given a set of data.

    Args:
        theta (float array): array that contains thetas.
        phi (float array): array that contains phis.
        n_sim (int): number of simulations.
        n_bins (int): number of bins.
        test_type (bool): if True, it returns a normality chi-squared
                           test. If False, it returns a test for
                           poisson data.

    Returns:
        Array containing reduced chi-squared values [theta, phi] for
        n_sim simulations, covariance matrix, and mean for
        theta and phi. This second return is usefull to compare
        with real data.
    """
    # New variables:
    eta = 1 - np.cos(theta)
    var_phi = phi/(2*np.pi)

```

---

```

var_phi[np.where(var_phi < 0)] = var_phi[np.where(var_phi < 0)] + 1

# Bins Edges:
# Since the mask we aim to apply on our data have a
# semi-pseudo azimuthal symmetry, we will bin theta in an uniform
# way leading that eta is not unniiform (bins with different width).
theta_bins = np.histogram(theta, bins=n_bins, range=(0,np.pi/2),\
                           density=1)[1]
eta_bins = np.cumsum(np.sin(theta_bins)*np.pi/(2*n_bins))
eta_bins = eta_bins/np.max(eta_bins)

var_phi_bins = np.linspace(0, 1, n_bins + 1)

# Relative frequency for each bin in each simulation:
f_eta_i = np.empty([n_sim, len(eta_bins) - 1]) # Empty list.
f_var_phi_i = np.empty([n_sim, len(var_phi_bins) - 1])
for i in range(n_sim):
    # Normalized histograms.
    hst_eta = np.histogram(eta[i, :], eta_bins, density = test_type)
    hst_var_phi = np.histogram(var_phi[i, :], var_phi_bins, \
                              density = test_type)

    # Relative frequency:
    f_eta_i[i] = hst_eta[0] # Frequency in each bin.
    f_var_phi_i[i] = hst_var_phi[0]

f_eta_i_mean = np.mean(f_eta_i, axis = 0) # Mean frequency.
f_var_phi_i_mean = np.mean(f_var_phi_i, axis = 0)

nu_eta_i = f_eta_i - f_eta_i_mean
nu_var_phi_i = f_var_phi_i - f_var_phi_i_mean

# I don't have to use all bins, because we have an compact interval.
# We can choose any collum to suppress in our calculations.
# Let's don't use the last bin for example.

nu_eta_i = nu_eta_i[:, 0:int(n_bins - 1)]
nu_var_phi_i = nu_var_phi_i[:, 0:int(n_bins - 1)]

```

```

if n_bins == 2:
    # Variance:
    m_ij_eta = np.var(nu_eta_i)
    m_ij_var_phi = np.var(nu_var_phi_i)

    # Inverse of variance:
    minv_ij_eta = 1/m_ij_eta
    minv_ij_var_phi = 1/m_ij_var_phi

    # Chi-Squared test:
    chi2_eta = np.array([np.dot(nu_eta_i[j], \
                                np.dot(minv_ij_eta, nu_eta_i[j]))/\
                                (n_bins - 1) for j in range(n_sim)])
    chi2_var_phi = np.array([np.dot(nu_var_phi_i[j], \
                                    np.dot(minv_ij_var_phi, \
                                             nu_var_phi_i[j]))/\
                                    (n_bins - 1) for j in range(n_sim)])
    return np.array([chi2_eta, chi2_var_phi, m_ij_eta, \
                    m_ij_var_phi, f_eta_i_mean, f_var_phi_i_mean])

else:
    # Covariance matrix:
    m_ij_eta = np.cov(nu_eta_i, rowvar = False)
    m_ij_var_phi = np.cov(nu_var_phi_i, rowvar = False)

    # Inverse of covariance matrix:
    minv_ij_eta = np.linalg.inv(m_ij_eta)
    minv_ij_var_phi = np.linalg.inv(m_ij_var_phi)

    # This is one additional check of matrix inversion:
    m_multpl_eta = np.dot(minv_ij_eta, m_ij_eta)
    m_multpl_var_phi = np.dot(minv_ij_var_phi, m_ij_var_phi)
    diag = np.zeros((n_bins - 1, n_bins - 1))
    i, j = np.indices(diag.shape)
    diag[i == j] = 1

    if (np.allclose(m_multpl_eta, diag) == True) and \
        (np.allclose(m_multpl_var_phi, diag) == True):
        chi2_eta = np.array([np.dot(nu_eta_i[j], \

```

---

```

        np.dot(minv_ij_eta, nu_eta_i[j]))/(n_bins - 1) \
        for j in range(n_sim)])
    chi2_var_phi = np.array([np.dot(nu_var_phi_i[j], \
        np.dot(minv_ij_var_phi, nu_var_phi_i[j]))\
        /(n_bins - 1) for j in range(n_sim)])
    return np.array([chi2_eta, chi2_var_phi, m_ij_eta, \
        m_ij_var_phi, f_eta_i_mean, \
        f_var_phi_i_mean])

def chi2_test(theta_test, phi_test, eta_cov, var_phi_cov, \
    eta_mean_test, var_phi_mean_test, \
    n_sim, n_bins, test_type, off_diag):
    """
    Evaluates the reduced chi-squared test with parameters given a data.

    Args:
        theta_test (float array): array that contains thetas.
        phi_test (float array): array that contains phis.
        eta_cov (float array): covariance matrix for thetas of a test.
        var_phi_cov (float array): covariance matrix for phis of a test.
        eta_mean_test (float array): mean values for eta from theory.
        var_phi_mean_test (float array): mean values for var_phi
            from theory.
        n_sim (int): number of simulations.
        n_bins (int): number of bins.
        test_type (bool): if True, it returns a normality chi-squared
            test. If False, it returns a test for
            poisson data.
        off_diag (bool): if True, it uses off diagonal terms cov matrix.
            If False, it uses only terms from diag
            of cov matrix.

    Returns:
        An array containing reduced chi-squared values for
        [theta, phi, masked_theta, masked_phi] for sim simulations.
    """
    # New variables:
    eta_test = 1 - np.cos(theta_test)
    var_phi_test = phi_test/(2*np.pi)

```

```

var_phi_test[np.where(var_phi_test < 0)] = var_phi_test[np.where\
                                                    (var_phi_test < 0)] + 1

# Bins Edges:
# Since the mask we aim to apply on our data have a
# semi-pseudo azimuthal symmetry, we will bin theta in an uniform
# way leading that eta is not unniiform (bins with different width).
theta_bins = np.histogram(theta_test, bins=n_bins, \
                           range=(0,np.pi/2),density=1)[1]
eta_bins = np.cumsum(np.sin(theta_bins)*np.pi/(2*n_bins))
eta_bins = eta_bins/np.max(eta_bins)

var_phi_bins = np.linspace(0, 1, n_bins + 1)

# Relative frequency for each bin in each simulation:
f_eta_test_i = np.empty([n_sim, len(eta_bins) - 1]) # Empty list.
f_var_phi_test_i = np.empty([n_sim, len(var_phi_bins) - 1])
for i in range(n_sim):
    # Normalized histograms.
    hst_eta_test = np.histogram(eta_test[i, :], eta_bins, \
                                density = test_type)
    hst_var_phi_test = np.histogram(var_phi_test[i, :], \
                                    var_phi_bins, \
                                    density = test_type)

    # Relative frequency:
    f_eta_test_i[i] = hst_eta_test[0] # Frequency in each bin.
    f_var_phi_test_i[i] = hst_var_phi_test[0]

nu_eta_test_i = f_eta_test_i - eta_mean_test
nu_var_phi_test_i = f_var_phi_test_i - var_phi_mean_test
nu_eta_test_i = nu_eta_test_i[:, 0:int(n_bins - 1)]
nu_var_phi_test_i = nu_var_phi_test_i[:, 0:int(n_bins - 1)]

if n_bins == 2:
    # Inverse of variance:
    eta_inv_cov = 1/eta_cov
    var_phi_inv_cov = 1/var_phi_cov
else:

```

---

```

    if off_diag == True:
        # Inverse of covariance matrix:
        eta_inv_cov = np.linalg.inv(eta_cov)
        var_phi_inv_cov = np.linalg.inv(var_phi_cov)
    else:
        eta_inv_cov = np.linalg.inv(np.diag(np.diag(eta_cov)))
        var_phi_inv_cov = np.linalg.inv(np.diag(\
                                            np.diag(var_phi_cov)))

chi2_eta_test = np.array([np.dot(nu_eta_test_i[j], \
                                np.dot(eta_inv_cov, nu_eta_test_i[j]))/(n_bins - 1)\
                            for j in range(n_sim)])
chi2_var_phi_test = np.array([np.dot(nu_var_phi_test_i[j], \
                                      np.dot(var_phi_inv_cov, nu_var_phi_test_i[j]))\
                              /(n_bins - 1) for j in range(n_sim)])
return np.array([chi2_eta_test, chi2_var_phi_test])

def many_chi2_theory_l_bins(mvs_folder, files_out, lmin, lmax):
    """
    Calculate Chi-Squared Test for Theory. That means it will save the
    frequency and covariance matrix that is used as a template.

    Args:
        mvs_folder (string): folder that contains multipole vectors.
        files_out (string): name of output files.
        lmin (int): minimum multipole.
        lmax (int): maximum multipole.

    Returns:
        Saves the output for "chi2_theory" function for several
        multipoles and simulations.
    """
    for l in range(lmin, lmax + 1):
        mvs = np.empty((3000, 1, 2))
        delete_wrong_mvs = []
        for i in range(1, 3000 + 1):
            mvs_i = np.load(mvs_folder.format(i, i, 1))
            mvs_select = mvs_i[mvs_i[:, 0] < np.pi/2]
            if mvs_select[:, 0].shape[0] == 1:

```

```

        mvs[i - 1] = mvs_select
    else:
        delete_wrong_mvs.append(i - 1)
updated_mvs = np.delete(mvs, delete_wrong_mvs, axis=0)

del mvs, delete_wrong_mvs

n_sim = updated_mvs.shape[0]
if l <= 30:
    n_bins = 1
else:
    n_bins = int(np.ceil(570*(l - 30)/1470 + 30))
    # This is necessary because for l = 1500 we would
    # have 600 bins.
chi_squared = chi2_theory(updated_mvs[:, :, 0], \
                          updated_mvs[:, :, 1], \
                          n_sim, n_bins, True)
np.save(files_out.format(l, n_bins), chi_squared)

def many_chi2_test_l_bins(mvs_folder, files_out, \
                          files_theory, lmin, lmax, off_diag):
    """
    Calculate Chi-Squared based in a theory estimation.
    It will use frequency and covariance matrix
    previously calculated and apply to new simulations.

    Args:
        mvs_folder (string): folder that contains multipole vectors.
        files_out (string): name of output files.
        files_theory (string): name of files that contains estimation.
        lmin (int): minimum multipole.
        lmax (int): maximum multipole.
        off_diag (bool): If True use all terms of covariance matrix.
                        If False, it will use only diag terms of
                        cov matrix.

    Returns:
        Saves the output for "chi2_test" function of several multipoles.
    """

```

---

```

for l in range(lmin, lmax + 1):
    mvs = np.empty((1000, 1, 2))
    delete_wrong_mvs = []
    for i in range(3001, 4000 + 1):
        mvs_i = np.load(mvs_folder.format(i, i, 1))
        mvs_select = mvs_i[mvs_i[:, 0] < np.pi/2]
        if mvs_select[:, 0].shape[0] == 1:
            mvs[i - 3001] = mvs_select
        else:
            delete_wrong_mvs.append(i - 1 - 3000)
    updated_mvs = np.delete(mvs, delete_wrong_mvs, axis=0)

del mvs, delete_wrong_mvs

n_sim = updated_mvs.shape[0]
if l <= 30:
    n_bins = 1
else:
    n_bins = int(np.ceil(570*(1 - 30)/1470 + 30))
theory = np.load(files_theory.format(l, n_bins))
chi_squared = chi2_test(updated_mvs[:, :, 0], \
                        updated_mvs[:, :, 1], \
                        theory[2], theory[3], \
                        theory[4], theory[5], \
                        n_sim, n_bins, True, off_diag)
np.save(files_out.format(l, n_bins), chi_squared)

del chi_squared, theory

def many_chi2_test_l_bins_real(mvs_file, files_out, \
                               files_theory, lmin, lmax, off_diag):
    """
    Calculate Chi-Squared based in a theory estimation. It will use
    frequency and covariance matrix previously calculated and apply
    to a single map.

    Args:
        mvs_file (float array): array containing all multipole vectors.
        files_out (string): name of output files.

```

*files\_theory (string): name of files that contains estimation.*  
*lmin (int): minimum multipole.*  
*lmax (int): maximum multipole.*  
*off\_diag (bool): If True use all terms of covariance matrix.*  
*If False, it will use only diag terms*  
*of cov matrix.*

*Returns:*

*Saves the output for "chi2\_test" function of several multipoles.*

"""

```

indexes_ = np.repeat(np.arange(2, 1500 + 1, 1), \
                    2*np.arange(2, 1500 + 1, 1))
mvs_file_open = np.load(mvs_file)
mpvs_ = np.vstack((indexes_, np.array(mvs_file_open).T)).T

for l in range(lmin, lmax + 1):
    mvs = np.empty((1, 1, 2))
    for i in range(1, 1 + 1):
        result_0 = mpvs_[mpvs_[ :, 0] == 1][ :, 1:]
        mvs[i - 1] = result_0[result_0[:, 0] < np.pi/2]
    if l <= 30:
        n_bins = 1
    else:
        n_bins = int(np.ceil(570*(l - 30)/1470 + 30))
    theory = np.load(files_theory.format(l, n_bins))
    chi_squared = chi2_test(mvs[:, :, 0], mvs[:, :, 1], \
                          theory[2], theory[3], theory[4], \
                          theory[5], 1, n_bins, True, off_diag)
    np.save(files_out.format(l, n_bins), chi_squared)

```