Bispectrum in BINGO Radiotelescope

Jordany Vieira - PhD Student at USP
Advisor - Elcio Abdalla

jordanyv@usp.br

11th Aegean Summer School Sep 2022
12/09

Outline

1. Bispectrum Team
2. Why 21cm and Bispectrum?
3. Bispectrum Module
4. BINGO Paper V
5. Conclusions
6. Actual Stage and Next Steps

Bispectrum Team

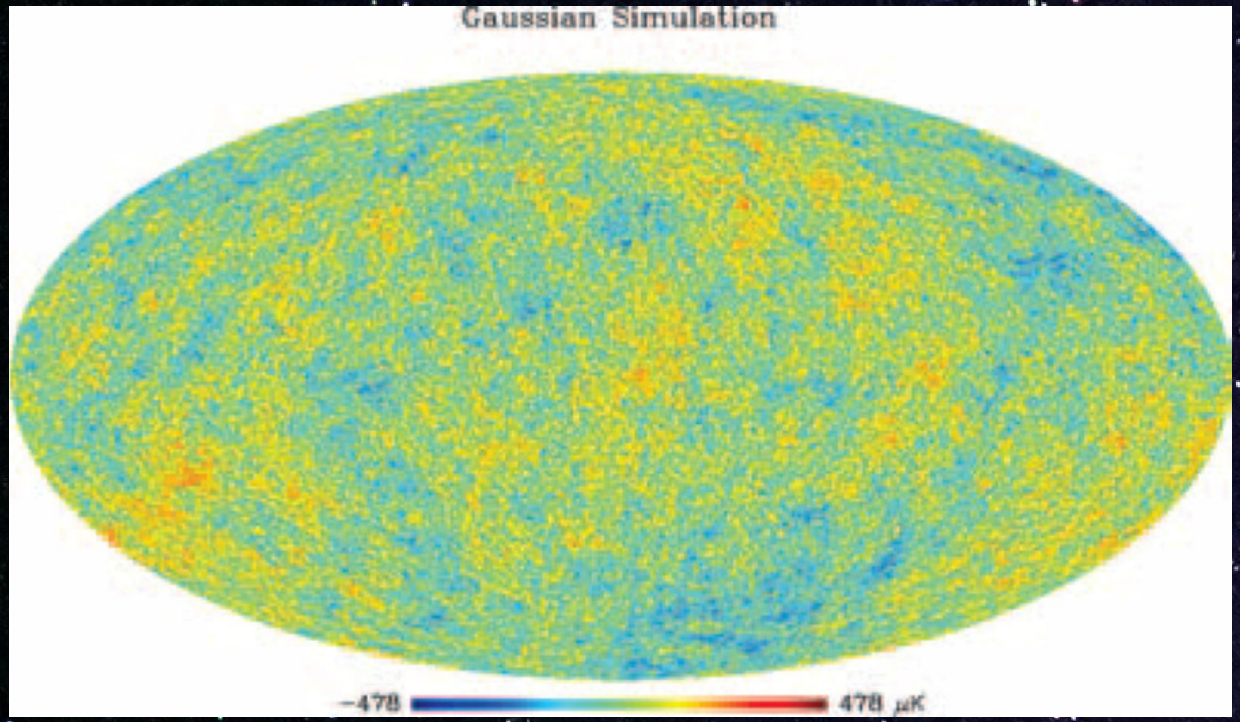
Jordany Vieira

Filipe Abdalla, Elcio Abdalla, Karin Fornazier, Alessandro Marins, João Alberto, Gabriel Hoerning, Ojaswi Jain and Amanda Santos.

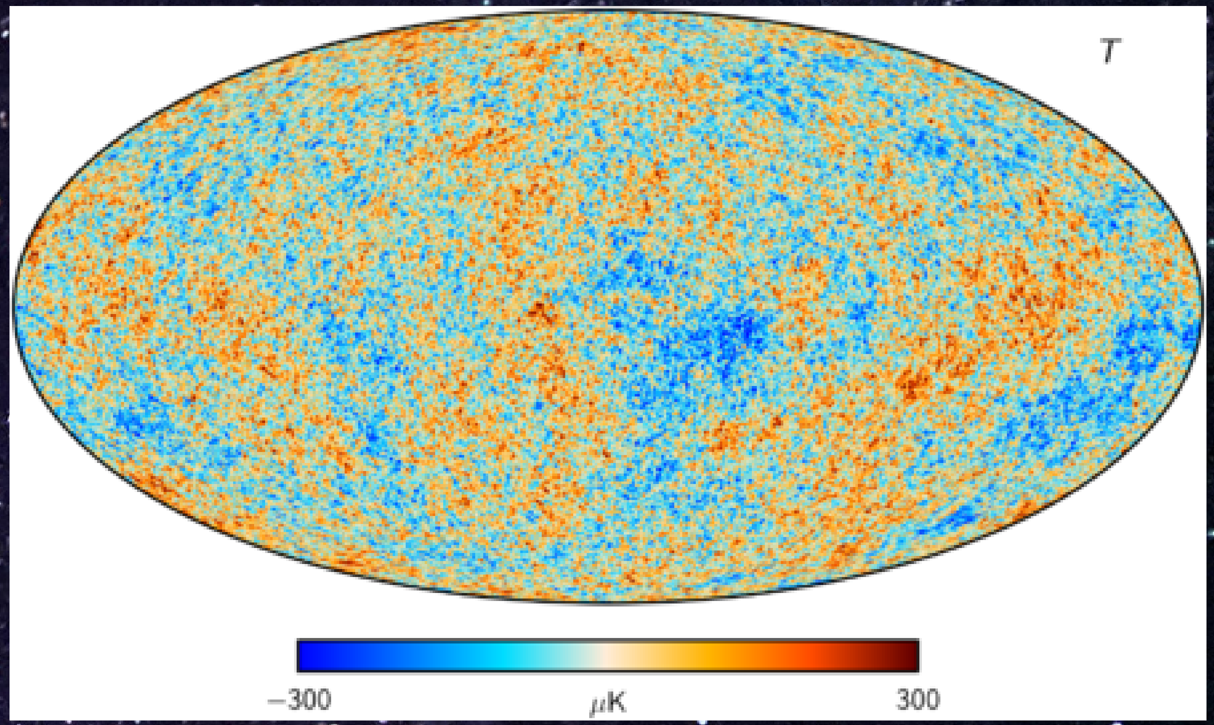
Why 21cm and Bispectrum?

- After decoupling and recombination, it becomes possible to observe the emission of 21-cm signals from neutral hydrogen (HI).
- There is HI in abundance in the Universe.
- Can be used to map the distribution of matter on large scales and its evolution in the radio band.
- Search for traces of non-Gaussianities in 21-cm fields associated with the inflationary period
- Find the explanation of the fluctuations in the matter power spectrum.

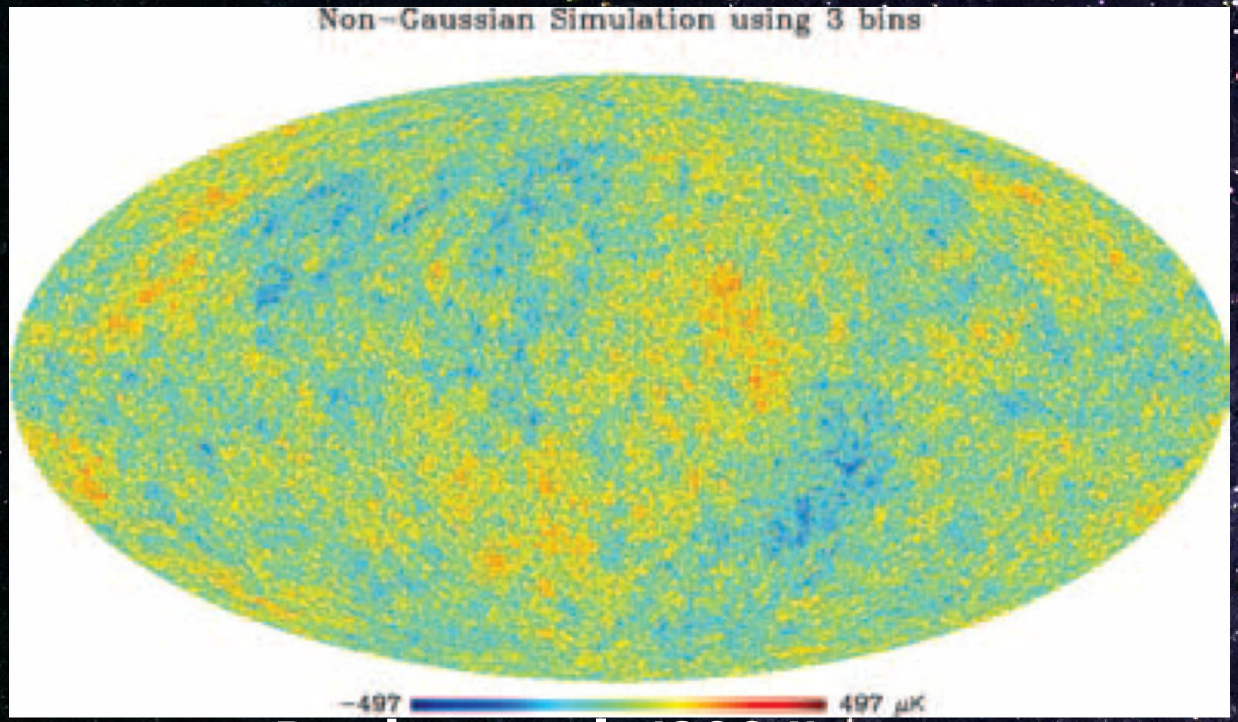
Gaussian



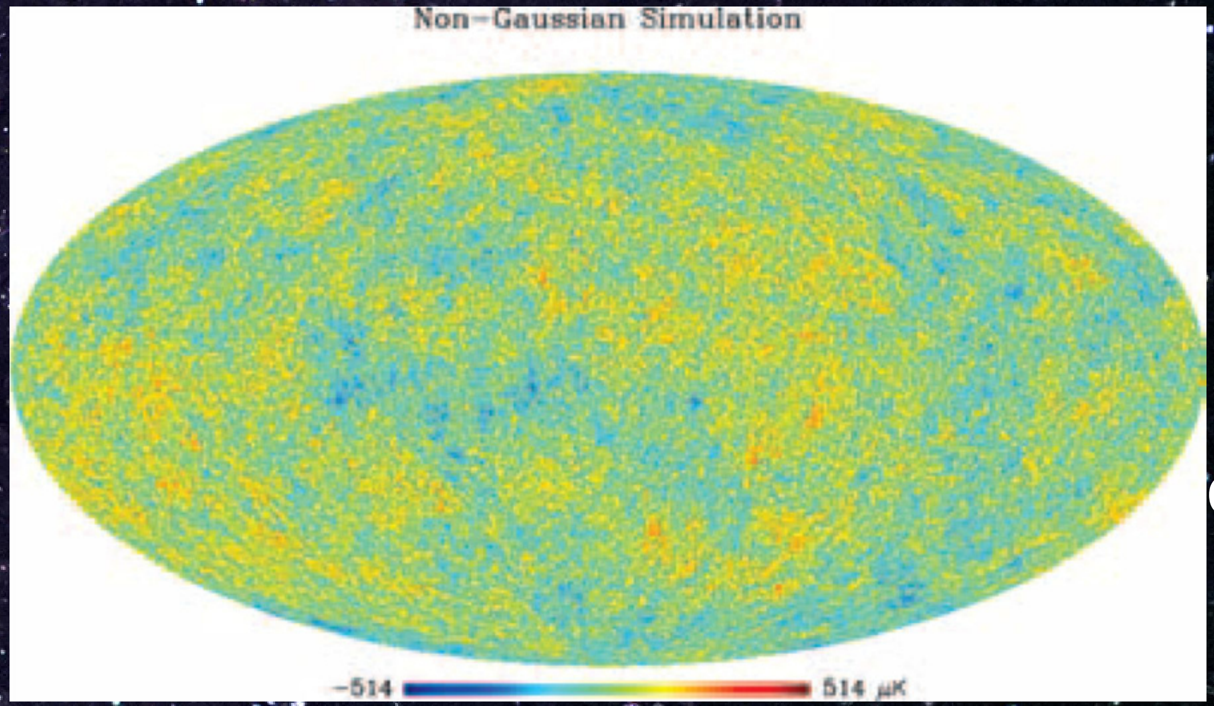
Rocha et al. (2004)



Planck Collaboration 2020, paper IV



Rocha et al. (2004)



G. Rocha et al. (2004)

Non-Gaussian

Bispectrum Module

$$B_{\ell_1\ell_2\ell_3} = \int d\Omega M_{\ell_1}(\Omega) M_{\ell_2}(\Omega) M_{\ell_3}(\Omega)$$

Bispectrum from sky maps

$$B_{\ell_1\ell_2\ell_3} = \int d\Omega \sum_{m_1,m_2,m_3} Y_{\ell_1 m_1}(\Omega) Y_{\ell_2 m_2}(\Omega) Y_{\ell_3 m_3}(\Omega) a_{\ell_1 m_1} a_{\ell_2 m_2} a_{\ell_3 m_3}$$

After spherical harmonic transformations

$$\int d\Omega Y_{\ell_1 m_1}(\Omega) Y_{\ell_2 m_2}(\Omega) Y_{\ell_3 m_3}(\Omega) = \sqrt{N_{\Delta}^{\ell_1\ell_2\ell_3}} \begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ m_1 & m_2 & m_3 \end{pmatrix}$$

Gaunt Integral

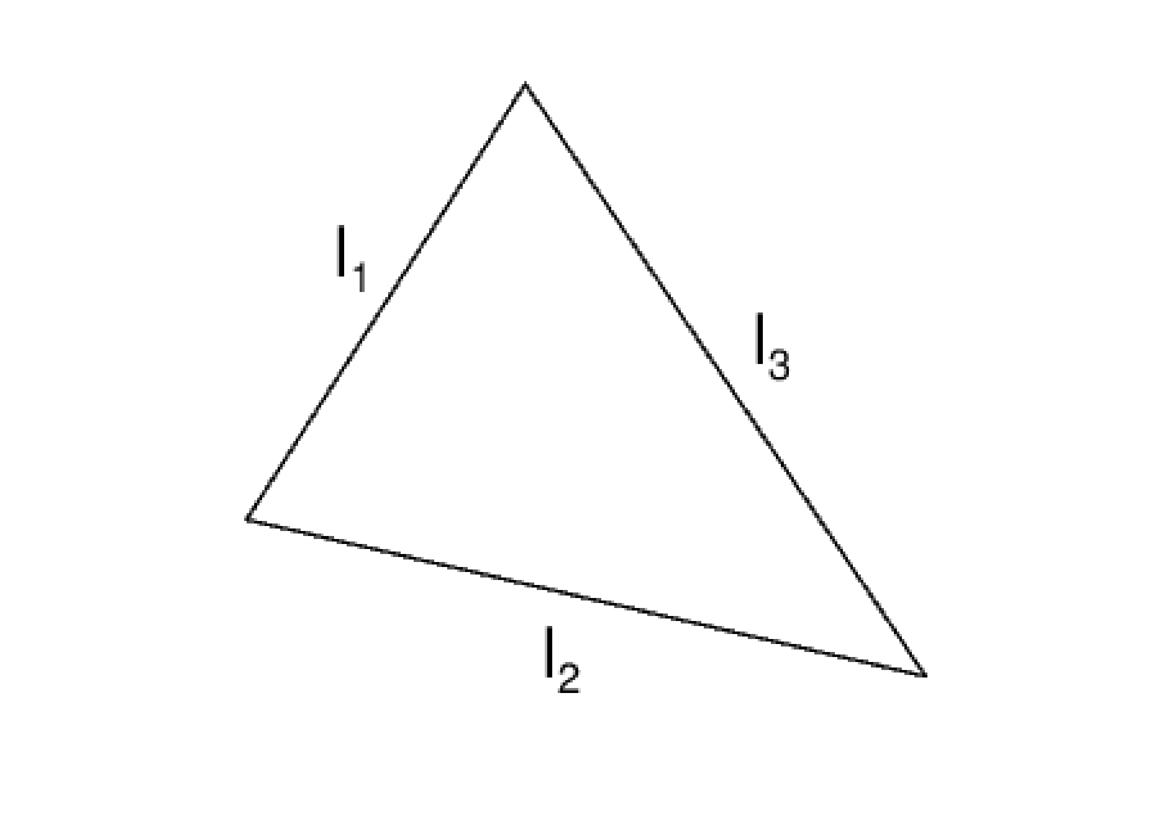
$$B_{\ell_1\ell_2\ell_3} = \sqrt{N_{\Delta}^{\ell_1\ell_2\ell_3}} \sum_{m_1,m_2,m_3} \begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ m_1 & m_2 & m_3 \end{pmatrix} a_{\ell_1 m_1} a_{\ell_2 m_2} a_{\ell_3 m_3}$$

$$N_{\Delta}^{\ell_1\ell_2\ell_3} \equiv \frac{(2\ell_1+1)+(2\ell_2+1)+(2\ell_3+1)}{4\pi} \begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ 0 & 0 & 0 \end{pmatrix}$$

Bispectrum to be calculated

M. Bucher et al. (2016)

Bispectrum triangle



Bispectrum Rules

$$|\ell_1 - \ell_2| \leq \ell_3 \leq \ell_1 + \ell_2$$

triangle inequality

$\ell_1 + \ell_2 + \ell_3$ even

parity invariance

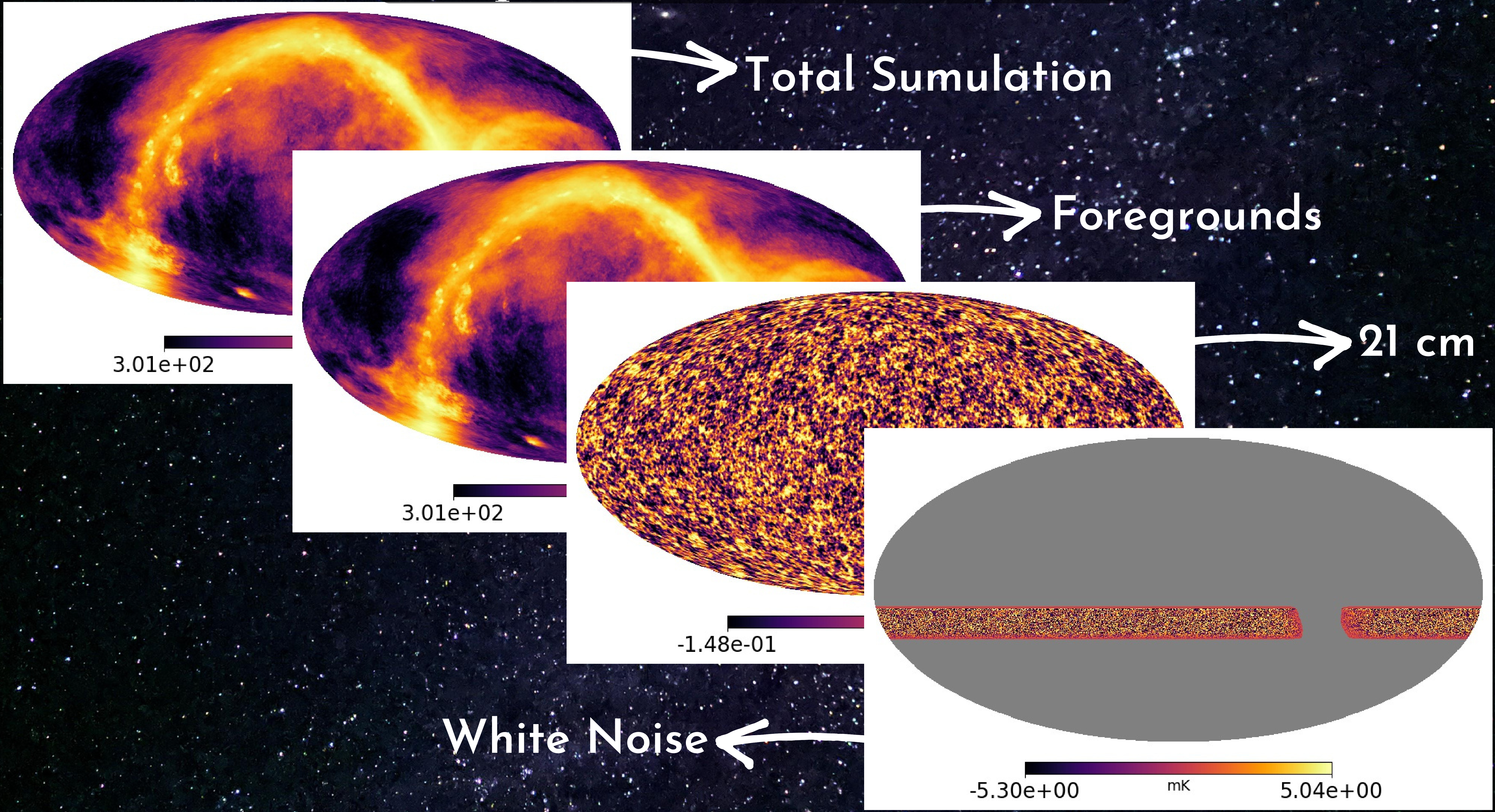
$$m_1 + m_2 + m_3 = 0$$

rotation invariance

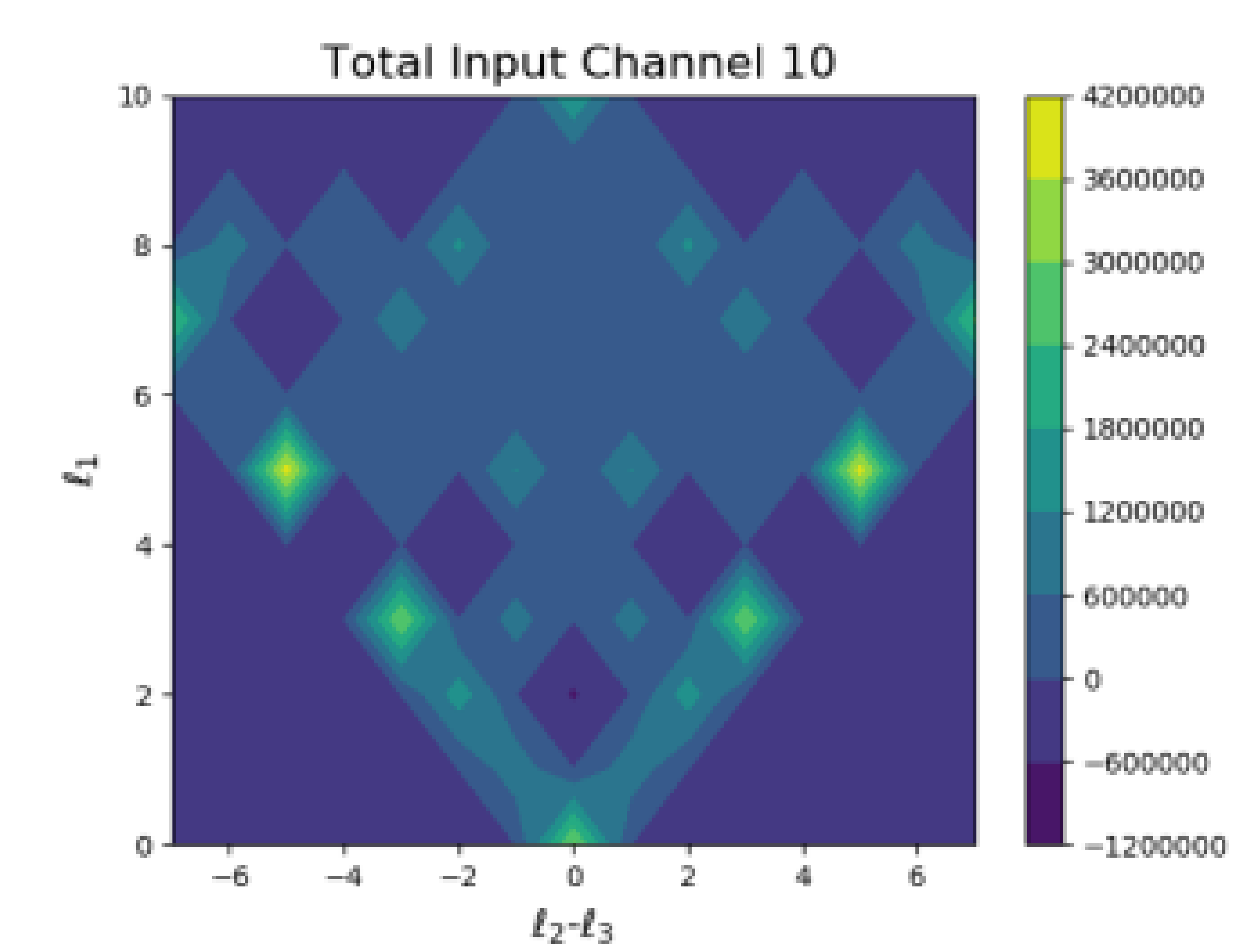
BINGO Paper V

- In our paper, we present two studies. Component Separation analysis and Bispectrum analysis.
- We use a FLASK code to simulate a 21cm map (H. S. Xavier et al. 2016) and PSM code for foreground components simulation (Delabrouille et al. 2013) and instrumental noise from BINGO paper II.
- Then we use the Bispectrum module to study these simulations before component separation by GNILC (Remazeilles et al. 2011 and Olivari et al. 2016) and after the component separation.

Components of the simulations



Equisize case



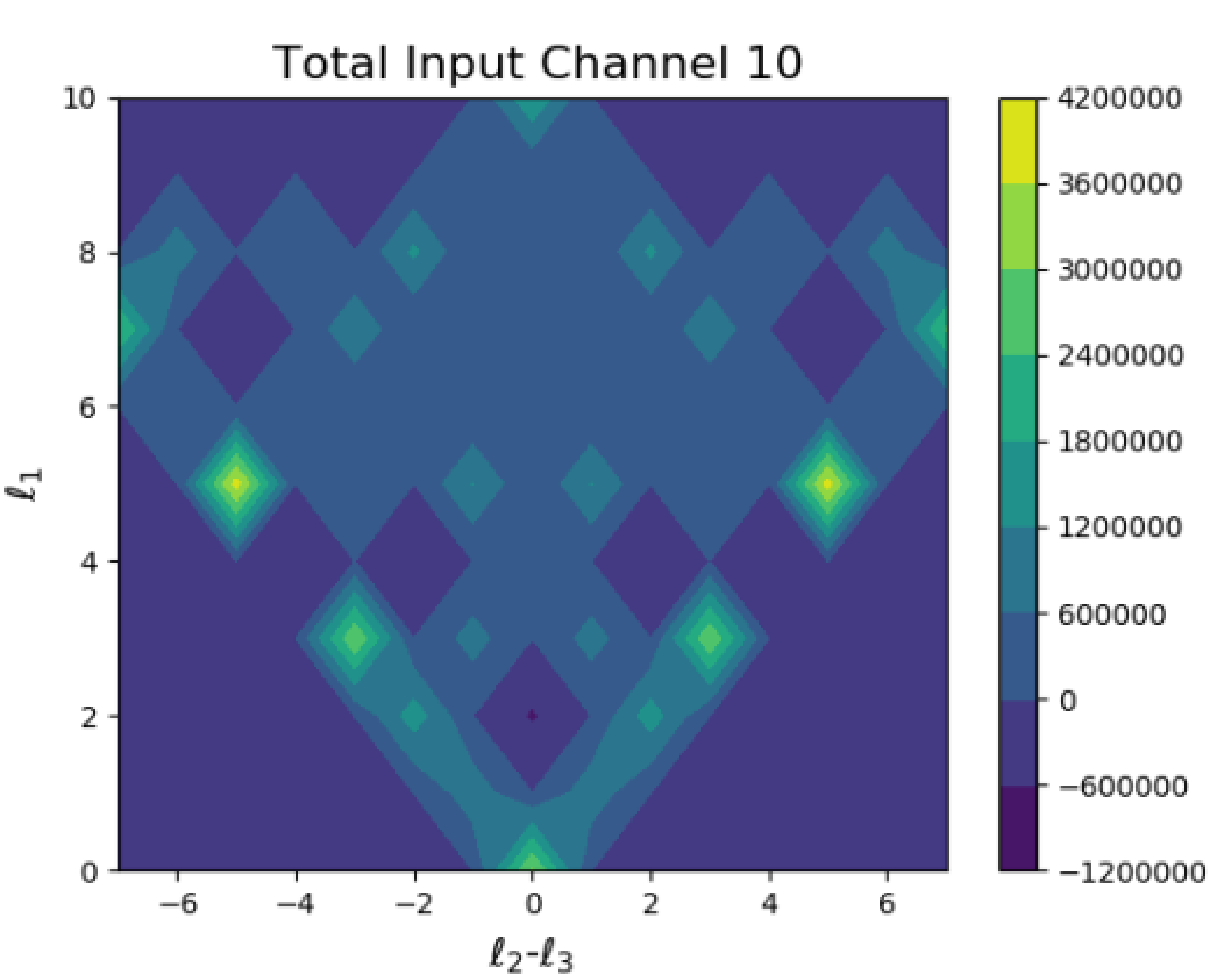
Equisize Rules

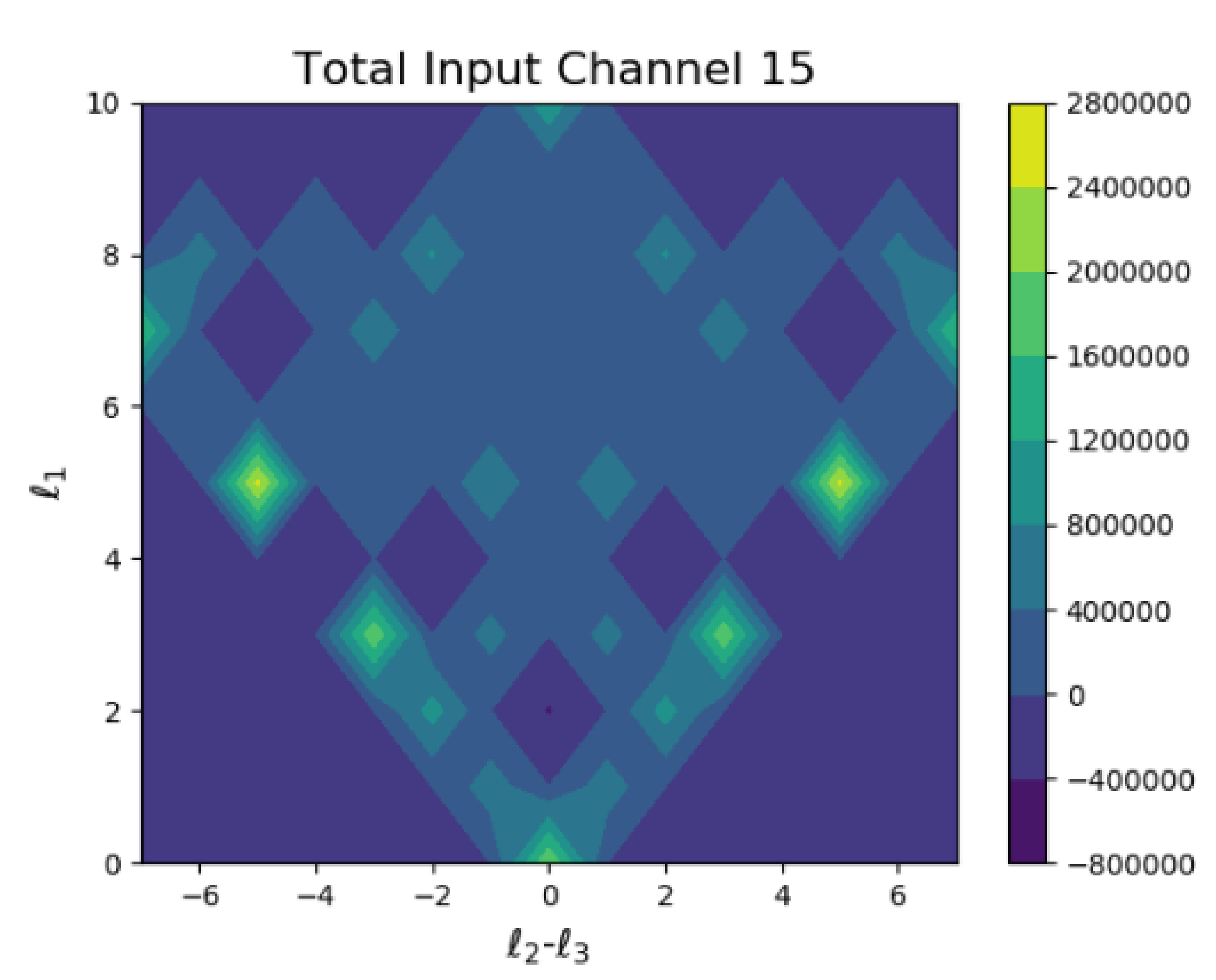
$$\ell_1 + \ell_2 + \ell_3 = \ell_0$$

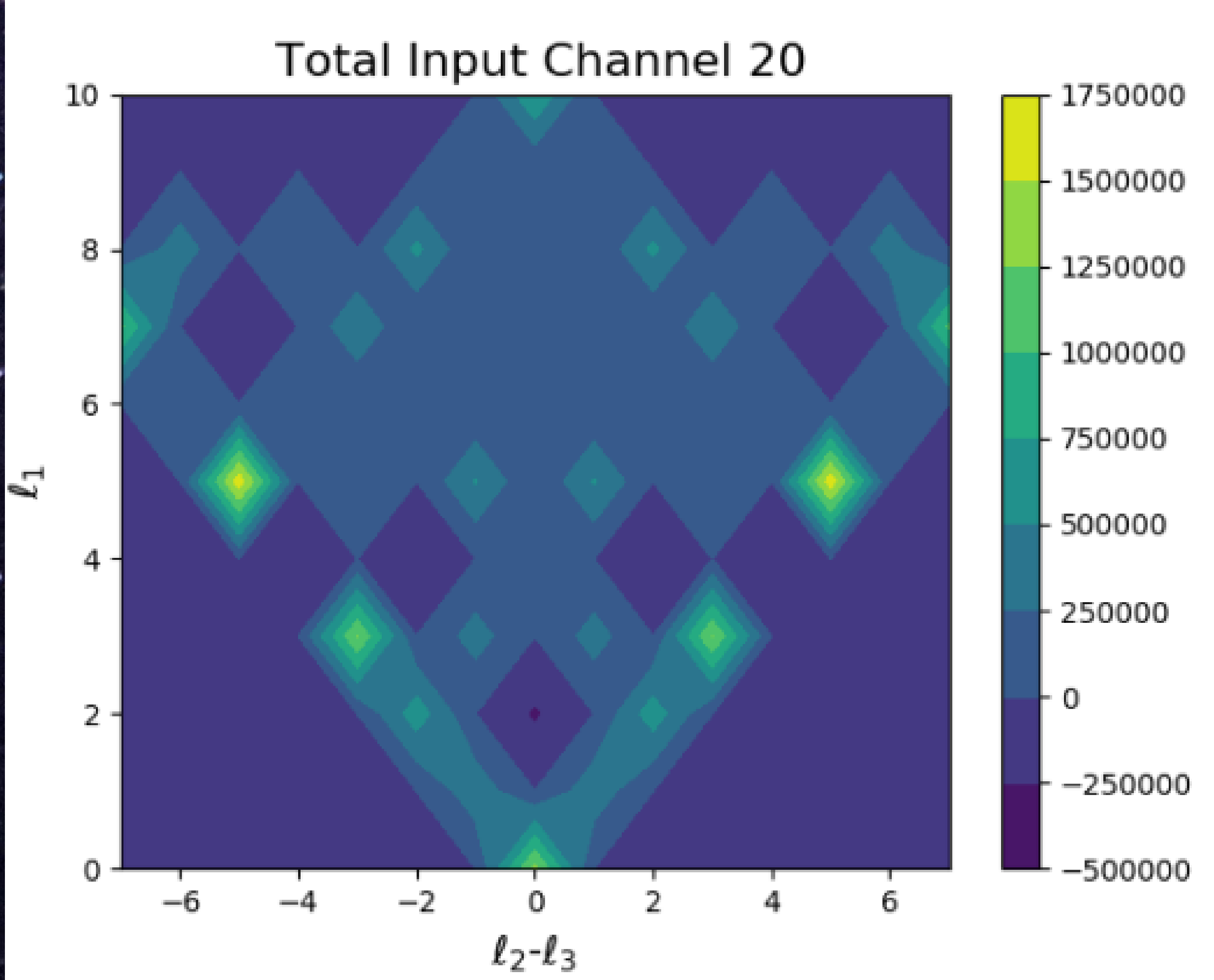
$$\ell_0 = 30$$

$$\ell_0 = 60$$

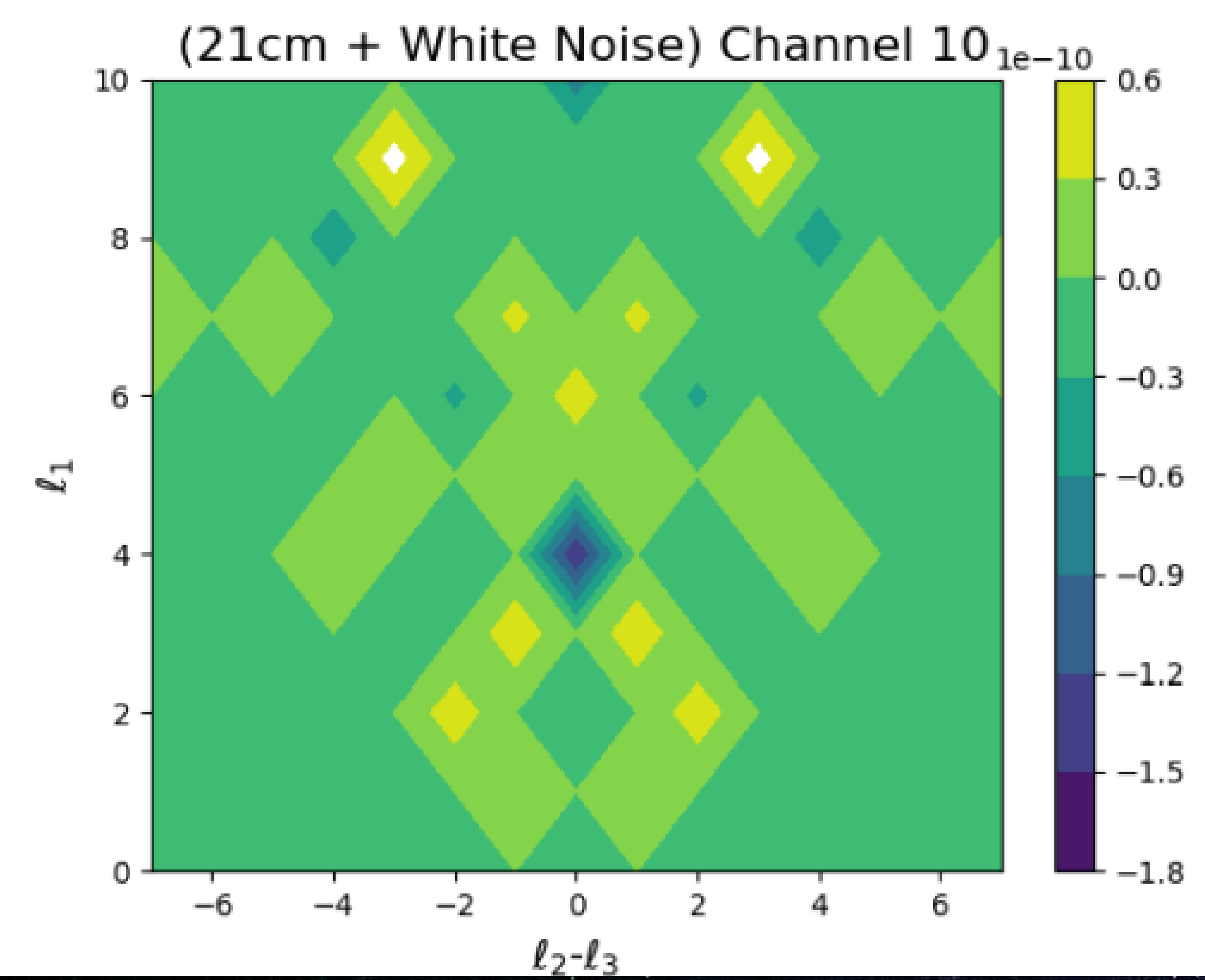
Results

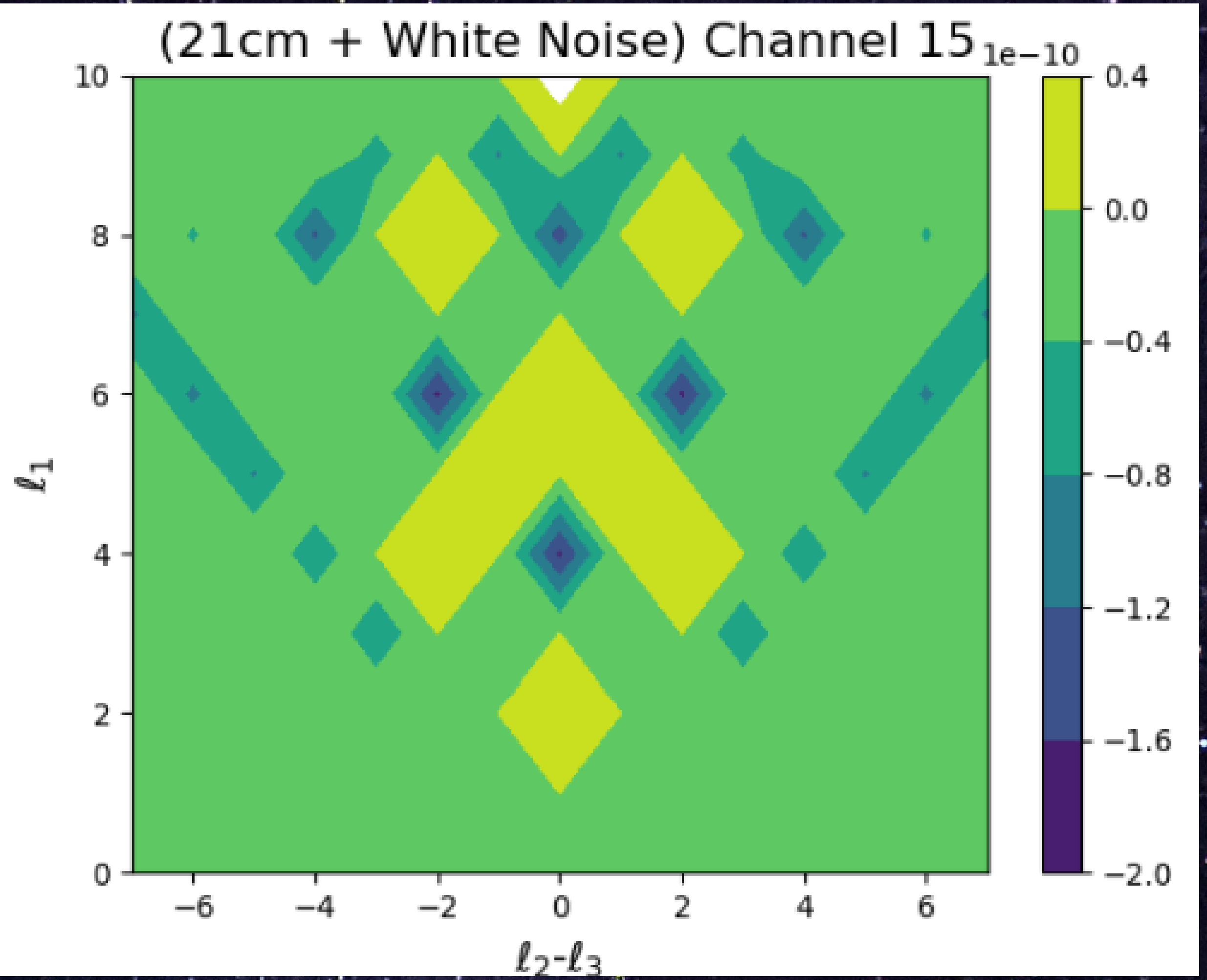

Total Input Channel 10
ℓ_1
ℓ_2-ℓ_3
4200000
3600000
3000000
2400000
1800000
1200000
600000
0
−600000
−1200000

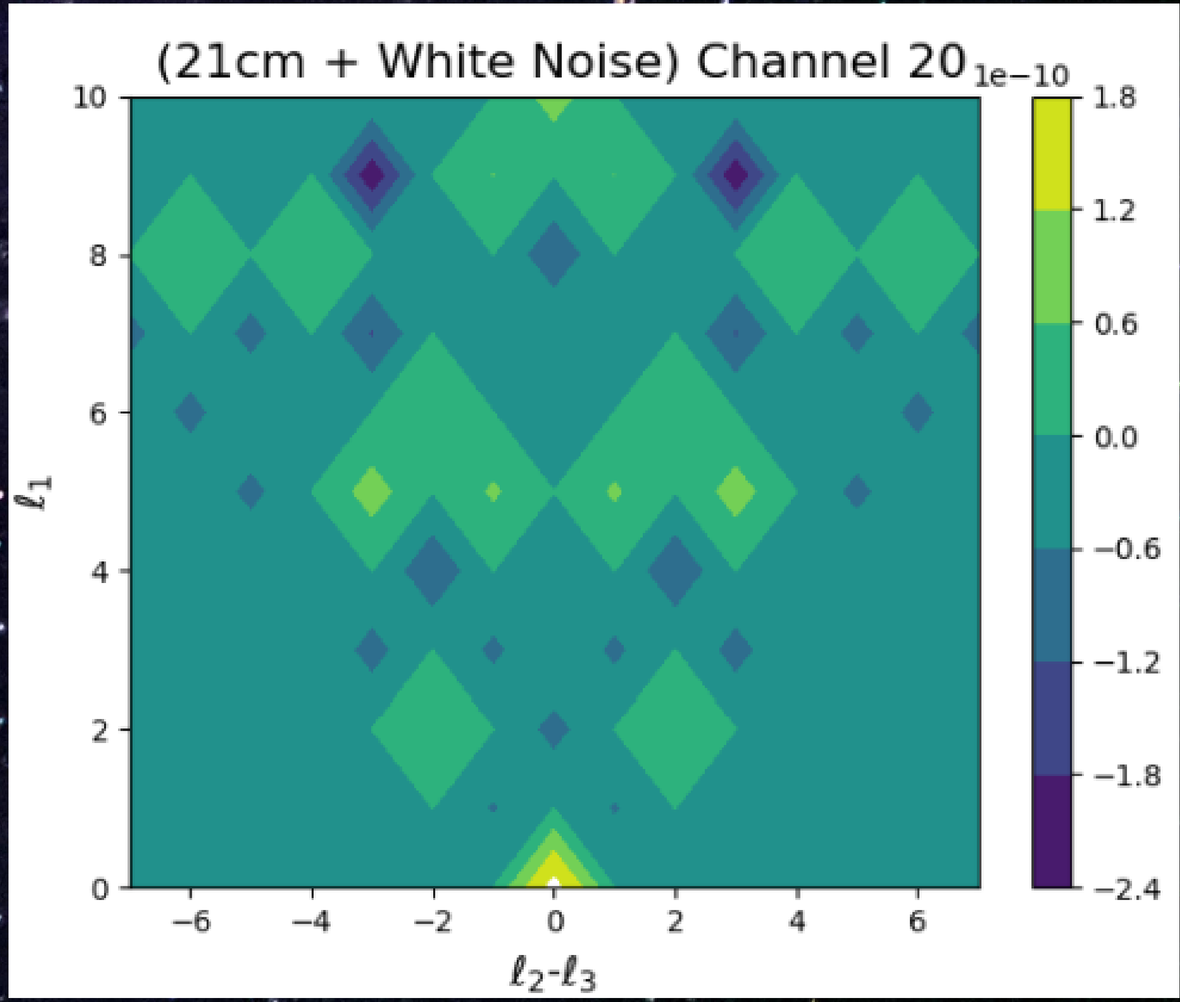

Total Input Channel 15
ℓ_1
ℓ_2-ℓ_3
2800000
2400000
2000000
1600000
1200000
800000
400000
0
−400000
−800000


Total Input Channel 20
ℓ_1
ℓ_2-ℓ_3
1750000
1500000
1250000
1000000
750000
500000
250000
0
−250000
−500000

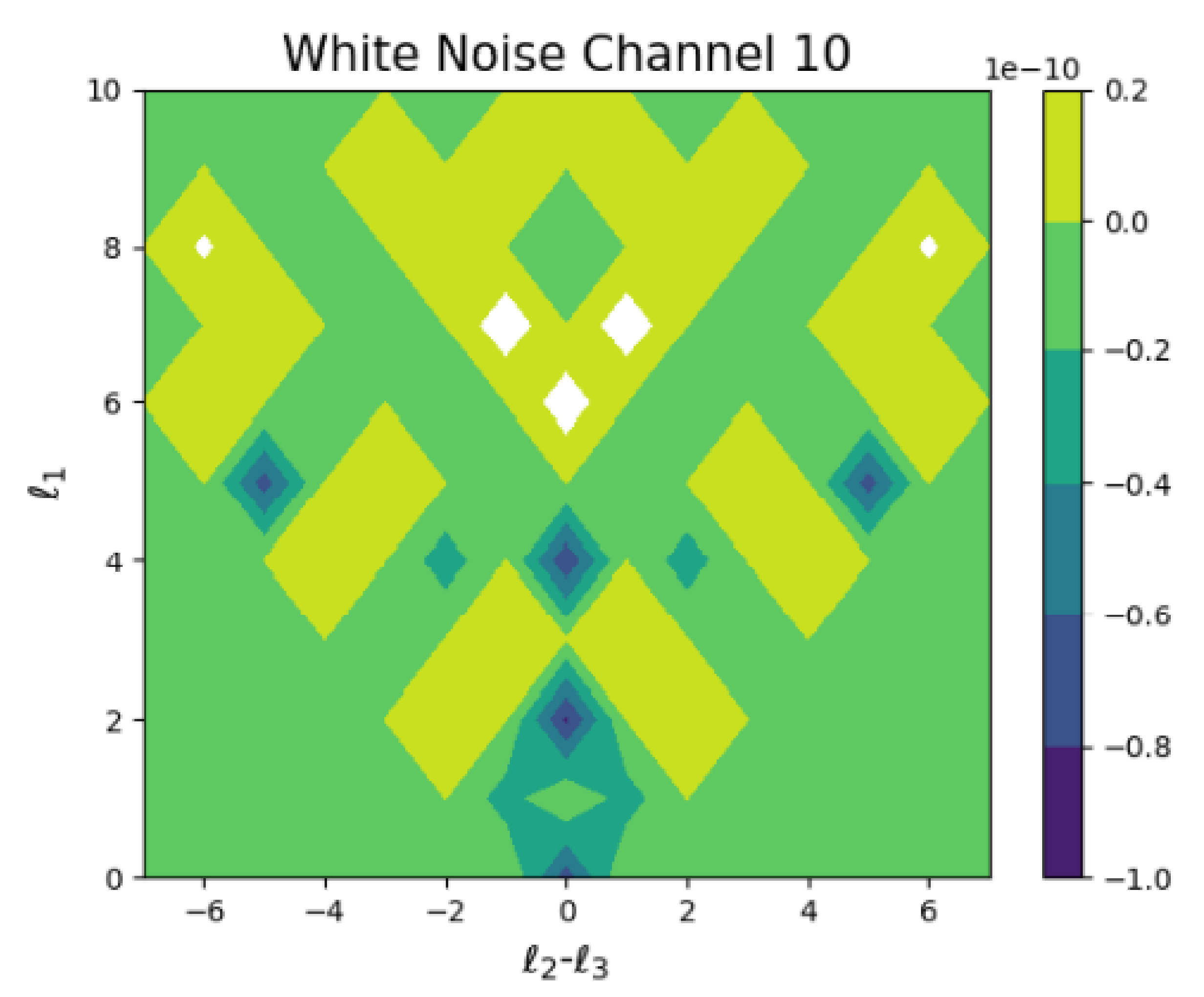
Results

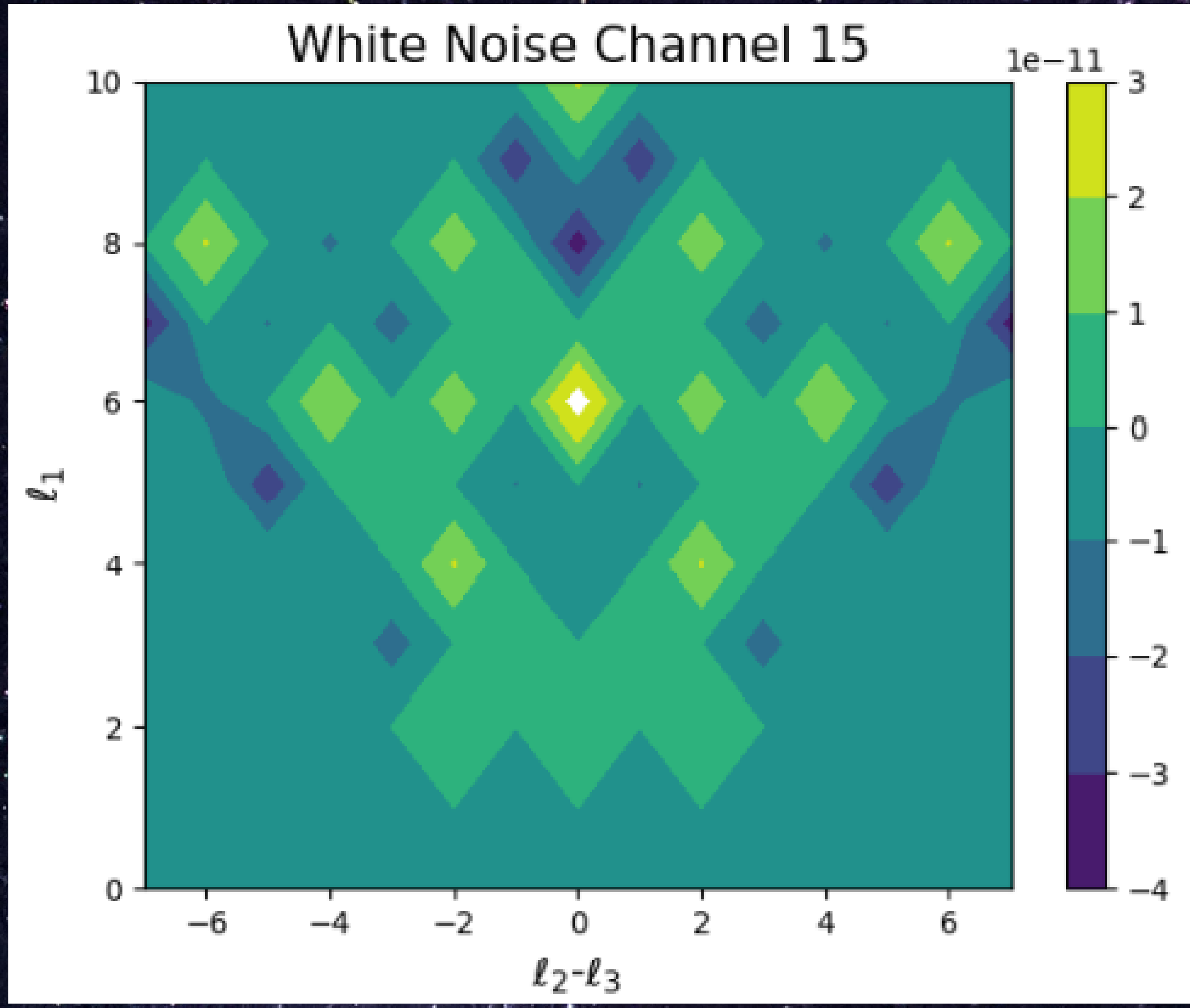

(21cm + White Noise) Channel 10
1e−10
ℓ_1
ℓ_2-ℓ_3
0.6
0.3
0.0
−0.3
−0.6
−0.9
−1.2
−1.5
−1.8

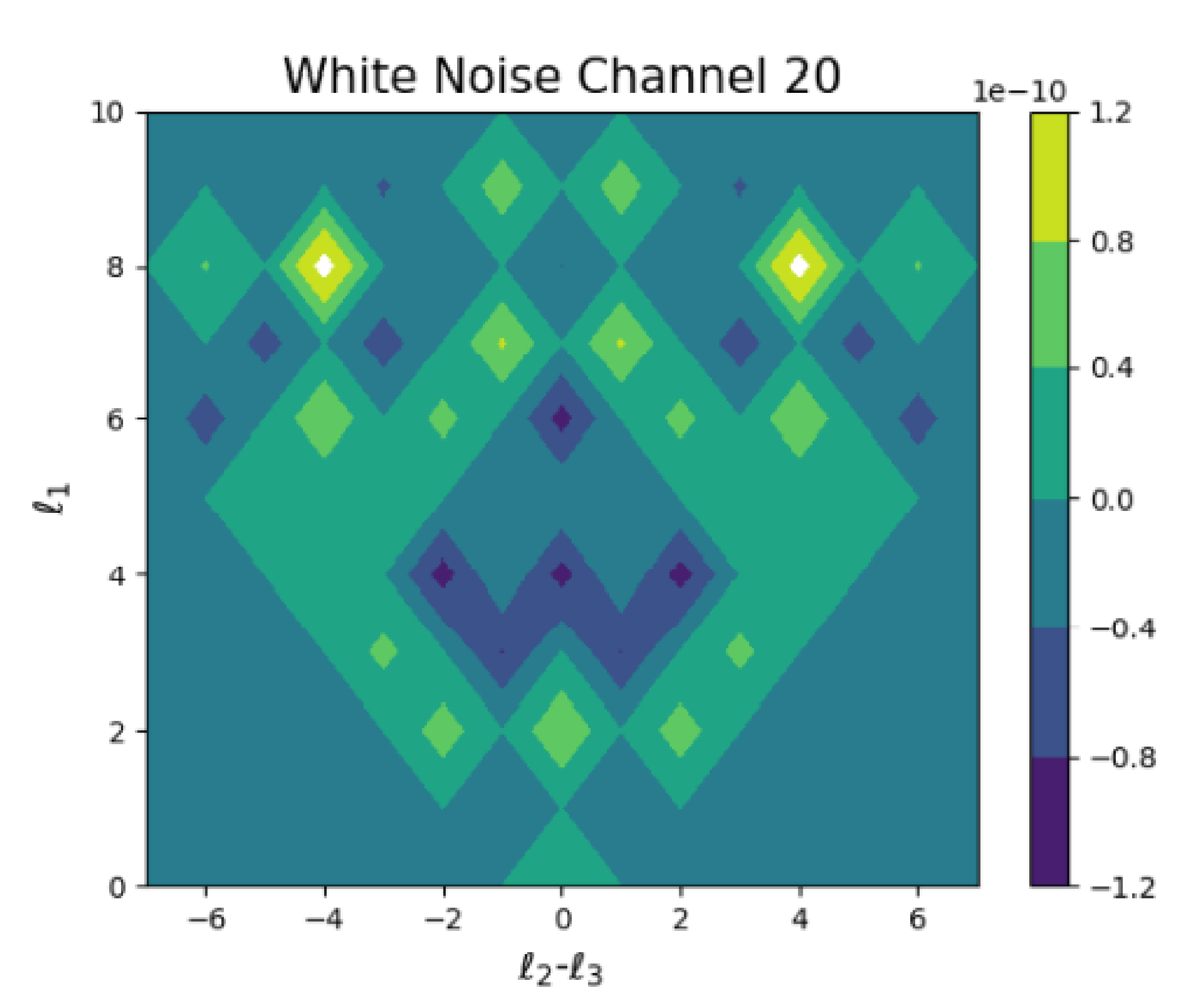

(21cm + White Noise) Channel 15
1e−10
ℓ_1
ℓ_2-ℓ_3
0.4
0.0
−0.4
−0.8
−1.2
−1.6
−2.0


(21cm + White Noise) Channel 20
1e−10
ℓ_1
ℓ_2-ℓ_3
1.8
1.2
0.6
0.0
−0.6
−1.2
−1.8
−2.4

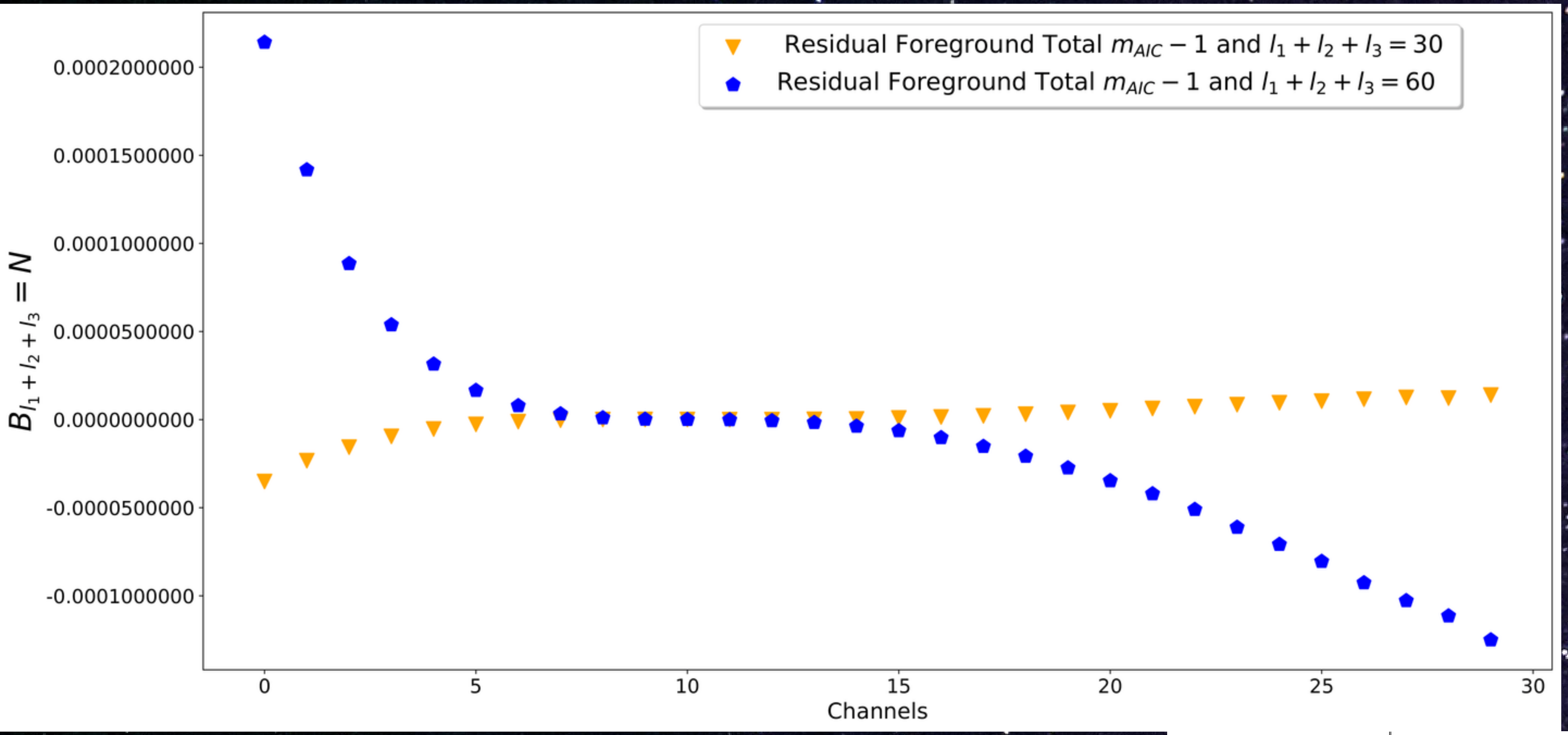
Results


White Noise Channel 10
1e−10
ℓ_1
ℓ_2-ℓ_3

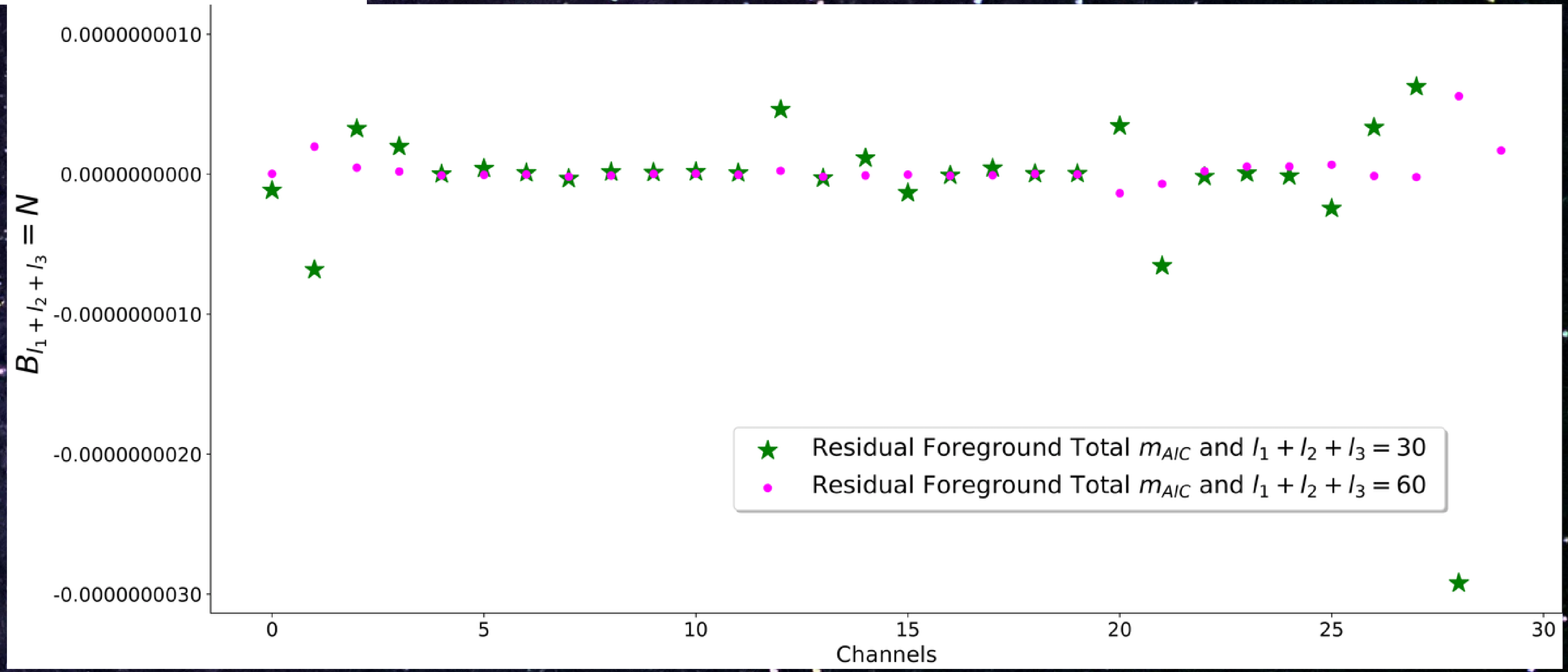

White Noise Channel 15
1e−11
ℓ_1
ℓ_2-ℓ_3


White Noise Channel 20
1e−10
ℓ_1
ℓ_2-ℓ_3

Bispectrum in residual Foregrounds using AIC



$$B_{\ell_1+\ell_2+\ell_3=N} = \sum_{\ell_1=1}^{N}\sum_{\ell_2=1}^{N}\sum_{\ell_3=1}^{N} B_{\ell_1\ell_2\ell_3}\delta_{\ell_1+\ell_2+\ell_3,N}$$



Conclusions

- Python calculation of the Gaunt Integral is slow, and it limits our multipole range (l).

- Our module can detect the residual foregrounds, then observe the non-Gaussianities.

- The Bispectrum module can be used in any sky intensity map to study non-Gaussianities.

Actual Stage and Next Steps

- Because of the limits in Python calculation of Gaunt integral, we choose two ways of solving the problem.

- The first way is the conversion of the bispectrum module from Python to C++ (CUDA) to improve the speed of calculation.

- The second way is to use the recursion of Wigner-3j symbols to calculate the Wigners more quickly and calculate the Gaunt more quickly.

Conversion of the Module

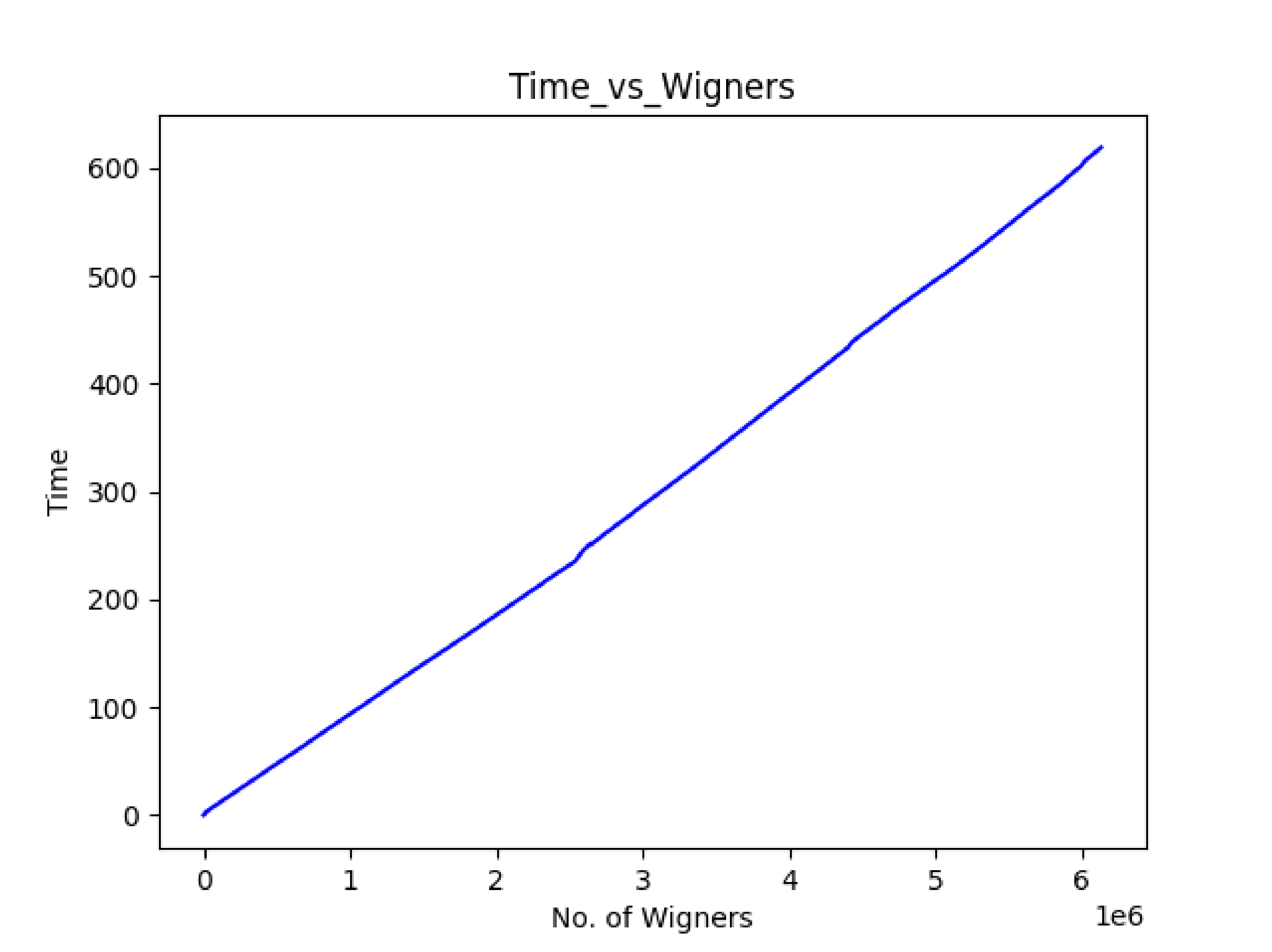
- Making the first conversion try, we find a big floating point error for l>70.

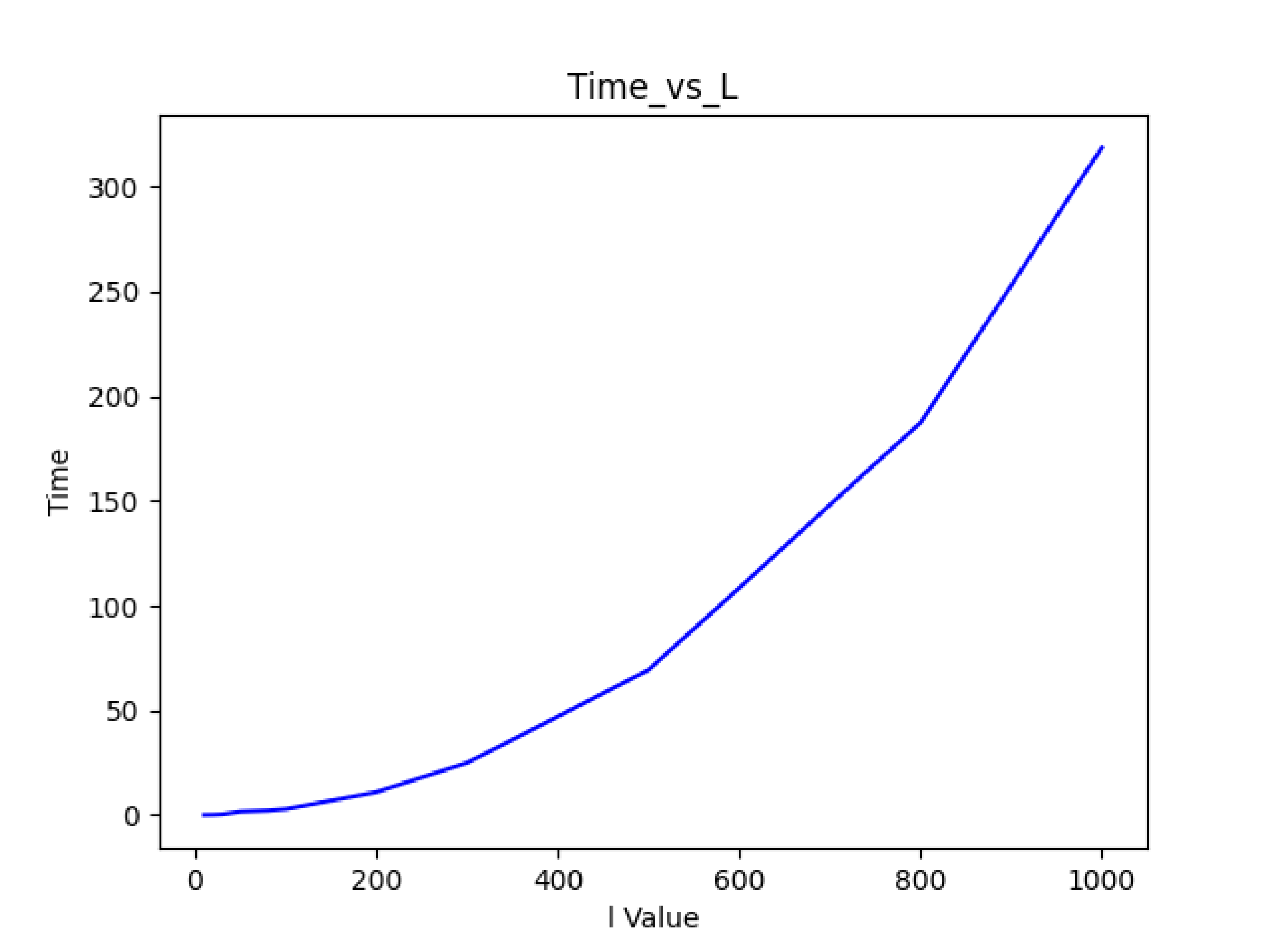
- To solve this issue, we will use prime numbers decomposition properties instead of the brute calculation of the Gaunts.

Wigner-3j recursion method

- Using the recursion of Wigner-3j symbols with bottom-up method, we find a high-speed calculation of the Wigners.

- Now we will try to make a calculation of the bispectrum using the Wigner-3j symbol.





IFUSP
Instituto de Física da USP

BINGO

Ευχαριστώ!

Thank You!

Obrigado!

Gaunt Integral equation

$$
\begin{aligned}
\mathrm{F3Y}(l_1, m_1, l_2, m_2, l_3, m_3) &= \Delta(l_1, l_2, l_3)\,\delta_{m_1+m_2+m_3,0}(-1)^{L+l_3+m_1-m_2} \\
&\times \left[\frac{(2l_1+1)(2l_2+1)(2l_3+1)}{4\pi}\right]^{1/2} \frac{L!}{(L-l_1)!\,(L-l_2)!\,(L-l_3)!} \\
&\times \left[\frac{(l_2-l_1+l_3)!\,(l_1-l_2+l_3)!\,(l_1+l_2-l_3)!}{(2L+1)!}\right]\left[(l_1-m_1)!\,(l_1+m_1)!\,(l_2-m_2)!\,(l_2+m_2)!\,(l_3-m_3)!\,(l_3+m_3)!\right]^{1/2} \\
&\times \sum_{\gamma=0}^{M} (-1)^{\gamma} \frac{1}{\gamma!(\gamma+l_3-l_1-m_2)!\,(l_2+m_2-\gamma)!\,(l_1-\gamma-m_1)!\,(\gamma+l_3-l_2+m_1)!\,(l_1+l_2-l_3-\gamma)!},
\end{aligned}
$$

Recursion Relation of Wigner-3j Symbols

$$-\sqrt{(l_3 \mp s_3)(l_3 \pm s_3 + 1)}\begin{pmatrix} l_1 & l_2 & l_3 \\ s_1 & s_2 & s_3 \pm 1 \end{pmatrix} =$$

$$= \sqrt{(l_1 \mp s_1)(l_1 \pm s_1 + 1)}\begin{pmatrix} l_1 & l_2 & l_3 \\ s_1 \pm 1 & s_2 & s_3 \end{pmatrix} + \sqrt{(l_2 \mp s_2)(l_2 \pm s_2 + 1)}\begin{pmatrix} l_1 & l_2 & l_3 \\ s_1 & s_2 \pm 1 & s_3 \end{pmatrix}$$