Tools for data analyses in Cosmology

- Aula 9 -

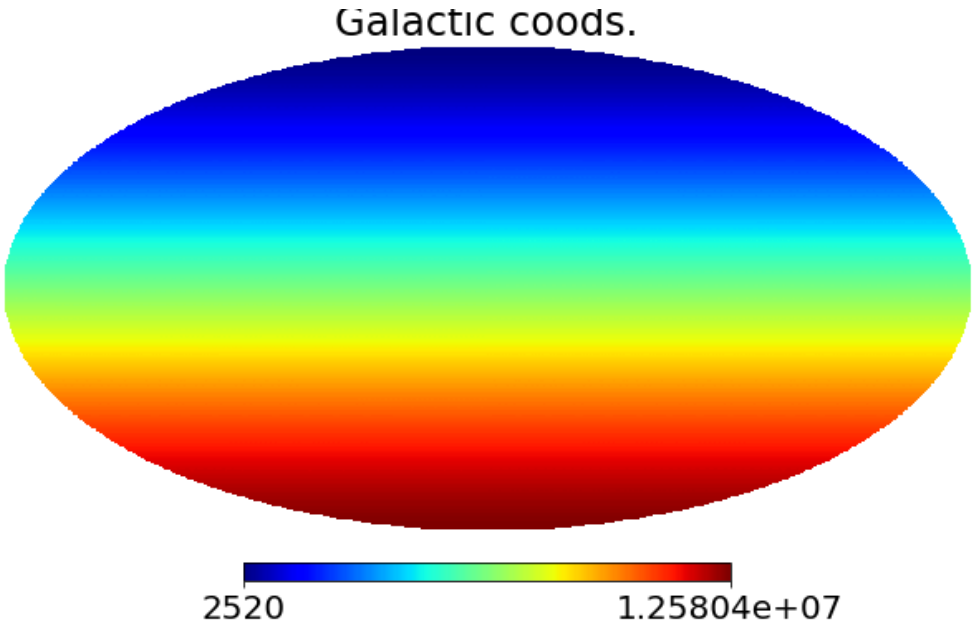
Camila Novaes

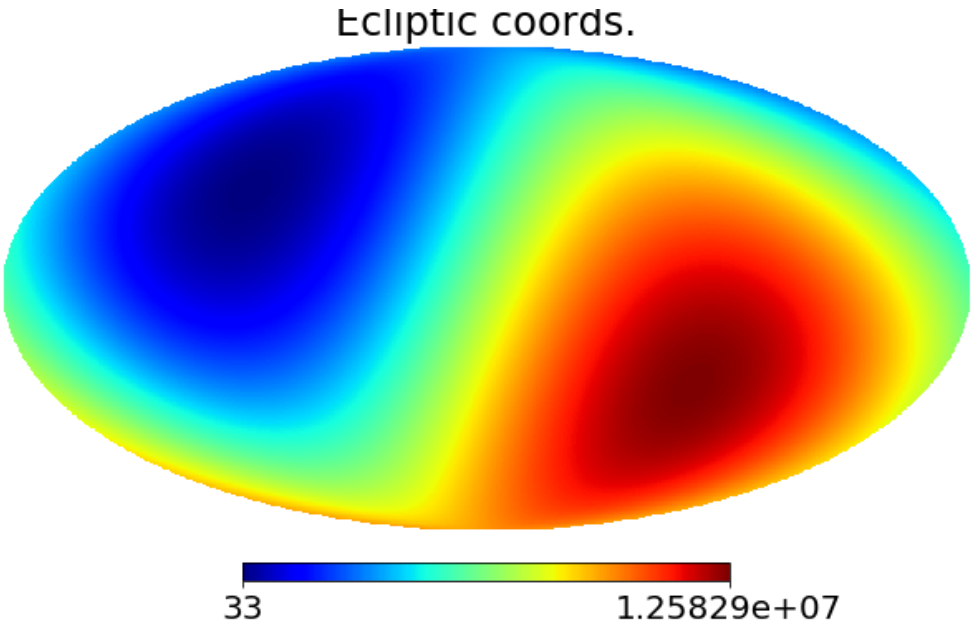
Observatório Nacional

June 6, 2017

Healpy

Previously on healpy class ... Rotation and geometry functions

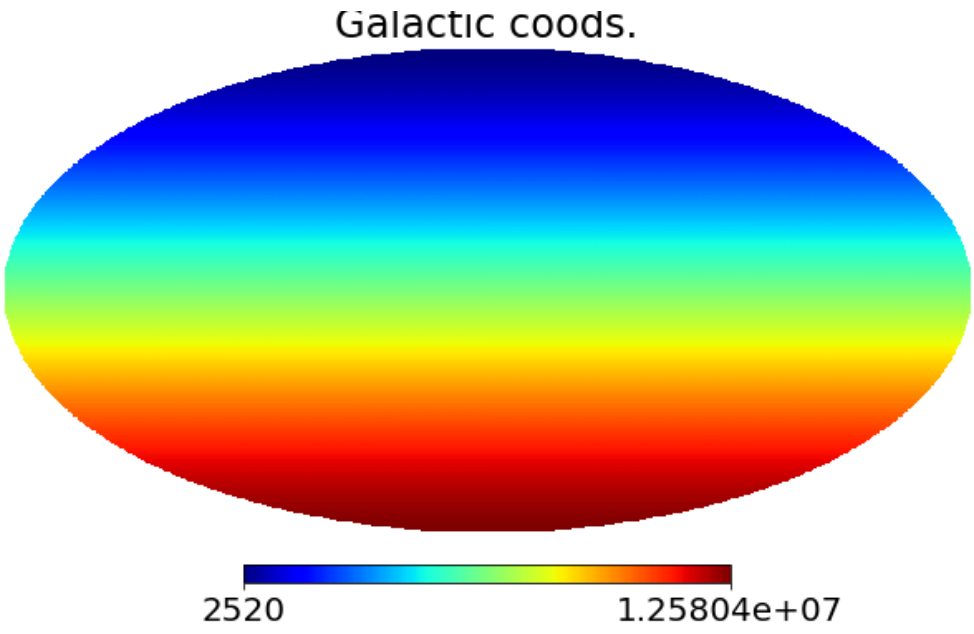


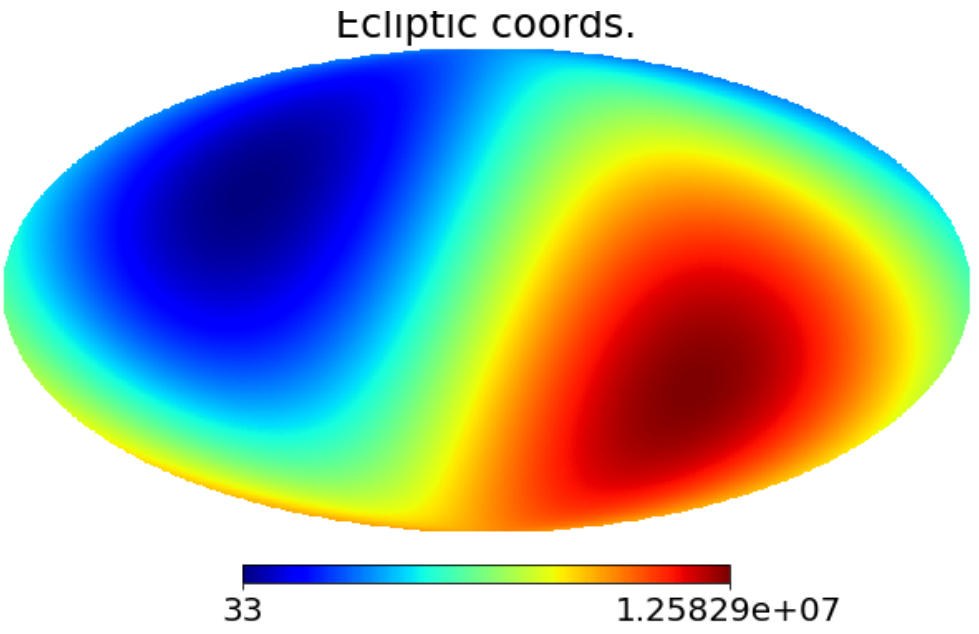


How to apply this coordinate transform in a map?

How to perform a rotation instead of a coordinate transform?

Previously on healpy class ... Rotation and geometry functions



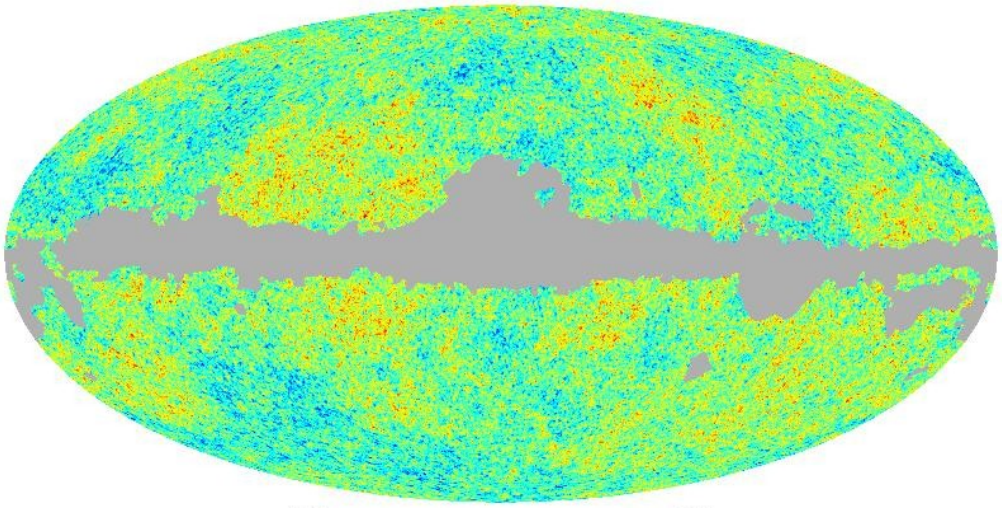


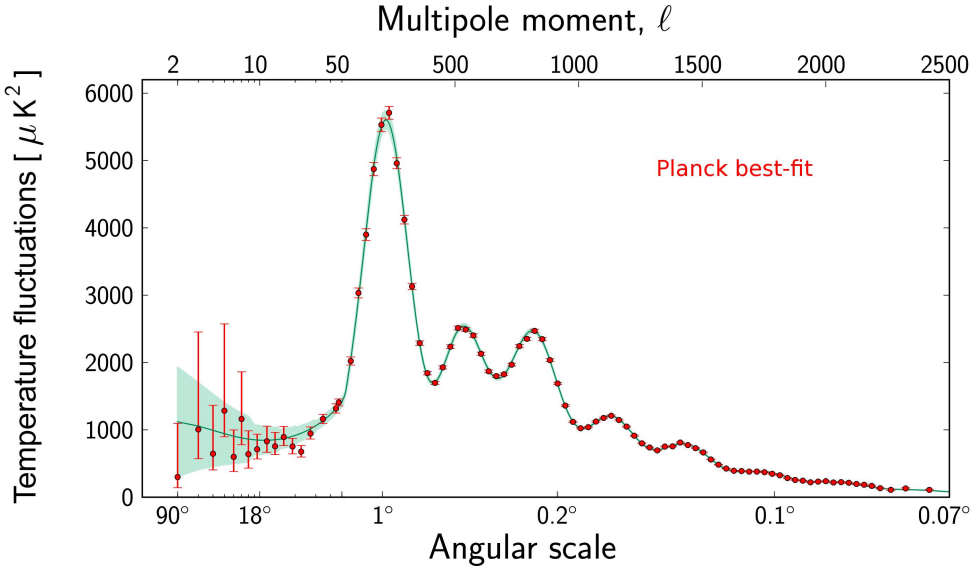
How to apply this coordinate transform in a map?

How to perform a rotation instead of a coordinate transform?

Lets see ...

- Spherical harmonic transforms -





C_ℓ, $a_{\ell m}$, smooth a map, smooth the $a_{\ell m}$, window function, ...

Spherical harmonic transforms

HEALPix conventions

A bandlimited function f on the sphere can be expanded in spherical harmonics, $Y_{\ell m}$, as

$$f(\gamma) = \sum_{\ell=0}^{l_{max}} \sum_{m} a_{\ell m} Y_{\ell m}(\gamma), \tag{4}$$

where γ denotes a unit vector pointing at polar angle $\theta \in [0, \pi]$ and azimuth $\phi \in [0, 2\pi)$. Here we have assumed that there is insignificant signal power in modes with $\ell > \ell_{max}$ and introduce the notation that all sums over m run from $-\ell_{max}$ to ℓ_{max} but all quantities with index ℓm vanish for $m > \ell$. Our conventions for $Y_{\ell m}$ are defined in subsection A.4 below.

The Spherical Harmonics are defined as

$$Y_{\ell m}(\theta, \phi) = \lambda_{\ell m}(\cos\theta) \mathrm{e}^{im\phi}$$

where the

$$\begin{aligned} \lambda_{\ell m}(x) &= \sqrt{\frac{2\ell+1}{4\pi}\frac{(\ell-m)!}{(\ell+m)!}} P_{\ell m}(x), \quad \text{for } m \geq 0 \\ \lambda_{\ell m} &= (-1)^m \lambda_{\ell |m|}, \quad \text{for } m < 0, \\ \lambda_{\ell m} &= 0, \quad \text{for } |m| > \ell. \end{aligned}$$

Spherical harmonic transforms

HEALPix conventions

Pixelating $f(\gamma)$ corresponds to sampling it at $N_{\rm pix}$ locations γ_p, $p \in [0, N_{\rm pix} - 1]$. The sample function values f_p can then be used to estimate $a_{\ell m}$. A straightforward estimator is

$$\hat{a}_{\ell m} = \frac{4\pi}{N_{\rm pix}} \sum_{p=0}^{N_{\rm pix}-1} Y^*_{\ell m}(\gamma_p) f(\gamma_p), \tag{5}$$

where the superscript star denotes complex conjugation, and an equal weight was assumed for each pixel. This zeroth order estimator, as well as higher order estimators, are imple-

Spherical harmonic transforms

$$\frac{T(\mathbf{r}_0,\hat{\mathbf{n}})-T_0(\mathbf{r}_0)}{T_0(\mathbf{r}_0)}=\sum_{l>0}\sum_{m=-l}^{+l}a_{lm}(\mathbf{r}_0)Y_l^m(\hat{\mathbf{n}})$$

$$\left\langle |a_{lm}(\mathbf{r})|^2\right\rangle = \left\langle a_{lm}(\mathbf{r})a^*_{l'm'}(\mathbf{r})\right\rangle \equiv \delta_{ll'}\delta_{mm'}C_l$$

$$\begin{aligned}
C_l &= \left\langle |a_{lm}(\mathbf{r})|^2\right\rangle = && \text{isotropy}\\
&= \left\langle \frac{\sum_m |a_{lm}(\mathbf{r})|^2}{2l+1}\right\rangle = && \text{ergodicity}\\
&= \left\langle \frac{\sum_m |a_{lm}(\mathbf{r})|^2}{2l+1}\right\rangle_{\text{space}} \simeq\\
&\simeq \frac{\sum_m |a_{lm}(\mathbf{r}_0)|^2}{2l+1}.
\end{aligned}$$

Spherical harmonic transforms

Cosmic Variance

$$\delta C_\ell = \sqrt{\frac{2}{2\ell+1}} C_\ell$$

Spherical harmonic transforms

Cosmic Variance

$$\delta C_\ell = \sqrt{\frac{2}{2\ell+1}} C_\ell$$

What if we have a partial sky???

Spherical harmonic transforms

Cosmic Variance

$$\delta C_\ell = \sqrt{\frac{2}{2\ell+1}} C_\ell$$

What if we have a partial sky???

$$\delta C_\ell = \sqrt{\frac{2}{(2\ell+1) f_{sky}}} C_\ell$$

Spherical harmonic transforms

healpy.sphtfunc.anafast

healpy.sphtfunc.anafast(*map1, map2=None, nspec=None, lmax=None, mmax=None, iter=3, alm=False, pol=True, use_weights=False, datapath=None*)

Computes the power spectrum of an Healpix map, or the cross-spectrum between two maps if *map2* is given. No removal of monopole or dipole is performed.

$$\texttt{Map} \quad \Rightarrow \quad C_\ell$$

Spherical harmonic transforms

How to use:

```
In [32]: # Just for temperature:::
    ...: mapa = hp.read_map('COM_CMB_IQU-smica_1024_R2.02_full.fits')
    ...:
    ...: Cls_TT = hp.anafast(mapa)
    ...:
```

Which is the ℓ_{max}?

Spherical harmonic transforms

How to use:

```
In [32]: # Just for temperature:::
    ...: mapa = hp.read_map('COM_CMB_IQU-smica_1024_R2.02_full.fits')
    ...:
    ...: Cls_TT = hp.anafast(mapa)
    ...:
```

Which is the ℓ_{max}?

$$\ell_{max} = 3072$$

```
In [53]: res = hp.nside2resol(1024, arcmin = True)

In [54]: print('Resolution=', res, 'arcmin')
Resolution= 3.43548641182 arcmin

In [55]: theta = (180./lmax)*60.

In [56]: print('Theta=', theta, 'arcmin')
Theta= 3.515625 arcmin
```

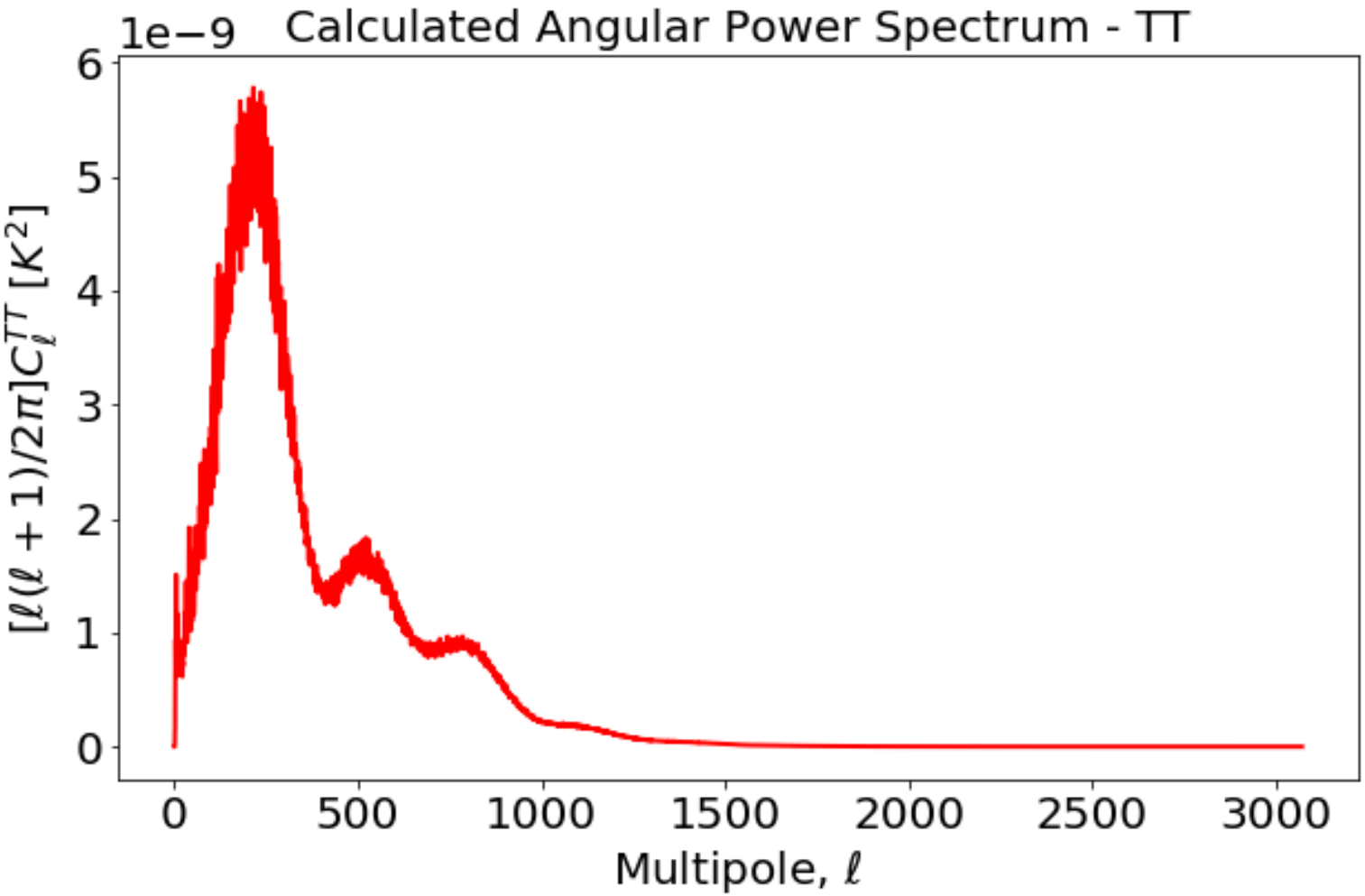
Spherical harmonic transforms

Script to plot the C_ℓ:

```
In [34]: #########
    ...: # Plot:::
    ...:
    ...: # TT:::
    ...: lmax = len(Cls_TT)
    ...: ell = np.arange(lmax)
    ...:
    ...: Dls_TT = ell*(ell+1)*Cls_TT/(2.*np.pi)
    ...:
    ...: pyplot.plot(ell, Dls_TT,linewidth=2.0, color="red") #label="CMB Cls"
    ...: #
    ...: #plt.ylim(-300.,1000.)
    ...: #pyplot.xscale('log')
    ...: pyplot.title('Calculated Angular Power Spectrum - TT',fontsize=20)
    ...: pyplot.xlabel('Multipole, $\ell$',fontsize=20)
    ...: pyplot.ylabel('$[\ell(\ell + 1)/2\pi] C_\ell^{TT}$ $[K^2]$',fontsize=20)
    ...: #pyplot.legend(loc='best')
    ...: #
    ...: fig = pyplot.gcf()
    ...: fig.set_size_inches(10, 6)
    ...: pyplot.savefig("Cls.png")
    ...: pyplot.show()
    ...:
```

Spherical harmonic transforms

Result:



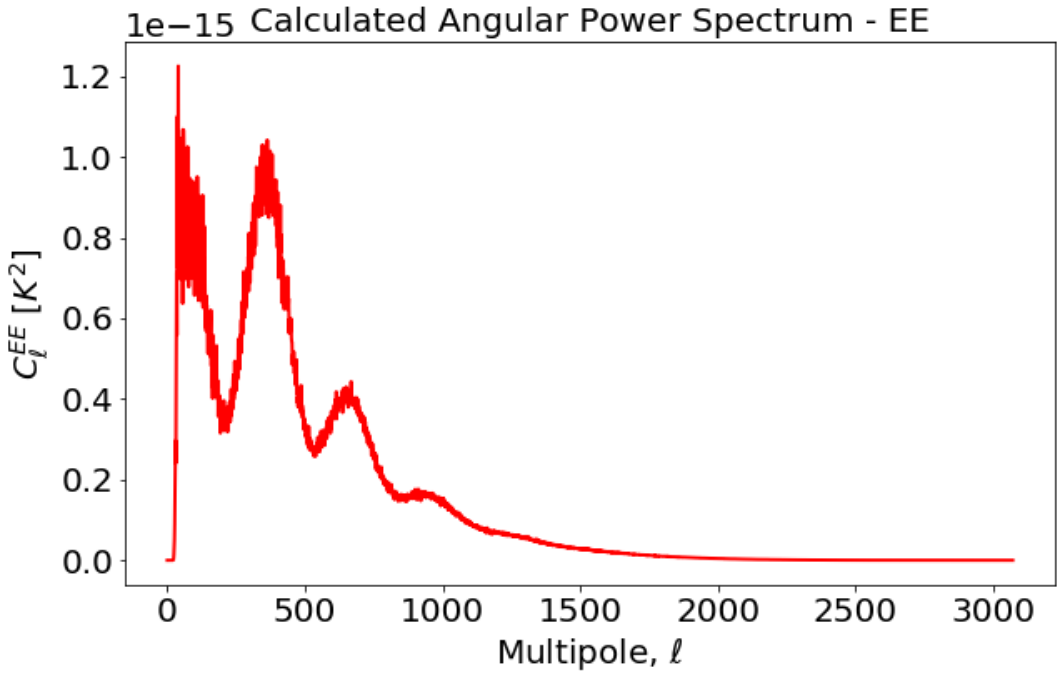
Spherical harmonic transforms

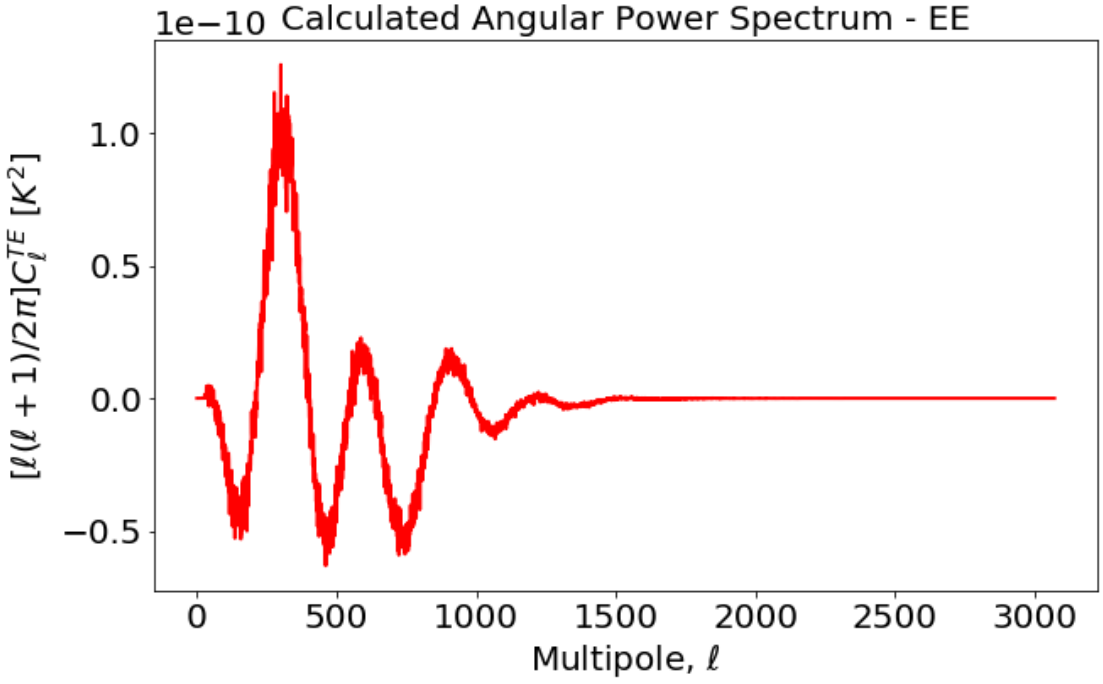
How to use:

```
In [45]: # For temperature and polarization:::
    ...: mapa = hp.read_map('COM_CMB_IQU-smica_1024_R2.02_full.fits', field=[0,1,2])
    ...:
    ...: Cls = hp.anafast(mapa)
    ...:
    ...: Cls_TT = Cls[0]
    ...: Cls_EE = Cls[1]
    ...: Cls_BB = Cls[2]
    ...: Cls_TE = Cls[3]
    ...: Cls_EB = Cls[4]
    ...: Cls_TB = Cls[5]
    ...:
```

Spherical harmonic transforms

Result:





Spherical harmonic transforms

healpy.fitsfunc.write_cl

healpy.fitsfunc.write_cl(*filename, cl, dtype=<type 'numpy.float64'>*)

Writes Cl into an healpix file, as IDL cl2fits.

$C_\ell \quad \Rightarrow \quad$ fits file

Spherical harmonic transforms

How to use:

```
In [105]: # For one Cls array:
     ...: hp.write_cl('calc_Cls0.fits', Cls[0])

In [106]: # For all:
     ...: hp.write_cl('calc_Cls1.fits', Cls)
```

Spherical harmonic transforms

healpy.fitsfunc.read_cl

healpy.fitsfunc.read_cl(*filename, dtype=<type 'numpy.float64'>, h=False*)

Reads Cl from an healpix file, as IDL fits2cl.

fits file $\Rightarrow$ C_ℓ

Spherical harmonic transforms

How to use:

```
In [98]: # To open just the FIRST extension:
    ...: Cls2 = hp.read_cl('COM_PowerSpect_CMB_R2.02.fits', h=True)

In [99]: len(Cls2)
Out[99]: 4

In [100]: len(Cls2[0])
Out[100]: 28

In [101]: # To access the header and chose the extension you want to open:
     ...: from astropy.io import fits
     ...: hdulist = fits.open('COM_PowerSpect_CMB_R2.02.fits')
     ...: hdulist.info()
     ...:
Filename: COM_PowerSpect_CMB_R2.02.fits
No.    Name         Type      Cards   Dimensions   Format
  0  PRIMARY     PrimaryHDU       4   ()
  1  TTLOLUNB    BinTableHDU     46   28R x 4C   [I, E, E, E]
  2  TELOLUNB    BinTableHDU     46   28R x 4C   [I, E, E, E]
  3  EELOLUNB    BinTableHDU     46   28R x 4C   [I, E, E, E]
  4  TBLOLUNB    BinTableHDU     46   28R x 4C   [I, E, E, E]
  5  EBLOLUNB    BinTableHDU     46   28R x 4C   [I, E, E, E]
  6  BBLOLUNB    BinTableHDU     46   28R x 4C   [I, E, E, E]
  7  TTHILBIN    BinTableHDU     46   83R x 5C   [E, I, I, E, E]
  8  TTHILUNB    BinTableHDU     42   2479R x 3C   [I, E, E]
  9  TEHILBIN    BinTableHDU     46   66R x 5C   [E, I, I, E, E]
 10  TEHILUNB    BinTableHDU     42   1967R x 3C   [I, E, E]
 11  EEHILBIN    BinTableHDU     46   66R x 5C   [E, I, I, E, E]
 12  EEHILUNB    BinTableHDU     42   1967R x 3C   [I, E, E]
```

Spherical harmonic transforms

How to use:

```
In [48]: Cls_TTLOLUNB = hp.read_cl(hdulist[1])
    ...: Cls_TTHILUNB = hp.read_cl(hdulist[8])
    ...:

In [49]: len(Cls_TTLOLUNB), len(Cls_TTHILUNB)
Out[49]: (4, 3)

In [50]: len(Cls_TTLOLUNB[0]), len(Cls_TTHILUNB[0])
Out[50]: (28, 2479)

In [51]: Cls_TTLOLUNB[0]
Out[51]: array([ 2,  3,  4, ..., 27, 28, 29], dtype=int16)

In [52]: Cls_TTHILUNB[0]
Out[52]: array([  30,   31,   32, ..., 2506, 2507, 2508], dtype=int16)
```

Spherical harmonic transforms

How to use:

```
In [84]: n1 = len(Cls_TTLOLUNB[0])

In [85]: n2 = len(Cls_TTHILUNB[0])

In [86]: n = n1 + n2

In [87]: ell=np.zeros(n)

In [88]: Dls=np.zeros(n)

In [89]: ell[0:n1] = Cls_TTLOLUNB[0]

In [90]: ell[n1:n] = Cls_TTHILUNB[0]

In [91]: Dls[0:n1] = Cls_TTLOLUNB[1]

In [92]: Dls[n1:n] = Cls_TTHILUNB[1]
```

Plot!

Spherical harmonic transforms

Result: