Internal Report IASF-BO 595/2011

On the accuracy of map rotation for reference frame tranformation in pixel and harmonic space: total intensity (temperature) maps

D. Molinari[1,2] and C. Burigana[1,3]

[1]*INAF-IASF Bologna, Via P. Gobetti 101, I-40129, Bologna, Italy*
[2]*Dipartimento di Astronomia, Università degli Studi di Bologna, Via C. Ranzani 1, I-40127 Bologna, Italy*
[3]*Dipartimento di Fisica, Università degli Studi di Ferrara, Via G. Saragat 1, I-44100 Ferrara, Italy*

Internal Report IASF-BO/595/2011

31 October 2011

On the accuracy of map rotation for reference frame tranformation in pixel and harmonic space: total intensity (temperature) maps

D. Molinari[1,2] and C. Burigana[1,3]

[1]*INAF-IASF Bologna, Via P. Gobetti 101, I-40129, Bologna, Italy*
[2]*Dipartimento di Astronomia, Università degli Studi di Bologna, Via C. Ranzani 1, I-40127 Bologna, Italy*
[3]*Dipartimento di Fisica, Università degli Studi di Ferrara, Via G. Saragat 1, I-44100 Ferrara, Italy*

SUMMARY – One fundamental instrument that scientists are mostly using these days is the possibility of joining multiple images of the same source at different frequencies. A lot of difficulties have to be gotten over and one of them is the fact that different images of the same source have different reference frames. A change of coordinates is necessary. In this work we analyse the quality of `HEALPix` tools in changing the reference frame of microwave full sky maps in temperature. This package, widely used by *Planck* Collaboration, uses two different ways for changing coordinates. We call them the pixel and harmonic method. Maps used for this analysis are: only CMB maps generated by `HEALPix Fortran 90` facility `synfast` from the WMAP 1st year best fit of CMB angular power spectrum (APS); maps directly obtained from WMAP satellite; and the same maps after a previous analysis of foreground that has been removed. In this work we point out how both method affect these different kind of maps with the aim of explaining what method is recommended depending on the characteristics of the maps that would be preserved during the rotation.

1 Introduction

The measurements of the anisotropies of the Cosmic Microwave Background (CMB), obtained by WMAP satellite, ground-based and balloon-borne experiments like DASI and BOOMERang, offered extraordinary powerful tests of the Cosmological Model. Maps of the Universe in the microwaves allowed scientists to put constraints to the cosmological parameters and consequently to understand the nature of the early stage of the Universe and of the cosmic structure formation [1, 2].

In recent years scientists have been looking for better measurements of the CMB. It has been done in two different ways: launching new satellites with more powerful instruments (like the *Planck* ESA mission) and improving the data analysis methods to remove more accurately various kinds of spurious noise from the available maps. This second method entails the challenge of always reach best knowledge of the physics of microwave emissions of all the sources standing between us and early Universe, first of all our own Galaxy. For this reason a great help can come from observations in other ranges of frequencies made by other experiments. In fact, knowledge of the emissions of a source at different frequencies allows to build its frequency spectra that can bring a lot of information about the physical processes that generated the emissions.

Collecting pictures of the same source made by different experiments can be very difficult. Problems may come from different characteristics of the instruments or size, resolution of maps that include the source. All these problems can be tackled by knowing the properties of the instruments involved in experiments. Another important issue is that satellites and ground-based missions generate maps with different astronomical coordinates. Consequently it is necessary a change of coordinates that may cause an increment in the map noises.

In this work we analyse qualities and performances in changing coordinates of `HEALPix` [3, 4], an important tool mostly used by the *Planck*[1] Collaboration and free downloadable from `HEALPix`[2] website. In section 2 we present the most important features of the `HEALPix` code and methods that can be used to change coordinates. In sections 3 and 4 we analyse performances of our routine that uses these methods upon maps respectively of only artificial CMB (with and without beam smoothing), foreground reduced WMAP maps and original WMAP maps. In the last section we present conclusions of our analysis.

2 `HEALPix` tools and rotation procedures

`HEALPix`, an acronym for Hierarchical Equal Area isoLatitude Pixelization of a 2D-sphere, is a software package for maps projection, pixelization and analysis of a 2D-sphere. The pixelization [5] can be thought of as mapping the sphere to twelve square facets (diamonds) on the plane followed by the binary division of these facets in pixels. The software collects routines in different computer languages (`C++`, `IDL`, `C` and `Fortran90` (`F90`)) and reads and writes `.fits` files. In this work, sky maps analysed contain full sky temperature signal, corresponding to total intensity Stokes parameter, in a single array of *npixel* dimension, where *npixel* is related to the resolution (*nside*) of the maps and given by the formula

$$npixel = 12 \times (nside)^2 \,, \tag{1}$$

The number of pixel is also associated to the sides of the sky area covered by every pixel by the formulas

[1] http://www.rssd.esa.int/
[2] http://healpix.jpl.nasa.gov/

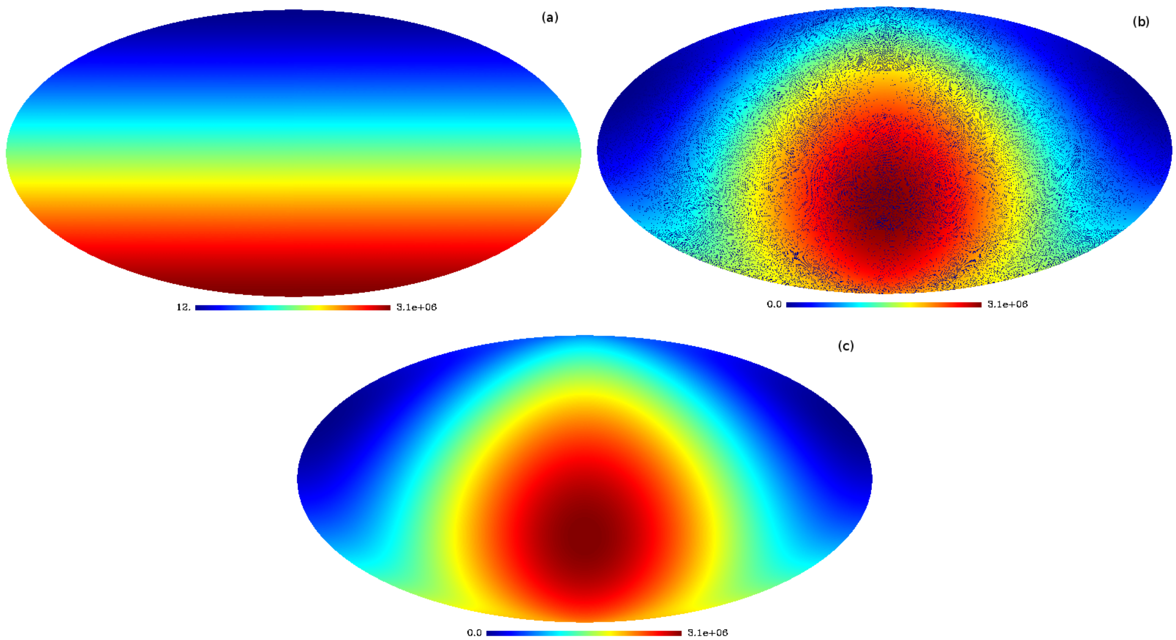


Figure 1: An example of what happens if the direct change of coordinates is used. (a) The initial map to be rotated in new coordinates. (b) The map in new coordinates. In this map there are pixels with zero value as consequence of errors in changing the coordinates. (c) Map in new coordinates obtained by rotating in the "opposite" direction. There are no void pixels.

$$\begin{aligned} area_{sr} &= \left(\frac{4\pi}{npixel}\right) \\ side_{sr} &= \left(\frac{4\pi}{npixel}\right)^{\frac{1}{2}} \\ side_{degree} &= \left(\frac{4\pi}{npixel}\right)^{\frac{1}{2}} \times \frac{180}{\pi} \end{aligned} \tag{2}$$

For example in a map with $nside = 64$ the area covered by every pixel is 0.916 degree. In `HEALPix` are already implemented functions for changing coordinates in `IDL` and `F90`, but they work in two very different ways: `IDL` routine works at pixel level by rotating 3D position vectors of every pixel with Euler Matrix, `F90` routine works at harmonic level by rotating a_{lm} coefficients obtained by a spherical harmonics expansion of the sky maps. These methods have both strong and weak points and, in order to reveal them, in the following subsections we will describe them in detail. CMB maps generated by WMAP satellite, are publicly available in the LAMBDA website [3] and are in Galactic ('G') coordinates, but it could be useful to have them in Equatorial ('C' or 'Q') or Ecliptical ('E') coordinates. In the following sections, rotation will be ever performed from Galactic coordinates to Ecliptical coordinates or vice versa.

2.1 Change of coordinates in pixel space

As already mentioned, the `IDL` facility `rotate_coord` provides a mean to rotate a set of 3D position vectors between astronomical coordinates systems by using Euler matrix. Every pixel of the map is determined by the 3D position vector of its center, so rotation can be done easily by associating the rotated vector of an original pixel to the pixel of the new coordinates. If we make the change by directly attributing the value of a pixel in the initial coordinates to

[3] http://lambda.gsfc.nasa.gov/

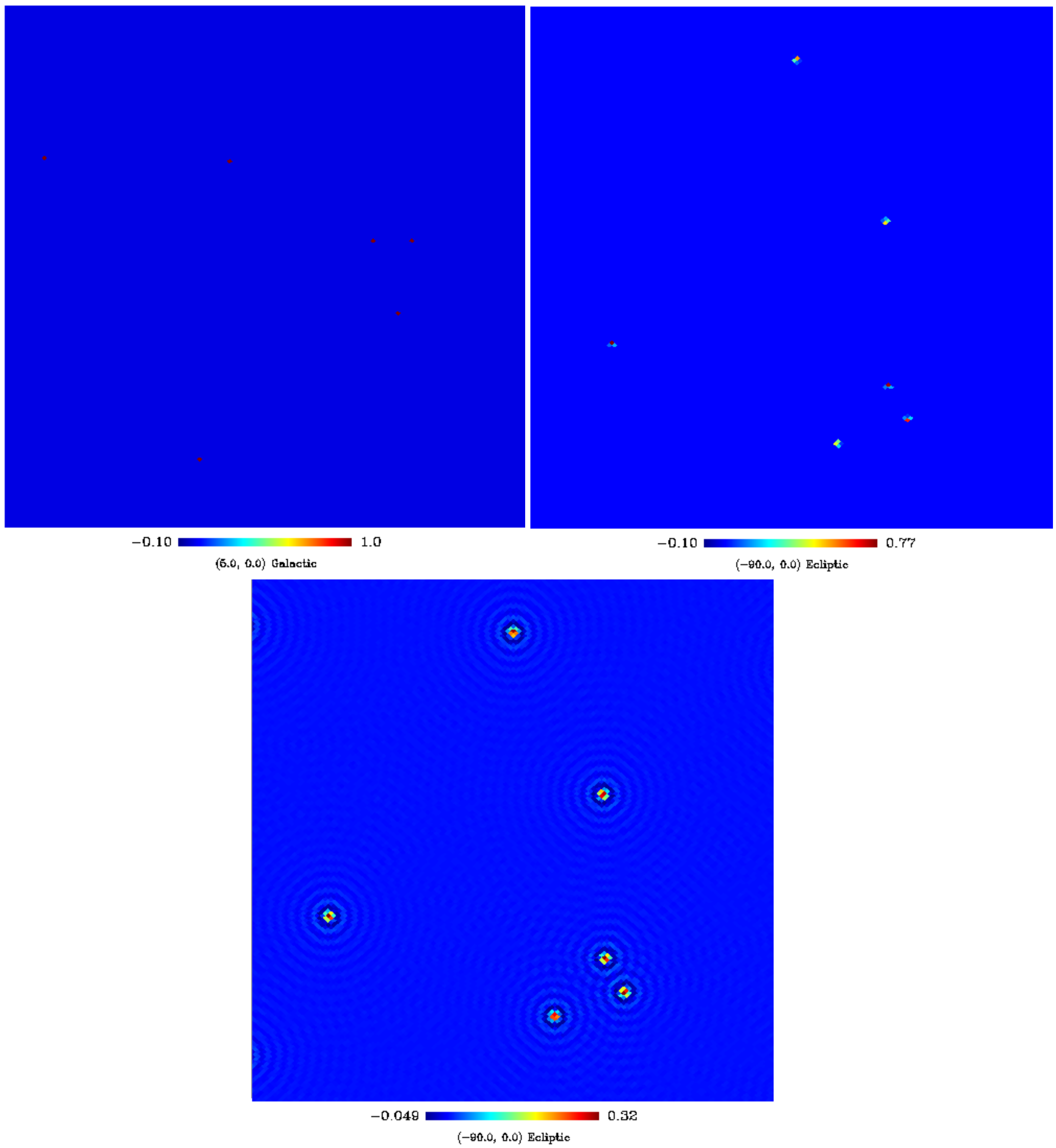


Figure 2: An example of what happens in rotating maps with strong point sources and very low background with the pixel and harmonic method. (a) The initial map to be rotated in new coordinates, and the map rotated (b) by pixel method and (c) by harmonic method. Respect to pixel method sources are smoothed on more pixel. However this error affects very weakly CMB temperature maps.

a pixel in the final coordinates, some of the pixels in the new map will be without a value as shown in Figure 1 (a-b). This is due in part to the approximations made during the change of coordinates, but the most part of them is due to the fact that pixels in different coordinates are not perfectly superimposed, so the rotated vector can not fall exactly upon the correct center of the pixel in new coordinates. Consequently program can attribute a pixel value not to the right pixel, but to one of its neighbours.

To avoid pixels without a value, changes of variables are made in the "opposite" direction. Our `IDL` procedure (called `rotation2.pro`) generates a new map in new coordinates and attribute every pixel of the new map to a pixel of the map in old coordinates. The new map obtained in this way has no null pixel as shown in Figure 1(c). In this case there are still errors due to a mismatch of pixels, but the new pixel will not be void. It will take the value from the wrong pixel, but this pixel is a neighbour of the right pixel, so in the case of CMB maps in which there are no great differences between a pixel and its neighbour, errors are limited.

2.2 Change of coordinates in harmonic space

Another way to rotate CMB maps is to work in Fourier space, by rotating directly a_{lm} coefficients obtained by a spherical harmonics expansion of the map. In order to do this, we wrote new routines in `F90` by using `HEALPix` subroutines. The `F90` subroutine `rotate_alm*` transforms the scalar a_{lm} coefficients to emulate the effect of an arbitrary rotation of the underlying map. The rotation is done by directly rotating a_{lm} coefficients. The general idea of the `F90` program (called `harmonic_rotation.f90`) is to read a temperature CMB map and to analyse it in order to obtain coefficients a_{lm} from its expansion in spherical harmonics. Then it rotates these coefficients to the new coordinates, synthesizes a new rotated map with the new coefficients a_{lm} and writes it in a new `.fits` file. In this case the mismatched pixels are absent because we are not using pixels, but there is another problem that we found when we rotated a $nside = 512$ map with all the pixels equal to zero except for a few thousands that have value 1 as shown in Figure 2 (a). In Figure 2 (b) there is the map rotated with "harmonic method". As shown by the color scale the effect of rotation is a strong smoothing of the non-zero pixels that lost two-third of their original signal and the appearance of alias pixels of the same intensity. The rotated original pixels are unrecognisable. For maps with strong point sources in a low background this routine is not useful, but for temperature CMB map in which there are not this kind of sources these errors are negligible.

There are another issue very interesting. The previous map has been also rotated by using the "pixel method". For both the rotated maps we eliminated alias pixels by sorting pixels in intensity and choosing the ones with highest value. In Figure 3 (a) are shown positions and intensities of some of these pixels of the two maps. In addition to the lack of intensity, positions of pixel differ all at least of one pixel and not ever in the same direction. This is probably a bug of the software that cause a different rotation result when one uses pixel or harmonic method. For a more deep understand of this effect, we generated a map with null pixel value except for 91 Normalized Gaussian Point Spread Function (PSF) with FWHM of 10 arcmin, similar to WMAP and *Planck* satellites characteristics. PSF are placed randomly on the map by avoiding superpositions of their codes. PSF are better modelled by an harmonic expansion of the map than single pixel sources, so we expected to cut down the aliasing effects observed in the map rotated with harmonic method in the previous case. We rotated the new map both with the pixel and harmonic method. In Figure 4 are shown the original map and the two rotated maps. As you can see in the table of Figure 4 (b) the peak intensities of the PSF after rotation are very close to 1 confirming that the aliasing effect in the map rotated with harmonic method is negligible, so we could move to a stronger

pixel method		harmonic method	
pixel num.	T	pixel num.	T
302	0,890625	301	0,368868
2539	0,875	2540	0,349093
11747	0,859375	11748	0,343346
23122	0,84375	23121	0,338425
25115	0,84375	25118	0,337704
26216	0,84375	26211	0,336631
29042	0,84375	29041	0,33647
48799	0,828125	48840	0,335981
49785	0,828125	49782	0,335489
55298	0,828125	55297	0,334935
.	.	.	.

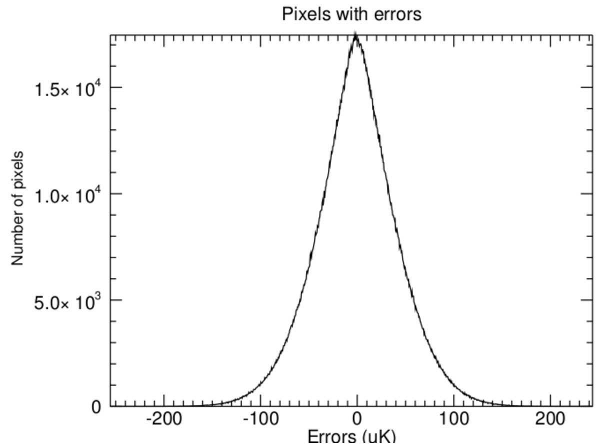


Figure 3: (*On the left:*) Table with some of pixel positions and temperature of non-zero pixels of rotated maps with pixel and harmonic methods. Notice that pixel positions differ for at least one pixel. (*On the right:*) Differences in temperature between the two rotated maps of an original synthetic only CMB map. Errors are taken by subtracting the two maps pixel per pixel and making an histogram of non-zero pixels. Differences are of the order of CMB fluctuation.

comparison between the two methods. In the same table one can see that there is still the mismatch in pixel positions of the PSF peaks between the rotated map with the two different methods. When we rotate the same temperature CMB map (see the follow for details on the map) by using both the methods and compare the two rotated maps, the differences are of the order of CMB fluctuation (order of 100 μK for a signal of at maximum 400 μK) as shown in Figure 3 (b). Probably the pixel mismatch is the cause of these errors and this is very problematic because we don't know what the exact rotation is and furthermore it is impossible to compare directly rotated maps with the two different methods.

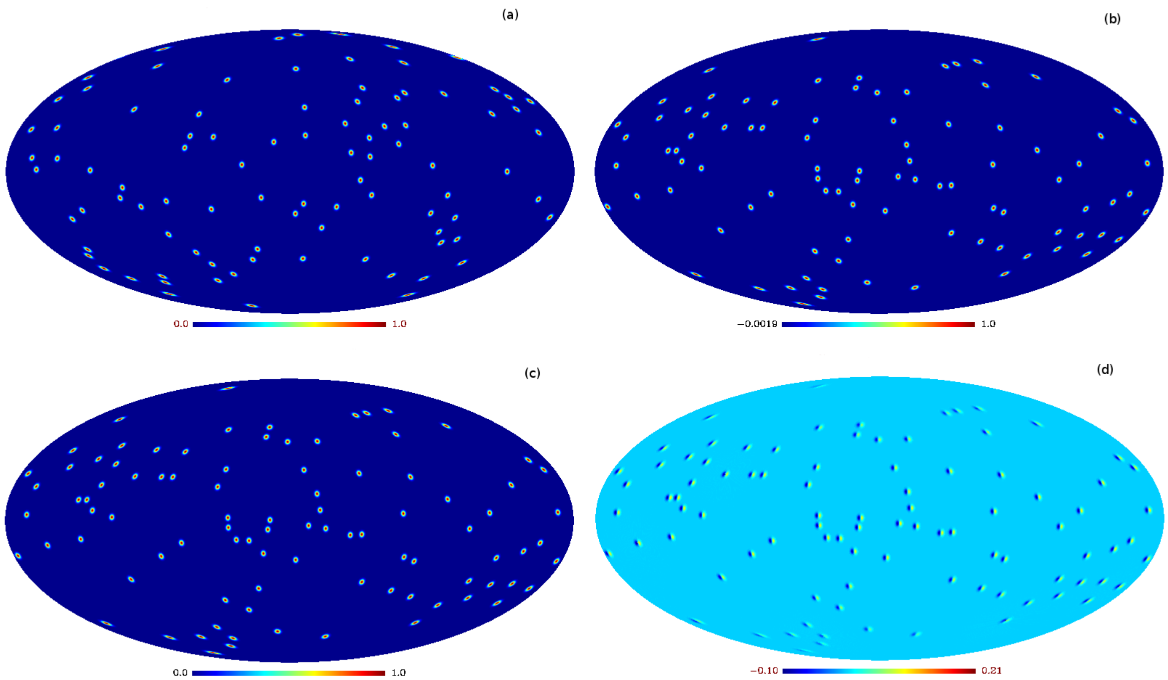
Consequently in the following sections we analyse the errors that affect pixels and the Angular Power Spectrum (APS) of the rotated map by making comparisons between the twice rotated map with one method and the original one separately and the comparison between the two methods will be done only after a statistical analysis of temperature errors that affect pixel and APS of the maps.

3 Results: pure CMB simulated maps

In this section we present results obtained by rotating synthetic maps of only CMB, without foreground. These maps are generated by an `HEALPix F90` facility called `synfast` with $nside = 512$ in Galactic coordinates. To generate a map, `synfast` needs only a file with the CMB APS. We used the APS released by WMAP 1st year best fit [6]. At the moment maps are generated without beam.

Before presenting the results we have to describe the method used to obtain them. Errors that we will present are obtained by changing the coordinates twice in order to return to a map in the initial coordinates. For example a CMB map in Galactic coordinates is rotated in Ecliptic coordinates and the new map is rotated back in Galactic coordinates. In the routine `medvarerror.pro` the two maps in Galactic coordinates are subtracted pixel per pixel. After subtraction, we histogram every pixel with a non-zero value and this is what we will report in the following figures.

A second analysis is done on the APS of the maps. By using the `HEALPix F90` facility called `anafast`, we obtained APS of the maps allowing us to compare the effect of rotation also in harmonic space by comparing the APS.



pixel method		harmonic method	
pixel num.	T	pixel num.	T
33177	0,995213	33177	0,995167
194969	0,996028	194969	0,996877
216865	0,995273	216866	0,99612
237120	0,995159	237120	0,996245
276657	0,994033	276658	0,997757
343006	0,996205	343007	0,997048
355401	0,995379	355402	0,997458
383056	0,995296	383057	0,996071
434803	0,99416	434804	0,997028
463781	0,994231	463782	0,995889
514991	0,994856	514992	0,997878
519187	0,996214	519188	0,997322
564934	0,994297	564935	0,996879
615976	0,996351	615977	0,996401
624513	0,995133	624514	0,996784
638380	0,995437	638381	0,99681
730875	0,993996	730876	0,99657
743496	0,994779	743497	0,997297
.	.	.	.

Figure 4: (a) Original map with null pixel except for 91 Normalized Gaussian Point Spread Function randomly placed. (b) The map rotated with harmonic method. (c) The map rotated with pixel method. (d) Map of the differences between the maps resulting from the rotation with the two methods. (e) Table with position and intensities of some of the PSF peaks. Note the pixel mismatch in peak position between the maps rotated with pixel and harmonic methods.

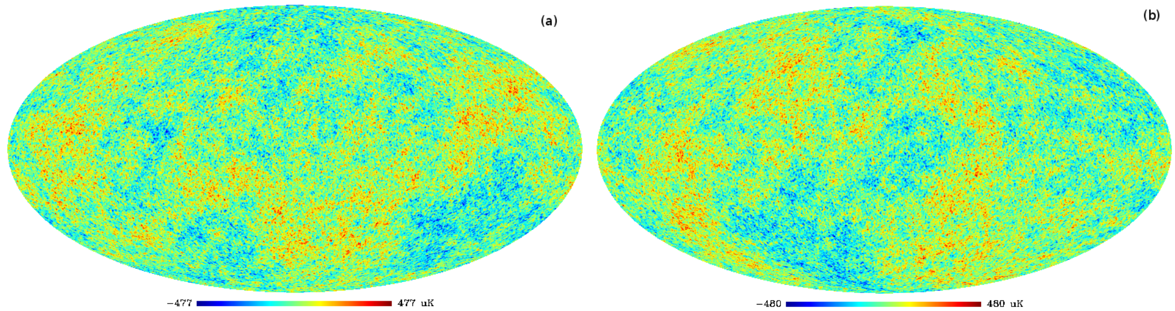


Figure 5: (a) Example of an only CMB map in Galactic coordinates and (b) the results after rotation in Ecliptic coordinates. Map looks very similar, but deep analysis revealed that the rotated map is affected by errors generated during rotation.

3.1 "Pixel" method

In Figure 5 we show the original synthetic map in Galactic coordinates and the map in Ecliptic coordinates rotated with the "pixel" method. After twice rotations, errors that affect maps are of the order of 100 μK as shown in Figure 6 (a). They are very high, in fact they are of the same order of CMB fluctuations.

In order to find a way to lower these errors, maps are expanded by the mean of the `HEALPix IDL` facility `ud_grade`. Programs used to perform change of coordinates are reported in Appendix A. In principle, only the final rotated map should be expanded, but we chose to expand both the maps to have an occasion of optimizing the code. It will be explained in details up ahead). To expand a map means to generate a map with a larger *nside*. Every pixel is substituted by a larger number of pixel having the same value of the previous one. No errors are generated in this procedure. After the change of coordinates, the expanded map in the new coordinates is downgraded to the original *nside*. In this step the routine makes an average over all the pixels that fall in the new larger pixel of the downgraded map. In this way the errors possibly committed during the change of coordinates are averaged out and although the number of pixels with errors increase in number (at $nside = 512$ number of pixels that shows an error is 20%, while at $nside = 4096$ this fraction increase to 98%), the errors committed are considerably lower, as shown in Fig. 6 in which are compared the errors committed by changing the coordinates of a CMB map (a) without the expansion and (b) by expanding the map to $nside = 4096$. Moreover the higher is the expansion the lower are the final errors.

If we make a comparison between these errors and the sensitivity of WMAP satellite from which the map has been obtained, we notice that W-band WMAP sensitivity after 7-years observation [7] is of the order of 56μK while errors committed by expanding maps to $nside = 4096$ are of the order of 5μK. Errors are also lower than sensitivity of *Planck* satellite [8]. So changes of coordinates does not more affect maps.

As a further optimization, we tried to limit the computational time that computers need to change the coordinates. We created a routine (called `rotation_matrix.pro`) that generates a matrix that contains the links between old coordinates and the new ones. In this routine one can choose the two coordinates and the resolution (*nside*) of the expanded map that has to be changed. Links are still determined by the `HEALPix IDL` facility `rotate_coord`. Hence we wrote a second routine (called `rotation3.pro`) that reads the matrix (saved as a `.fits` file) and makes the change of coordinates. Time that computer needs to generate the matrix is the same that it needs to change the coordinates with the old routine, but now the first routine has to be run only one time and then the matrix generated in this way can be used by the routine `rotation3.pro` every time is needed and for any map. Computers need a

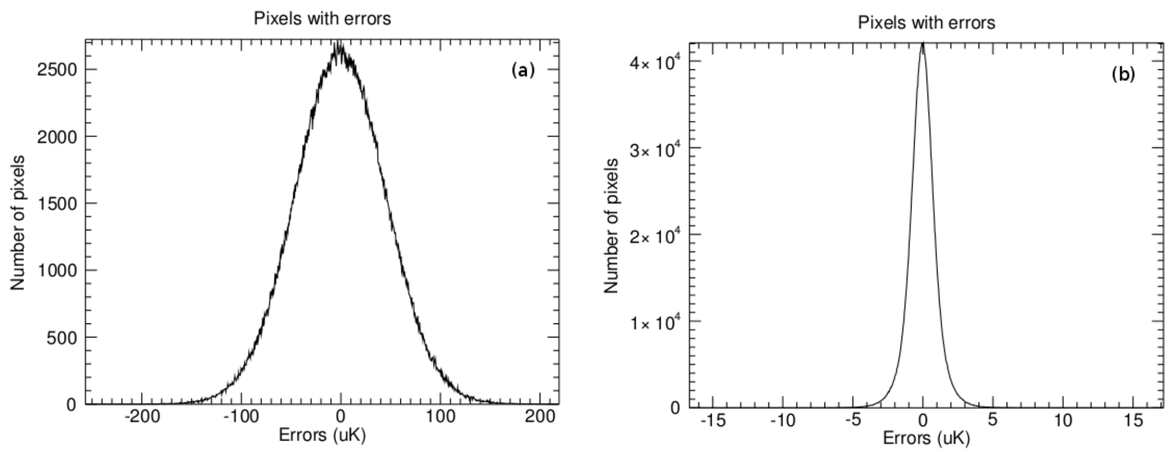


Figure 6: (a) Histogram of temperature errors of non-zero pixel after the subtraction between the original map and the one rotated twice. (b) The same of (a), but in this case map was prograded to $nside = 4096$ before rotation and downgrade to the original $nside = 512$ after rotation.

lower time to read a file than to calculate the change of coordinates, as shown in Figure 7 in which are displayed, for different resolutions, (a) the time that our computer needed to directly change the coordinates with `rotation2.pro` and (b) the ones that it needed to run `rotation3.pro`. In order to obtain these time estimations we used one node of the CINECA PLX cluster that contains 2 Intel(R) Xeon(R) Westmere six-core E5645 processors, with a clock of 2.40GHz and 47 GB of allocatable memory. For all the map resolutions, the use of matrix allow to save almost 40% of computational time independently of the nside of the map. This means that the higher is the nside of the map, the higher is the computational time saved. The results obtained and the errors committed with this two routines are exactly the same.

We showed that by prograding and downgrading the map to higher resolution the errors that affect the rotated maps decrease. There are two important limits for this kind of analysis. The first one is the computational time. In Figure 7 it is shown that the rotation can least only several seconds to be done, but we want to notice that CINECA PLX is one of the most powerful computers cluster in the word. It is ranked at the 54^{th} position in the TOP500 list[4]. A more common computer can spend several hours to rotate the maps and the more is the prograded map resolution, the more is computational time needed. The second and most important limit is the RAM memory that computers need to allocate the maps. A map at $nside = 512$ in double precision can need about 25Mb of allocatable memory, but a map at $nside = 4096$ needs more than 1.6GB. A program that allocate this huge memory for the original map and the rotated map and by considering all the vectors used to calculate the rotation is impracticable for a common computer. The allocatable memory needed grows up with the square of the nside of the map. For this two important reasons it is impossible to reach higher resolution than $nside = 4096$ without the use of computers cluster.

Before presenting results about our APS analysis of rotated maps, we want to report an interesting consideration about the capability of the `HEALPix F90` facilities `synfast` and `anafast` in creating maps from a given APS and extracting APS from maps respectively. By using `synfast` we generated a synthetic CMB map at $nside = 512$ without Galactic foreground and without beam. Then, by using `anafast`, we analysed the CMB map re-obtaining its APS. As expected, `synfast` can reproduce maps with high multipoles (as allowed by the facility it can reach $l_{max} = 3 \times nside$ where l_{max} is the highest multipole and $nside$ the

[4]http://www.top500.org/list/2011/06/100

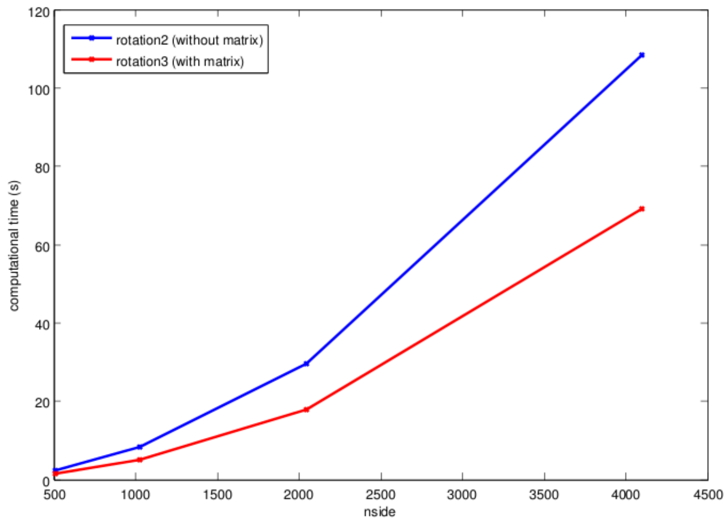


Figure 7: Computational time necessary at different resolution (blue) to directly change the coordinates and (red) to read matrix previously generated and change coordinates.

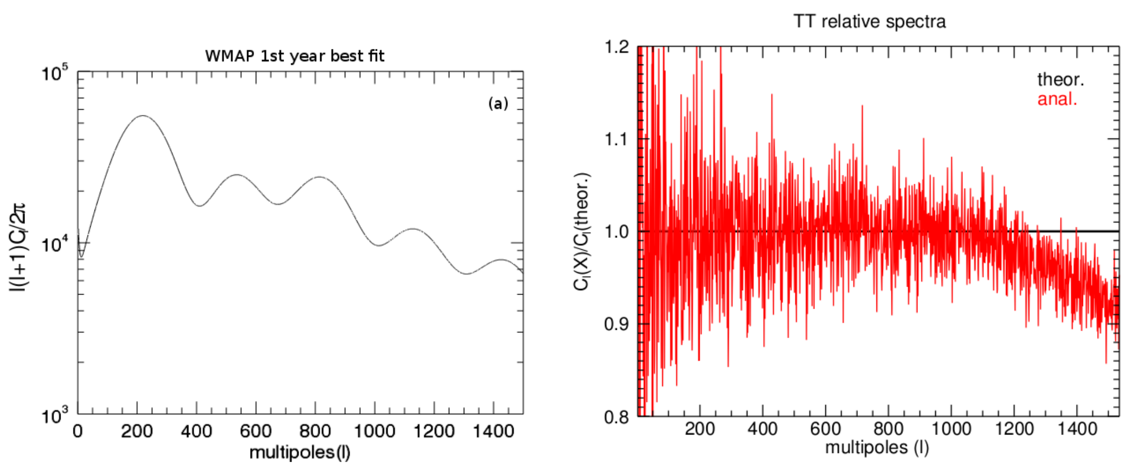


Figure 8: (a) WMAP 1st year best fit of the CMB APS. (b) Angular power losses by simulating CMB map with *synfast* and analyzing them with *anafast*. It is normalized to the WMAP 1st year best fit used to synthesize the map. APS are corrected for the window function due to the map pixelization.

resolution of the map), and `anafast` can reproduce the initial APS up to $l_{max} = 3 \times nside$ maintaining low errors. In fact, as shown in Figure 8 the power losses are negligible at 512 and 1024 multipoles while at 1530 multipole the power loss is of the order of 10%. Consequently it is important to take into account the APS analysis results up to $l_{max} = 2 \times nside$ because at higher multipoles results show a little but not negligible power loss. In the following analysis we compare the APS of rotated maps with the APS extracted by `anafast` from the original map, in order to avoid this error.

Obtained initial APS of maps used in this section, we changed coordinates with routine `rotation3.pro` without expansion and with an expansion to $nside$ = 4096 pixel. We re-analyzed APS of rotated maps and compared them with the initial APS. Results are shown in Figure 9: (a) absolute APS and (b) normalized to the initial APS.

From this figure we can observe two important properties of HEALPix tools. Although at pixel intensity level, expansion to higher resolution causes lower errors in the rotated maps, in harmonic space APS of rotated maps with expansion to higher resolution has higher power

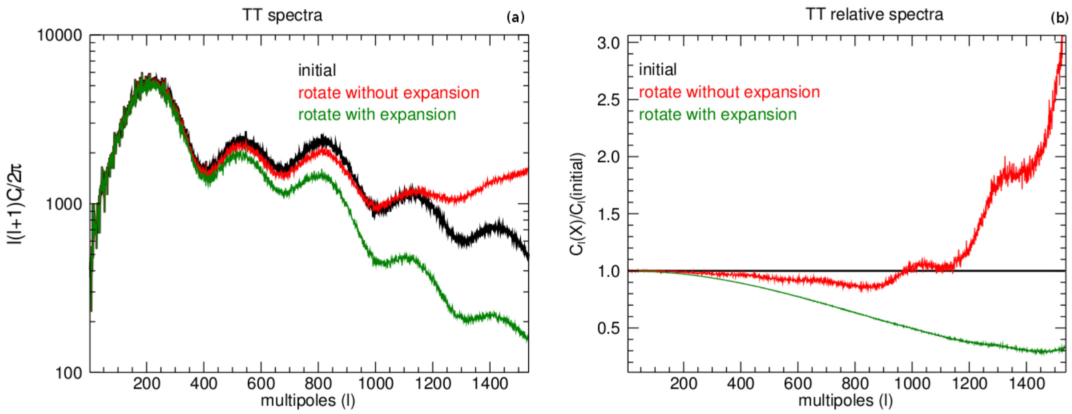


Figure 9: Comparison between APS of initial map and the rotated ones. (a) absolute APS, (b) normalized to the initial APS. The angular power spectra are corrected for the window function due to the map pixelization.

losses. In fact the power losses for the not expanded map are of the order of 10% at about $l = 800$ and for multipoles higher than 1024 there is an important increase in power due to the errors committed during rotation. In fact pixel intensity errors change the correlation between pixel on small scales. On the other hand for the map rotated with the expansion, a comparison between its APS and the initial one tells us that APS rotated after the map being expanded to higher resolution has power losses at 1024 multipole of the order of 50% yet. At $l = 1535$ power losses for the expanded map is more than 65%. Part of this effect can be explained by considering that, when the code downgrades the map from higher resolution to the original ones, it calculates the mean value of the pixel that fall in the same greater pixel of the downgraded map. This mean calculation can modify the correlation among pixel and cause the power loss observed in the APS. However the pixel method is optimized to work at pixel level, so we will consider the method with the addition of the expansion to higher resolution as the best one because it affects pixel intensities lower than the method without the expansion of the map.

3.2 "Harmonic" method

Rotation in harmonic space is executed by a `F90` program written by the writers and reported in Appendix B. With this language we can find and isolate all the routines that most contribute to the errors during the rotation. In the following sections we analyze every single step of the program, how it affects rotated maps, and finally we compare these results with those obtained in the previous section in which we used `HEALPIx IDL` routines.

3.2.1 `Fortran90` subroutines or facilities?

Firstly we analyse the capability of the program to extract coefficients a_{lm} of a spherical harmonics expansion of the map and to resynthesize the map from the same coefficients. No rotations are involved for the moment. This procedure is made by the `HEALPIx Fortran90` subroutines `map2alm` and `alm2map`. Such as with the `HEALPIx F90` facilities `anafast` and `synfast`, we can synthesize and analyse maps until a number of multipoles that are at maximum $l_{max} = 3 \times nside$. We generated two maps with resolution of $nside = 512$ but with a number of multipole of 1535 and 1024, respectively equal to three and two times the resolution of the map. We analysed the map with the `HEALPIx F90` subroutine `map2alm` to extract a_{lm} coefficients and then we resynthesize the map with `HEALPix F90` subroutine `alm2map`.

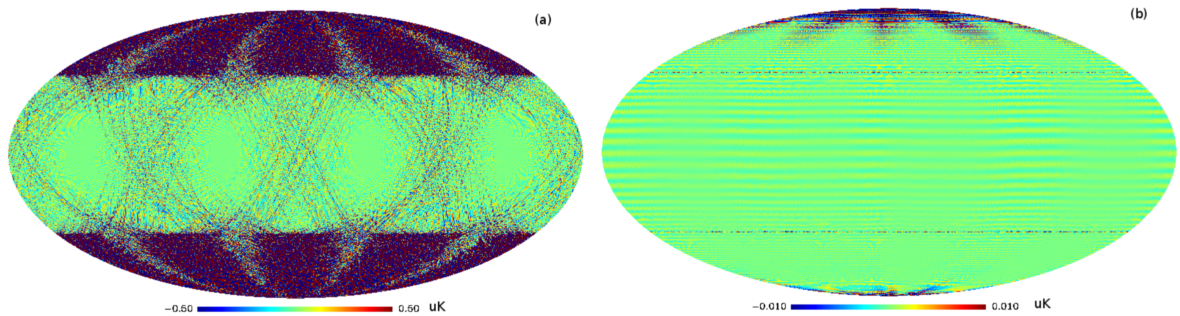


Figure 10: Distribution of the errors committed by the routines during the analysis of a CMB temperature map. A synthesized CMB map generated with `synfast` is analyzed by `HEALPix F90` subroutine `map2alm` to extract a_{lm} coefficients and then we resinthesize the map with `HEALPix F90` subroutine `alm2map`. We show the results from the subtraction of the resinthesized map and the original one. (a) Maps generated, read and wrote with a number of multipoles equal to three times the resolution of the map. (b) The same as (a), but with a number of multipoles only equal to two times the resolution of the map. Notice the different scales.

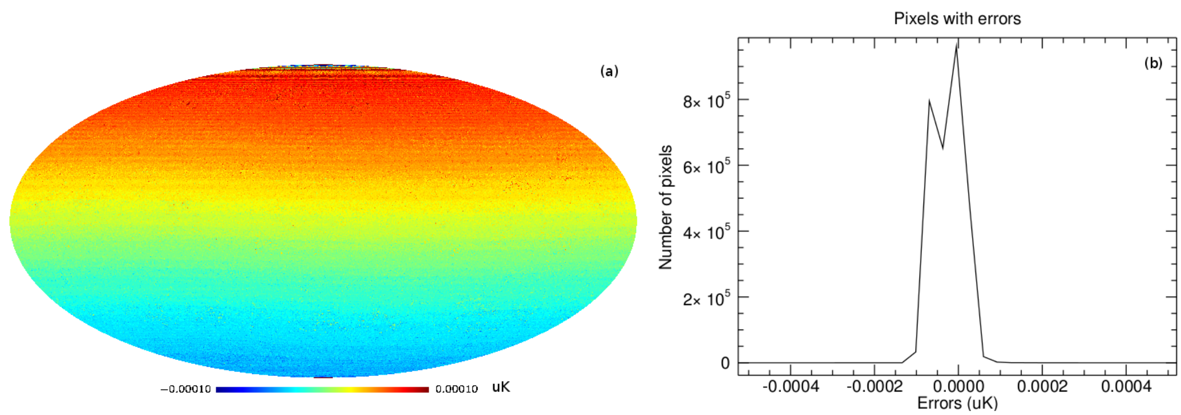


Figure 11: Errors committed by analysing CMB maps with `HEALPIx` facilities `anafast` and `synfast`. (a) Errors distribution, (b) their histogram. Compare them with the histograms of the errors showed in the previous figures.

In Figure 10 are shown the errors that affect the resynthesized map respect to the original map. We point out the fact that this extreme analysis allows us to reach a number of multipoles equal to three times the resolution of the maps but it affects the resynthesized map with important errors. However their values are really negligible compared to the sensitivity of current missions. In Figure 10 it is shown that by using multipoles only up to 2 times the resolution of the map, errors fall down more than one order of magnitude in respect to the other case. Considering these results about "harmonic method" and the power loss in APS of rotated maps with "pixel method" in the following analysis we take into account multipoles up to 1024 equal to 2 times the map resolution.

We made the same analysis with the `HEALPIx F90` facilities. By using `anafast` we analyzed the map until a number of multipoles equal to 1535 and by using `synfast` we resynthesized the map. In Figure 11 we show the errors committed by the facilities.

Errors that affect the maps are more than two orders of magnitude lower than errors committed by `HEALPIx F90` subroutines used previously. Consequently, it should be more useful to analyse and synthesize maps with `HEALPIx` facilities than with `HEALPIx` subroutines. There are subroutines that can read a_{lm} coefficients directly, but the output of these subroutines are in a different format than the ones that rotation subroutine needs. Otherwise the errors committed by the subroutines are at least three orders of magnitude lower than

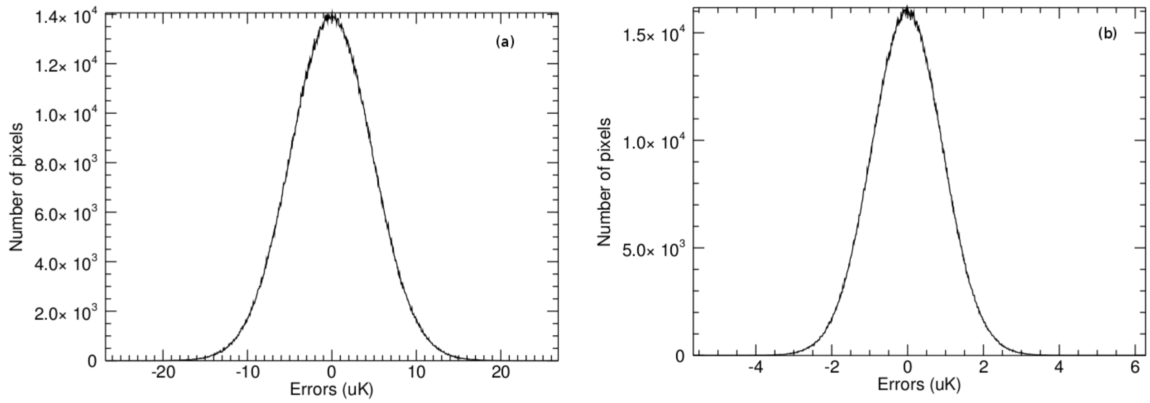


Figure 12: Plots of the errors affecting the rotated maps. These maps are obtained by prograding to (a) $nside = 1024$ and (b) $nside = 4096$ the original maps before analyzing them to recollect the a_{lm} coefficients.

the sensitivity of WMAP satellite and two orders of magnitude lower than the sensitivity of *Planck* satellite. For these reason we keep to use `HEALPix F90` subroutines.

3.2.2 Prograding and downgrading: effects on maps

The most important errors that affect maps comes from the prograding and downgrading procedures. Such as in the pixel space in which we increase the resolution of the maps before rotating them, we prograde the maps before analysing them to obtain a_{lm} coefficients.

Figure 12 shows that the subroutine that has to recollect a_{lm} coefficients from prograded maps affects them with important errors. These errors effectively decrease as the resolution increase, but at $nside = 4096$ they are two orders of magnitude higher than errors committed without prograding the maps as shown in Figure 10 (b). This part of the program affects maps so much that all the others sources of error become negligible. In Figure 12 are shown these errors for (a) a prograded map to $nside = 1024$ and (b) a prograded map to $nside = 4096$.

Another disadvantage of prograding and downgrading CMB maps is the computational time necessary to run the program because of the higher number of pixel from which subroutine has to obtain a_{lm} coefficients. As in the pixel space, computational time depends on the resolution of prograded maps. The higher is the resolution of prograded maps, the higher is the computational time. At the moment is clear that to prograde and downgrade maps needs a lot of computational time and also affect maps with great errors. Probably with computer of new generation or clusters, we could reach sufficiently higher resolution and prograding maps could be useful, but at the moment we will not expand the maps when we rotate them with the harmonic method.

3.3 Errors on pixelized maps and APS: comparisons between "pixel" and "harmonic" method

It is clear that the best combination of steps for rotating maps in harmonic method is to read the map, recover a_{lm} coefficients, rotate them and at the end regenerate the rotated map from the new a_{lm} coefficients.

Figure 13 shows the histograms of the errors that affect maps rotated twice with both methods: (a) the harmonic and (b) the pixel one.

As shown in figure harmonic method is very high-quality, affecting maps with errors of the order of 0.05μK when used on maps of only synthetic CMB, that is when noise due to foreground is zero. Pixel method doesn't reach similar results in this case. However both

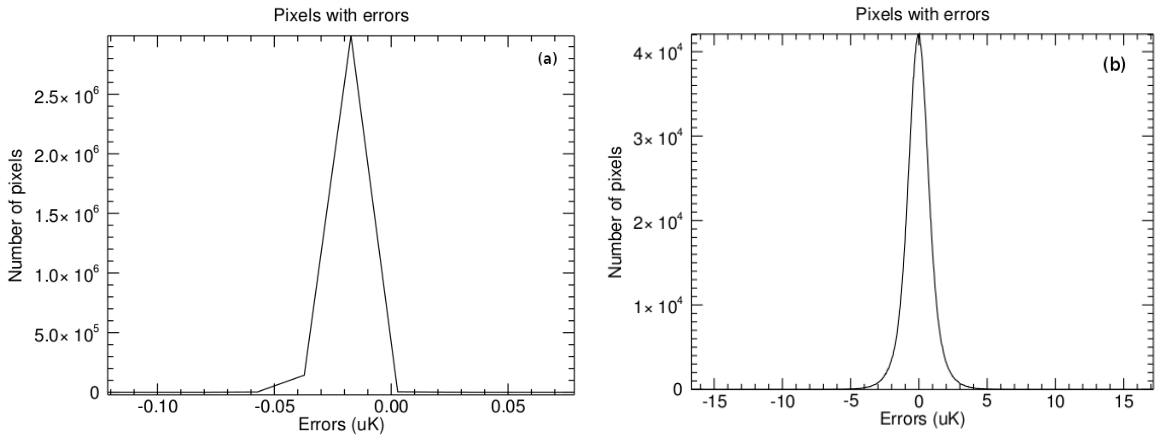


Figure 13: Plots of the errors affecting the rotated maps. These maps are obtained by rotating using (a) the harmonic method and (b) pixel method. In this last method we prograded map to $nside = 4096$.

the methods affect map with errors very much lower than sensitivity of WMAP and *Planck* missions.

In Figure 14 are shown APS of rotated maps by using both methods.

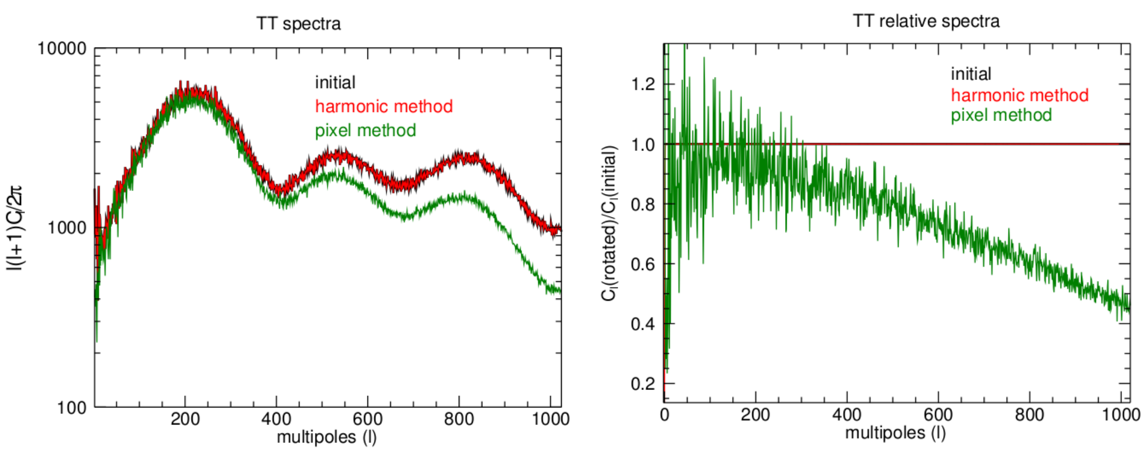


Figure 14: Plots of APS of rotated map by using both methods compared to the initial map: (a) absolute values, (b) relative to the initial APS. In the pixel method we prograded map to $nside = 4096$. APS are corrected for the window function due to the map pixelization.

As in pixel space, harmonic method is the most powerful in rotating maps, affecting APS of rotated maps with very low errors. Instead, there is a considerably power loss in the APS of map rotated with pixel method. As already shown in Figure 9 at 512 multipole it is about 15% but at 1024 multipole it is higher than 50%. The power lost with harmonic method is everywhere negligible compared with the pixel method.

3.4 CMB maps with beam

In the previous section change of coordinates has been done over maps without foreground obtained by the synthesis of the WMAP 1st year best fit of CMB APS. These maps were generated without a beam smoothing, that is with a beam of 0 arcmin of FWHM. Mission like WMAP or *Planck* generate maps observing the sky with instruments that present a beam. Synthesis can be done by using the `HEALPix F90` facility `synfast`. This routine allows us to generate maps with an arbitrary beam. By using the *Planck* beam we generate real maps that we will obtain from *Planck* satellite and over which we will work estimating the foreground.

According to [11] FWHM of the *Planck* Low Frequency Instrument at 44GHz is about 28 arcmin. Our maps were generated with a beam with FWHM of 20 arcmin. An example is shown in Figure 15 compared with a map generated without a beam (b).

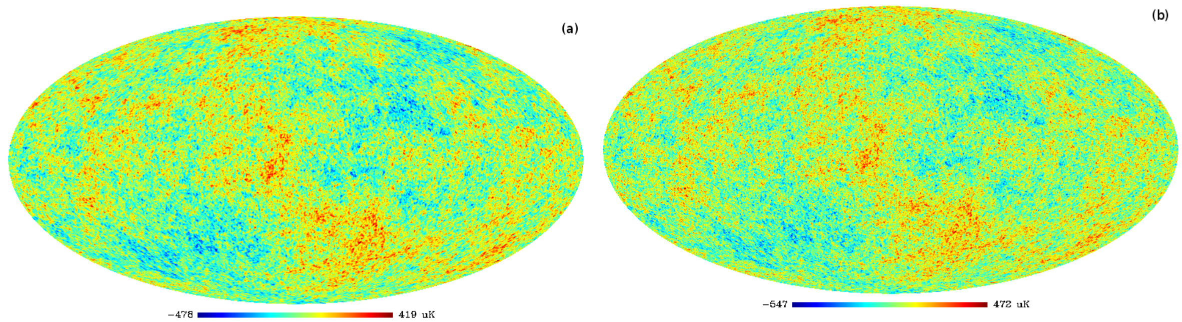


Figure 15: Examples of synthetic CMB maps generated by the **HEALPix F90** facility **synfast**: (a) with a beam with FWHM of 20 arcmin, (b) without a beam.

We applied the same procedure of the previous section. We generated a CMB temperature map with a beam of 20 arcmin of FWHM in Galactic coordinates. Then we rotated map in Ecliptic coordinates with both the methods, the pixel and the harmonic one. We analyse rotated map with `HEALPix F90` facility `anafast` obtaining their APS. Finally, we change again coordinates of the map back to the Galactic ones and compare its pixel values with the ones of the original map. The results are reported in Figures 16 and 17.

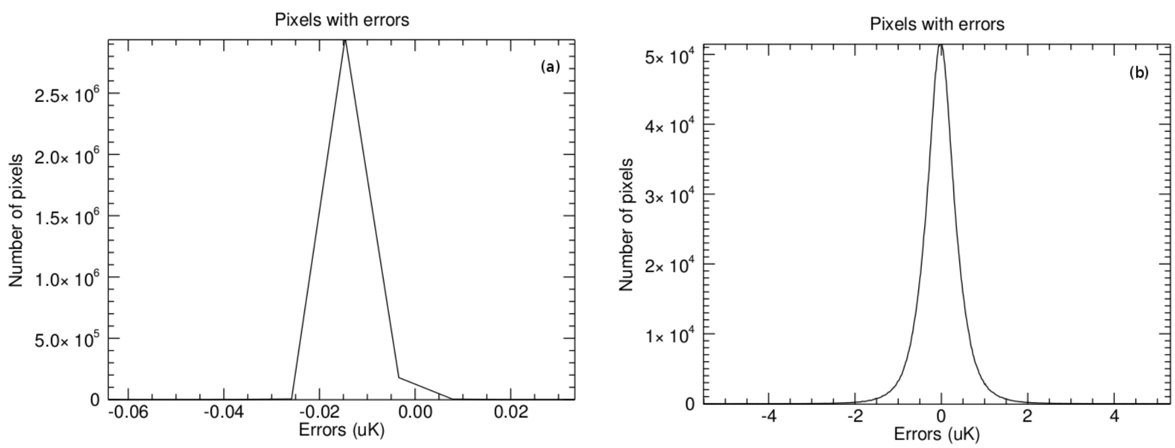


Figure 16: Histograms of the errors affecting the twice rotated maps in the presence of a non-zero beam (beam with FWHM of 20 degrees). These maps are obtained by rotation using (a) the harmonic method and (b) pixel method. In this last method we prograded map to $nside = 4096$ before the change of coordinates.

At pixel level, comparing Figure 13 and 16, we observe an quality improvement for both the methods that cut down errors by more than a half. This is probably due to the presence of the beam that smooths CMB temperature peaks. In fact when pixel method makes a mistake, it chooses a pixel near the correct one. It has a closer value respect to the case without beam as a consequence of beam smoothing. On the other hand, harmonic method doesn't find strong point peaks that make great problems in recovering a_{lm} coefficients. Such as in the case of no beam, harmonic method is still the most powerful in rotating and maintaining APS unchanged. In APS we observe a decrease in power loss for the pixel method while the harmonic method keeps its performance unchanged. The presence of the beam that smooths

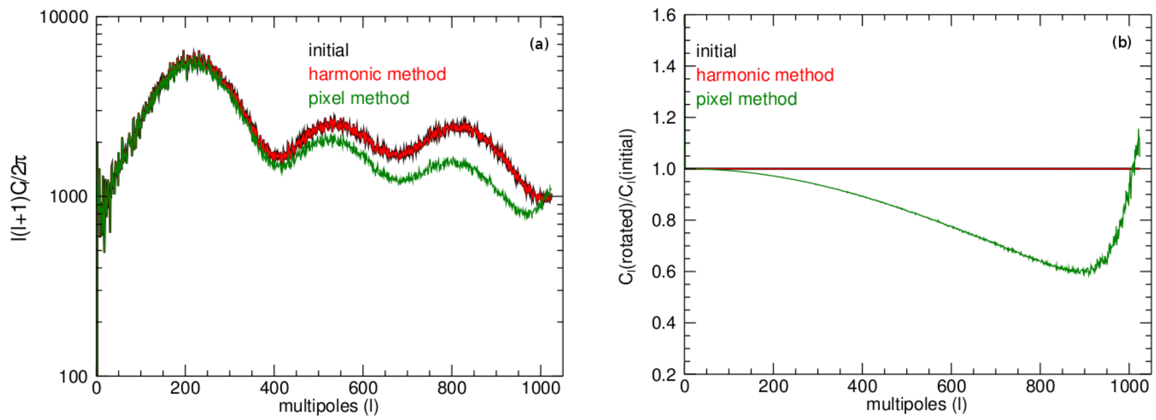


Figure 17: Plots of APS of rotated map by using both methods compared to the initial map: (a) absolute values, (b) relative to the initial APS. In the pixel method we prograded map to $nside = 4096$. APS are corrected for the window function due to the map pixelization and to the beam.

pixel intensities and causes a lower change in pixel correlations is the reason of the lower power loss for pixel method.

4 Results: application to WMAP maps

When an instrument observes the sky in the microwaves, it collects all signals coming from very different sources, not only the Cosmic Microwave Background, but also Galactic dense and cold cores, galaxy clusters that cause the Sunyaev-Zeldovich effect, and, most of all, our Galaxy with principally the synchrotron, dust and free-free emissions. CMB is only a component of the signal that we collect when observing the sky in microwaves. WMAP and *Planck* maps, generated by their instruments, have all these contributions. These are the real maps and probably the ones that is useful to rotate in different coordinates. For this reason, we apply both the `HEALPix` methods to change coordinates and report their performances on WMAP maps. Figure 18 shows patches of WMAP maps accessible from the LAMBDA website[5]. There are two different kind of maps: (a) the original map generated by the satellite and (b) the foreground reduced map in which a previous analysis has been done and most part of the non-CMB signal has been removed.

In this section we concentrate on foreground reduced maps. In the following section we analyze `HEALPix` performances on original (not-cleaned) WMAP maps.

4.1 Foreground reduced maps

As evident in Figure 18 (b), although part of the foreground has been removed, the presence of the Galaxy at low latitudes is still strong. For a correct analysis of the CMB is necessary the presence of a mask. The mask chosen for this map is a cut below 20 degrees of latitude, so about the 65% of the sky map is available for our analysis. We have two approaches available: first put a mask on the map in which the masked pixels have zero value and after rotate the map, otherwise first make the change of coordinates and only then apply the mask on the rotated map. The mask in Ecliptic coordinates is obtained by rotating the mask in Galactic coordinates with the pixel method prograding the map to $nside = 4096$.

We explored both the two approaches and found that there are no differences between them but for a lower loss of power in APS at high multipoles for the harmonic method when

[5]http://lambda.gsfc.nasa.gov/

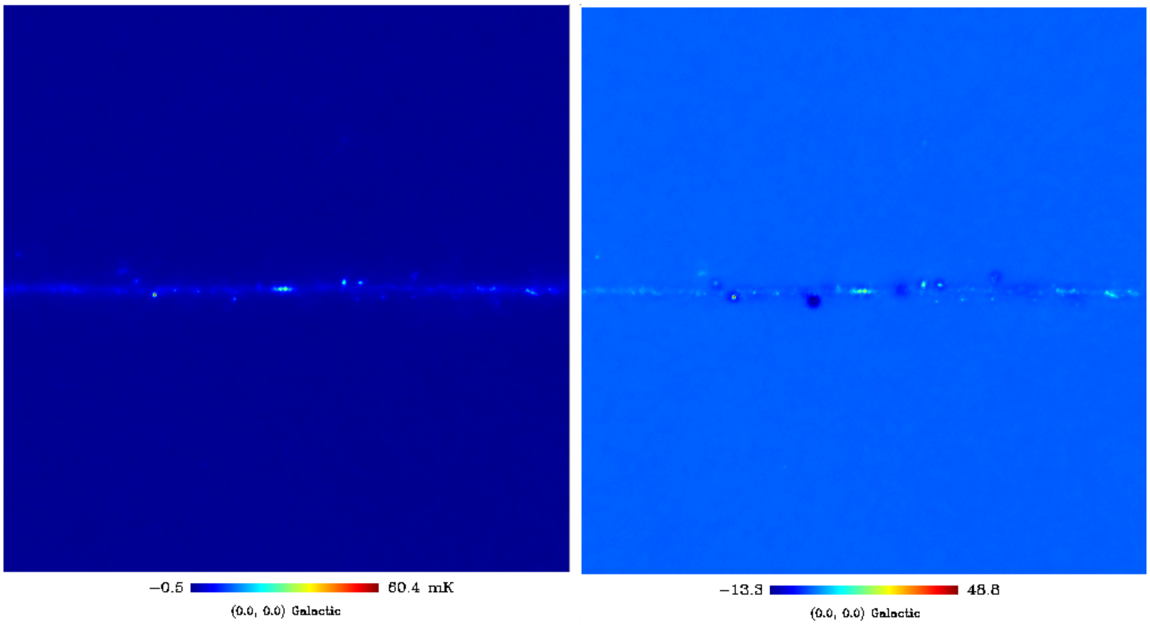


Figure 18: Examples of patches along the Galactic equator of sky maps generated by WMAP satellite. (a) Original map at 94 GHz, (b) the same of (a) but with mostly of the foreground removed.

the mask is applied before the rotation. In Figures 19, 20 and 21 we present results when the rotation of maps is done before the application of the mask. It is important to emphasize that the mask is applied only when we recover APS from the maps. When we extract the histograms of pixel errors no mask are used.

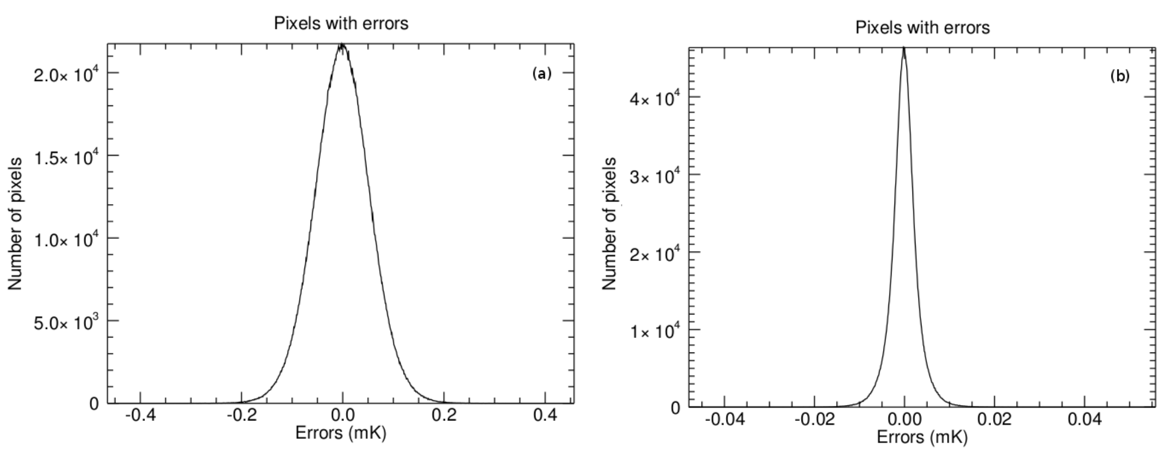


Figure 19: Plots of the errors affecting the rotated foreground reduced maps. These maps are obtained by rotating using (a) the harmonic method and (b) pixel method. In this last method we prograded map to $nside = 4096$.

In Figure 19 we show an important change in the performances of the method. Now it is the pixel method that less affects rotated maps. Its errors are an order of magnitude lower than the errors committed by the harmonic method. Our interpretation is that a_{lm} coefficients are more sensible to the presence of the foreground especially when there is a very strong Galactic emission along the Galactic equator. On the other hand, pixel method is not sensible to higher fluctuations in pixel values. Errors increase of one order of magnitude respect to the only CMB case because of the higher pixel values at low latitudes that pixel method come across when it chooses wrong pixels. Errors that affect rotated maps in pixel

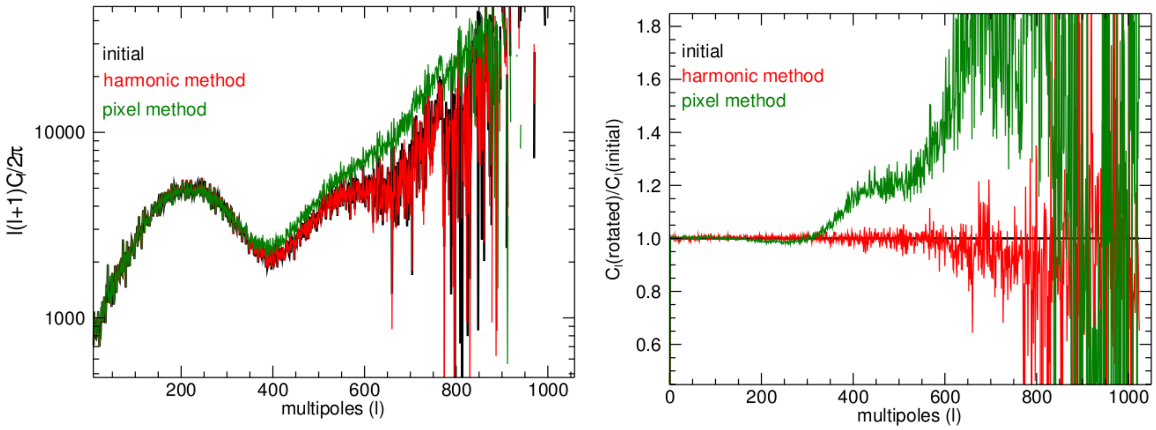


Figure 20: Plots of APS of rotated foreground reduced map by using both methods compared to the initial map: (a) absolute values, (b) relative to the initial APS. In the pixel method we prograded map to $nside = 4096$. APS are corrected for the window function due to the map pixelization and to the beam. The noise APS has been also removed.

method are lower than WMAP sensitivity, but similar to the *Planck* one.

Figure 20 shows the APS of the maps after the masking process. As already mentioned, mask is a cut at 20 degree of latitude so more than 65% of the sky is available to obtain the APS. In this way most part of the Galactic APS is cut away. We also performed a rough noise estimation by considering that it is a white-noise with a flat APS. At high multipoles noise is dominant so by calculating the mean of the C_l between 900 and 950 multipoles we estimated the flat noise APS and subtracted it from the C_l estimated by `anafast`. The mask and the noise estimation allow us to reach the first part of the second peak of CMB APS. In fact CMB is the most important contribution up to 600 multipole. At higher multipoles the presence of a residual noise completely covers CMB signal. Better available masks and noise estimates allow to reach higher multipoles, but it is beyond our aims. To the contrary of the comparison results at pixel level, harmonic method is still the most powerful to save the original APS of the map. Its errors at a multipole $\simeq 600$ is lower than 10%. The pixel method is able to reproduce APS of the map only up to a multipole of $\simeq 350$. At higher multipoles the errors rapidly increase and reach more than $\simeq 40\%$ at a multipole of $\simeq 600$.

4.2 Original maps

In the previous section we analyzed performences of `HEALPix` tools to change coordinates over foreground reduced maps, but often the analysis of foreground is the topic of a scientific work and it is in these cases that the change of coordinates is particularly important. For this reason we applied the same procedure of the previous sections to WMAP maps without any pre-clean of the foreground. An example is shown in Figure 18 (a). It is a map of the microwave temperature sky at 94 GHz obtained by WMAP satellite.

Such as for the foreground reduced maps the pixel method less affects rotated maps. Its errors is lower than 10 μK, similar to the *Planck* sensitivity so this errors will have to be taken into account in future analyses. Errors generated by the harmonic method show a little increase respect to the foreground reduce case. As in the previous case the problem is the higher signal fluctuations around the Galactic equator that cause higher errors.

In this last case is very useless analyse the map in the harmonic space by obtaining its APS. At high multipoles (higher than 500) the presence of noise is the most important contribution to APS, while at low multipoles (lower than 500) the presence of Galactic emission and all the other sources completely cover the CMB APS. However a preliminary analysis was done

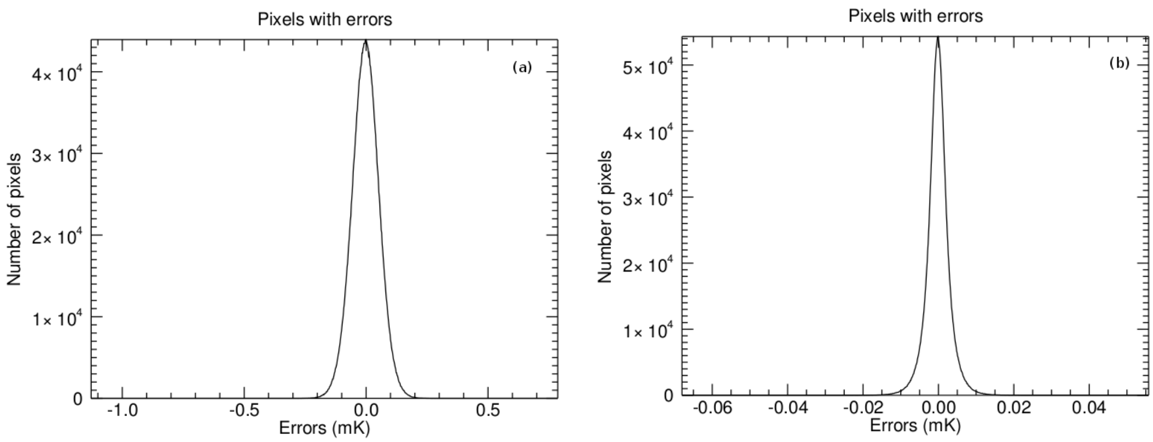


Figure 21: Plots of the errors affecting the rotated WMAP maps. These maps are obtained by rotating using (a) the harmonic method and (b) pixel method. In this last method we prograded map to $nside = 4096$.

and it confirms that also in this case maps rotated with the harmonic method have an APS closer to the initial one.

5 Conclusions

In this report we tested performances of `HEALPix` tools in changing from Galactic coordinates to Ecliptic ones all sky CMB temperature maps. We considered different cases. In section 2 we rotate only synthetic CMB maps generated by `HEALPix F90` facility `synfast` by using the WMAP 1st year best fit of CMB APS. In section 3 we considered real maps generated by WMAP satellite with the presence of foreground and with part of foreground reduced by previous analysis.

The `HEALPix` tool provides two different ways for changing coordinates. We called them pixel and harmonic method. The first make the change by rotating a set of 3D position vectors between astronomical coordinates systems by using Euler matrix. The second transforms the scalar a_{lm} coefficients to emulate the effect of an arbitrary rotation of the underlying map. In this work we compared these two methods. Comparison is done both in pixel and harmonic space. In pixel space we demonstrated that it is impossible to make a direct comparison of the two rotated maps, so we made the change of coordinates twice, returning to the original reference frame and calculating pixel per pixel the errors committed. In harmonic space we collected the APS of rotated maps and compared them with the APS of the original map.

Results obtained are very interesting demonstrating that there is not a better method, but it depends on the type of map that one wants to rotate and what type of characteristic of the map one wants to keep less affected by errors: pixels intensities or APS. In fact, considering the case of only CMB fluctuation maps, harmonic method is the best one affecting pixels with errors of the order of 0.05μK while power losses at 1024 multipole are less than 5% in APS. These errors are very low respect to the ones committed by pixel method that affects pixel with errors of the order of 3μK while power losses at 1024 multipole are about 50%. Things are more complicated if one wants to rotate maps with foreground added to CMB signal. In fact, our simulations on WMAP maps show that pixel method less affects maps, introducing errors of the order of 20μK, much smaller than those of 200μK obtained with the use of the harmonic method. On the other hand, harmonic method works better in APS recovery. It loses only 10% of the power at a multipole of $\simeq 600$, to be compared with pixel method that causes a power loss of 40% or more. It is important to stress that in all cases errors committed by methods are lower than the sensitivity of WMAP satellite maps and, only in real WMAP maps with foreground, rotation affects maps with errors similar to the sensitivity of the *Planck* satellite.

This quality difference for the two methods in pixel errors between the two cases studied is due to the presence of foreground that at lowest latitudes completely covers CMB signal. As demonstrated for strong point sources a_{lm} coefficients are not able to correctly reproduce high signal gradient. This problem has consequences also in the presence of foreground with high signal gradients while moving towards high latitudes. Harmonic method affects pixels near these high gradients with important errors that cause the consequence that this method is not the best for rotating CMB maps with foreground. So it is very important to choose the best method considering the scientific aims for which one needs to change map coordinates.

6 Appendix A: IDL procedures for rotation with the "pixel" method

6.1 Rotation routine: rotation2.pro

```
pro rotation2,MAP=map,COORD_IN=coord_in,COORD_OUT=coord_out,side

;Program for map rotation of only T full sky maps between astronomical coordinates.
;It uses the following HEALPix IDL subroutines:
; - rotate_coord
; - udgrade
;Maps ordering must be 'nested'

time=systime(/second)

read_fits_map,map,testmap

;Create vector to contain the rotated map
fine=size(testmap)
map_rotate=dblarr(fine(1))
map_rotate_n=reorder(map_rotate,/r2n)
nside=sqrt(fine(1)/12l)
fine2=12l*side^2

;Prograde the initial and final maps to higher resolution
ud_grade,testmap,testmap2,NSIDE_OUT=side,ORDER_IN='nest'
ud_grade,map_rotate_n,map_rotate_n2,NSIDE_OUT=side,ORDER_IN='nest'

;Rotate the prograded initial map
vector_in=dblarr(fine2,3)
vector_out=dblarr(fine2,3)
posi=indgen(fine2,/long)
posj=indgen(fine2,/long)

pix2vec_nest,side,posi,vector_in
vector_out=rotate_coord(vector_in,Inco=coord_out,Outco=coord_in)
vec2pix_nest,side,vector_out,posj
map_rotate_n2(posi)=testmap2(posj)

;Downgrade rotated maps to the original resolution
ud_grade,testmap2,testmap,NSIDE_OUT=nside,ORDER_IN='nest'
ud_grade,map_rotate_n2,map_rotate_n,NSIDE_OUT=nside,ORDER_IN='nest'

;Write the rotated map in a .fits file
write_fits_map,'rotate_n.fits',map_rotate_n,ordering='nest'

;A new rotation returning to the initial coordinates
;to estimate how this routine affects map pixel
testmap(*)=0.d0

;Prograde the final map to higher resolution
ud_grade,testmap,testmap2,NSIDE_OUT=side,ORDER_IN='nest'

;Rotate the map
pix2vec_nest,side,posi,vector_in
vector_out=rotate_coord(vector_in,Inco=coord_in,Outco=coord_out)
vec2pix_nest,side,vector_out,posj
testmap2(posi)=map_rotate_n2(posj)

;Downgrade rotated maps to the original resolution
ud_grade,testmap2,testmap,NSIDE_OUT=nside,ORDER_IN='nest'
```

```
;Write the rotated map in a .fits file
write_fits_map,'indietro_n.fits',testmap,coordsys=coord_in,ordering='nest'

time2=systime(/second)

print, 'Time to rotate the map = ',time2-time

end
```

6.2 Matrix creation: rotation_matrix.pro

```
pro rotation_matrix,nside,COORD_IN=coord_in,COORD_OUT=coord_out

time=systime(/second)

;Program for matrix generation for change coordinates
;from an astronomical coordinate system to another.
;The program generates also the inverted matrix
;for the change from the new coordinates to the old ones.
;It uses the following HEALPix IDL subroutines:
; - rotate_coord

;Generic map with the resolution requested
npixel=12l*nside^2
vector_in=fltarr(npixel,3)
vector_out=fltarr(npixel,3)
posi=indgen(npixel,/long)
posj=indgen(npixel,/long)
data=lonarr(npixel,2)

;Rotate the map to generate the matrix
pix2vec_nest,nside,posi,vector_in
vector_out=rotate_coord(vector_in,Inco=coord_out,Outco=coord_in)
vec2pix_nest,nside,vector_out,posj
data(*,0)=posi(*)
data(*,1)=posj(*)

;Save the matrix in an IDL file
filename='matrice'+STRTRIM(nside,1)+coord_in+'-'+coord_out
save,data,FILENAME=filename,/verbo

;Calculate the reverse matrix as well

;Rotate the map to generate the inverse matrix
pix2vec_nest,nside,posi,vector_in
vector_out=rotate_coord(vector_in,Inco=coord_in,Outco=coord_out)
vec2pix_nest,nside,vector_out,posj
data(*,0)=posi(*)
data(*,1)=posj(*)

;Save the matrix in an IDL file
filename='matrice'+STRTRIM(nside,1)+coord_out+'-'+coord_in
save,data,FILENAME=filename,/verbo

time2=systime(/second)

print, time2-time

end
```

6.3 Rotation with pregenerated matrixes: rotation3.pro

```
pro rotation3 ,MAP=map,COORD_IN=coord_in ,COORD_OUT=coord_out , side

;Program for rotating full sky maps from an astronomical coordinate to another.
;It uses a matrix to associate pixel of a coordinate to the pixel of the other.
;Matrix is generated by rotation_matrix.pro
;It uses the following HEALPix IDL subroutines:
; - udgrade
;Maps ordering must be 'nested'

time=systime(/second)

;PATH to the folder where is contained the matrix.
;There is a different matrix for every map resolutions.
PATH='/../'

read_fits_map ,map,testmap

nside=12l*side^2

;side points to the resolution of the prograded maps
;(in this case it's supposed to rotate from 'Galactic' coordinates to the 'Ecliptic' ones)
case side of
          512l: begin
                restore ,FILENAME=PATH+'matrice512G-E'
                rotazand=data
                restore ,FILENAME=PATH+'matrice512E-G'
                rotazrit=data
                end
          1024l:begin
                restore ,FILENAME=PATH+'matrice1024G-E'
                rotazand=data
                restore ,FILENAME=PATH+'matrice1024E-G'
                rotazrit=data
                end
          2048l:begin
                restore ,FILENAME=PATH+'matrice2048G-E'
                rotazand=data
                restore ,FILENAME=PATH+'matrice2048E-G'
                rotazrit=data
                end
          4096l:begin
                restore ,FILENAME=PATH+'matrice4096G-E'
                rotazand=data
                restore ,FILENAME=PATH+'matrice4096E-G'
                rotazrit=data
                end
endcase

fine=size(testmap)

;vector that will contain the map in the new coordinates
map_rotate=fltarr(fine(1))
map_rotate=reorder(map_rotate,/r2n)

nside=sqrt(fine(1)/12l)
fine2=12l*side^2

;prograde the initial and final maps to high resolution
ud_grade, testmap, testmap2 ,NSIDE_OUT=side ,ORDER_IN='nest'
ud_grade, map_rotate, map_rotate2 ,NSIDE_OUT=side ,ORDER_IN='nest'
```

```
;perform the rotation with the use of the matrix that
;associate pixel of a coordinate to the pixel of the other
map_rotate2(rotazand(*,0))=testmap2(rotazand(*,1))

;return to the original resolution
ud_grade,testmap2,testmap,NSIDE_OUT=nside,ORDER_IN='nest'
ud_grade,map_rotate2,map_rotate,NSIDE_OUT=nside,ORDER_IN='nest'

;Write the rotated map in a .fits file
write_fits_map,'rotate_n.fits',map_rotate(*,0),coordsys=coord_out,ordering='nest'

;A new rotation returning to the initial coordinates to estimate
;how this routine affect map pixel
testmap(*)=0.d0

;Prograde the final map to higher resolution
ud_grade,testmap,testmap2,NSIDE_OUT=side,ORDER_IN='nest'

;Rotate map to the original coordinates
testmap2(rotazrit(*,0))=map_rotate_n2(rotazrit(*,1))

;Downgrade rotated maps to the original resolution
ud_grade,testmap2,testmap,NSIDE_OUT=nside,ORDER_IN='nest'

;Write the rotated map in a .fits file
write_fits_map,'indietro_n.fits',testmap,coordsys=coord_in,ordering='nest'

time2=systime(/second)

print, 'Tempo impiegato = ',time2-time

end
```

7 Appendix B: Fortran program for rotation with "harmonic" method

7.1 The program: harmonic_rotation.f90

```
program harmonic_rotation

  !!This program reads HEALPix full sky maps,
  !! recovers its alm coefficients,
  !! rotate them in a new astronomical coordinate system,
  !! recreate a full sky map in the new coordinates by the rotated alm coefficients
  !! save the new map in a file
  !! Maps entering must have RING notation

  !!MODULES
  use healpix_types
  use fitstools, only: getsize_fits, input_map, output_map, dump_alms, number_of_alms
  use alm_tools, only: map2alm, alm2map, rotate_alm
  use coord_v_convert, only: coordsys2euler_zyz
  use udgrade_nr, only: udgrade_ring

  IMPLICIT NONE

  !!Variables definition
  integer(4) :: nside, nmaps, ordering, nlmax, nmmax, nside_ud, nlmax_ud, nmmax_ud
  integer(4) :: i,j
  integer(8) :: ntotpix, ntotpix_ud
```

```
real(4)      :: latitude, time_in,time_out,time_tot
real(8)      :: annoin,annofin, psi, theta, phi
real(8),dimension(2) :: zbound
real(8),dimension(:,:), allocatable :: map_in, map_in_ud, map_rotate_ud, map_rotate
real(8),dimension(:,:), allocatable :: map_out, map_out_ud, errori
real(8),dimension(:,:),allocatable :: weights
complex(8),dimension(:,:,:),allocatable :: alm_TGC
character(len=120) :: filename_in, filename_out_rotate, filename_out_back, filename_err
character(len=80) :: coord_in, coord_out
character(len=80),dimension(64) :: header_in, header_out
integer(4),dimension(8) :: ivalues


!!Initial variables value
!!PATH of the map that has to be rotated
filename_in='/.../map_512_1024_-12_20_r.fits'
!!PATH of file in which put the rotated maps
filename_out_rotate='/.../map_rotate.fits'
filename_out_back='/.../map_back.fits'
!!Resolution
nside_ud=512;
!!Eventually sky cut
latitude=90.0 !!degrees
zbound(1)=-sin(latitude*DEG2RAD)
zbound(2)=zbound(1)
yearin=2011.0
yearfin=2011.0
!!Astronomical coordinates
coord_in='G'
coord_out='E'

!!Other variables. Do not change.
ntotpix=getsize_fits(filename_in,nmaps=nmaps,ordering=ordering,nside=nside)
nlmax=2*nside
nmmax=nlmax
nlmax_ud=2*nside_ud
nmmax_ud=nlmax_ud
ntotpix_ud=12*(nside_ud)**2;

call date_and_time(values=ivalues)
time_in=ivalues(5)*3600+ivalues(6)*60+ivalues(7)+ivalues(8)/1000.0

allocate(map_in(0:ntotpix-1,1:nmaps))
allocate(map_in_ud(0:ntotpix_ud-1,1:nmaps))
allocate(map_rotate(0:ntotpix-1,1:nmaps))
allocate(map_rotate_ud(0:ntotpix_ud-1,1:nmaps))
allocate(map_out(0:ntotpix-1,1:nmaps))
allocate(map_out_ud(0:ntotpix_ud-1,1:nmaps))
allocate(alm_TGC(1:nmaps,0:nlmax_ud,0:nmmax_ud))
allocate(weights(1:2*nside_ud,1:nmaps))
allocate(errors(0:ntotpix-1,1:nmaps))

!!Read the initial map and eventually prograde it
call input_map(filename_in,map_in,ntotpix,nmaps,header=header_in)
if (nside .NE. nside_ud) then
      call udgrade_ring(map_in,nside,map_in_ud,nside_ud)
else
      map_in_ud=map_in
endif

!!Recover alm coefficients from the map
weights(:,:)=1   !!No weights
```

```fortran
  call alm_tools_mp_map2alm_sc_d(nside_ud,nlmax_ud,nmmax_ud,map_in_ud,alm_TGC,zbound,weights)

  !!Rotation of alm coefficients
  call coordsys2euler_zyz(yearin, yearfin,coord_in,coord_out,psi,theta,phi)
  call alm_tools_mp_rotate_alm_d(nlmax_ud,alm_TGC,psi,theta,phi)

  !!Map creation in the new coordinates system
  call alm_tools_mp_alm2map_sc_d(nside_ud,nlmax_ud,nmmax_ud,alm_TGC,map_rotate_ud)

  !!Eventually downgrade the map and write it in file
  if (nside .NE. nside_ud) then
        call udgrade_ring(map_rotate_ud,nside_ud,map_rotate,nside)
  else
        map_rotate=map_rotate_ud
  endif
  header_out=header_in
  call output_map(map_rotate,header_out,filename_out_rotate)

  !!A new rotation to the initial coordinates to estimate
  !!how this routine affects map pixel
  call coordsys2euler_zyz(annoin, annofin,coord_out,coord_in,psi,theta,phi)
  print*, shape(alm_TGC)
  call alm_tools_mp_rotate_alm_d(nlmax_ud,alm_TGC,psi,theta,phi)

  !!Map creation in the old coordinates system
  call alm_tools_mp_alm2map_sc_d(nside_ud,nlmax_ud,nmmax_ud,alm_TGC,map_out_ud)

  !!Eventually downgrade the map and write it in file
  if (nside .NE. nside_ud) then
        call udgrade_ring(map_out_ud,nside_ud,map_out,nside)
  else
        map_out=map_out_ud
  endif
  header_out=header_in
  call output_map(map_out,header_out,filename_out_indietro)

  !!Total time
  call date_and_time(values=ivalues)
  time_out=ivalues(5)*3600+ivalues(6)*60+ivalues(7)+ivalues(8)/1000.0

  time_tot=time_out-time_in
  print*,time_tot

  print*,'Finished'
end program harmonic_rotation
```

7.2 Makefile for compilation

```makefile
#Compiler
F90C = ifort

# Set up appropriate suffix list
.SUFFIXES: .o .f .for .f90 .c

# Flags for f95 o gfortran
HEALPIX_LIB=/.../Healpix_2.20a/lib
HEALPIX_INC=/.../Healpix_2.20a/include
CFITSIO=/usr/local/lib/cfitsio
FFLAGS  = -cm -w -vec_report0 -sox

# Link libraries and options
LDFLAGS= -lhealpix -lcfitsio
```

```
#Single suffix rules
.f90 .f .for :
        $(F90C) -O3 -I$(HEALPIX_INC) $(FFLAGS) -o $@ $< -I$(INCDIR)
            -I$(HEALPIX_INC) -L$(HEALPIX_LIB) -L$(CFITSIO) ${LDFLAGS}
```

Acknowledgements – We acknowledge support by PRIN MIUR 2009 grant n. 2009XZ54H2 and ASI through ASI/INAF Agreement I/072/09/0 for the *Planck* LFI Activity of Phase E2. Some of the results in this paper have been derived using the `HEALPix` package [4]

References

[1] Larson D., et. al., 2011, ApJS, 192, 16

[2] de Bernardis P., et al., 2000, Nature, 404, 955

[3] Górski K.M., Hivon E., Wandelt B.D., 1999, arXiv:astro-ph/9812350

[4] Górsky K.M., et al., APJ, 622, 759, 2005

[5] Calabretta M.R., Roukema B.F., 2007, MNRAS 381, 865

[6] Spergel D.N., et al., 2003, arXiv:astro-ph/0302209

[7] Jarosik N., et al., arXiv:1001.4744

[8] Mandolesi N., et al., 2011, in Advances in Plasma Astrophysics Proceedings, Giardini Naxos, Italy, 6-10 September 2010, IAU Symposium No. 274, A. Bonanno, E. de Gouveia dal Pino & A. Kosovichev, eds., p. 268

[9] Peebles P.J.E., Seager S., Hu W., 2000, ApJ, 539, L1

[10] Wyithe J.S.B., Cen R., 2007, ApJ, 659, 890

[11] *Planck* Collaboration, 2011, arXiv:1101.2022