Bull. Spec. Astrophys. Obs., 2005, 58, 40-50 © Special Astrophysical Observatory of the Russian AS, 2005

GLESP package for full sky CMB maps data analysis and its realization in the FADPS data processing system

O.V. Verkhodanov[a], **A.G. Doroshkevich**[b], **P.D. Naselsky**[c,d], **D.I. Novikov**[e], **V.I. Turchaninov**[f], **I.D. Novikov**[b,d,g], **P.R. Christensen**[d] **L.-Y.P. Chiang**[d]

[a] Special Astrophysical Observatory of the Russian AS, Nizhnij Arkhyz 369167, Russia,
[b] Astro Space Center of Lebedev Physical Institute, Profsoyuznaya 84/32, Moscow, Russia
[c] Rostov State University, Zorge 5, Rostov-Don, 344090, Russia
[d] Niels Bohr Institute, Blegdamsvej 17, DK-2100 Copenhagen, Denmark
[e] Imperial College, London, United Kingdom
[f] Keldysh Institute of Applied Math, Russian Academy of Science, 125047 Moscow, Russia
[g] University Observatory, Juliane Maries Vej 30, DK-2100, Copenhagen, Denmark

Received July 23, 2004; accepted September 10, 2004.

A new scheme of sky pixelization GLESP (Gauss–LEgendre Sky Pixelization) is developed for CMB maps. The scheme is based on the Gauss–Legendre polynomials zeros and allows one to create strict orthogonal expansion of the map. A corresponding code has been implemented and comparison with other methods has been done. The package has been realized using basic principles of the FADPS data reduction system. The structure and the main procedures of the package are described.

Key words: cosmology: cosmic microwave background – cosmology: observations – methods: data analysis

1. Introduction

The process of the cosmic microwave background (CMB) radiation data analysis contains several steps including

1) registration of time ordered data,
2) pixelization,
3) map — spherical harmonics transformation,
4) component separation,
5) statistics analysis,
6) $C(\ell)$–spectrum calculation and
7) cosmological parameters estimation.

In this paper we consider steps (2), (3) and (6).

Starting from the COBE experiment using the so-called Quadrilateralized Sky Cube Projection (see Chan and O'Neill 1976, O'Neill and Laubscher 1976, Greisen and Calabretta 1993), the problem of the whole sky CMB pixelization has attracted great interest. At least three methods of the CMB celestial sphere pixelization have been proposed and implemented after the COBE pixelization scheme: the Icosahedron pixelizing by Tegmark (1996), the Igloo pixelization by Crittenden and Turok (1998, hereafter CT98) and the HEALPix[1] method by Górski et al. (1999) (with the last modification in 2003). Two important questions mentioned already by Tegmark (1996) are now under discussion: a) what is the optimal method for the choice of the N_{pix} positions of pixel centers, shapes and sizes to provide (as good as possible) the compact uniform coverage of the sky by pixels with equal areas, and b) what is the best way to approximate any convolutions of the maps by sums using pixels ?

All the above mentioned pixelization schemes were devoted to solving the first problem as accurate as possible, and the answer to the second question usually follows for the chosen pixelization scheme.

In this paper we change the focus of the problem to processing on the sphere and then determine the scheme of pixelization. We would like to remind that pixelization of the CMB data on the sphere is only some part of the general problem, which is the determination of the coefficients of the spherical harmonic decomposition of the CMB signal for both anisotropy and polarization. These coefficients, which we call a_{lm}, are used in subsequent steps in the analysis of

[1] currently http://www.eso.org/science/healpix/

the measured signal, and in particular, in the determination of the power spectra, $C(\ell)$, of the anisotropy and polarization (see review in Hivon et al. 2002), in some special methods for components separation (Stolyarov et al. 2002; Naselsky et al. 2003a) and phase statistics (Chiang et al. 2003, Naselsky et al. 2003a,b; Naselsky et al., 2004; Coles et al. 2004).

Here we propose a specific method to calculate the coefficients a_{lm}. It is based on the so-called Gaussian quadratures and is presented in Sec. 2 . In this specific pixelization scheme corresponds the position of pixel centers along the θ–coordinate to so–called the Gauss – Legendre quadrature zeros and it will be shown (Sec. 5) that this method increases the accuracy of calculations essentially.

Thus, the method of calculation of the coefficients a_{lm} dictates the method of the pixelization. We call our method GLESP, the Gauss–Legendre Sky Pixelization (Doroshkevich et al. 2003; Verkhodanov et al. 2003, 2004). We have developed a special code for the GLESP approach and a package of codes which are necessary for the whole investigation of the CMB data including the determination of anisotropy and polarization power spectra, C_ℓ, the Minkowski functionals and other statistics.

This paper is devoted to description of the main idea of the GLESP method, the estimation of the accuracy of the different steps and of the final results, the description of the GLESP code and its testing. We do not discuss the problem of integration over a finite pixel size for the time ordered data in this paper. The simplest scheme of integration over pixel area is to use equivalent weight relatively to the center of the pixel. The GLESP code uses this method as HEALPix and Igloo do.

Also, we described the basic procedures of the package and their interactions. The package is realized using principles of the flexible data processing system (FADPS) operating at the RATAN-600 (Verkhodanov et al. 1993; Verkhodanov 1997a).

2. Main ideas and basic relations

The standard decomposition of the measured temperature variations on the sky, $\Delta T(\theta,\phi)$, in spherical harmonics is

$$\Delta T(\theta,\phi) = \sum_{\ell=2}^{\infty}\sum_{m=-\ell}^{m=\ell} a_{\ell m} Y_{\ell m}(\theta,\phi)\,, \qquad (1)$$

$$Y_{\ell m}(\theta,\phi) = \sqrt{\frac{(2\ell+1)}{4\pi}\frac{(\ell-m)!}{(\ell+m)!}} P_\ell^m(x) e^{im\phi}\,, \qquad (2)$$

$$x = \cos\theta\,,$$

where $P_\ell^m(x)$ are the associated Legendre polynomials. For a continuous $\Delta T(x,\phi)$ function, the coefficients of decomposition, $a_{\ell m}$, are

$$a_{\ell m} = \int_{-1}^{1} dx \int_{0}^{2\pi} d\phi \Delta T(x,\phi) Y^*_{\ell m}(x,\phi)\,, \qquad (3)$$

where $Y^*_{\ell m}$ denotes complex conjugation of $Y_{\ell m}$. For numerical evaluation of the integral Eq.(??) we will use the Gaussian quadratures, a method which was proposed by Gauss in 1814, and developed later by Christoffel in 1877. As the integral over x in Eq.(??) is an integral over a polynomial of x, we may use the following equality (Press et al. 1992):

$$\int_{-1}^{1} dx \Delta T(x,\phi) Y^*_{\ell m}(x,\phi) =$$

$$= \sum_{j=1}^{N} w_j \Delta T(x_j,\phi) Y^*_{lm}(x_j,\phi)\,, \qquad (4)$$

where w_j is a proper Gaussian quadrature weighting function. Here the weighting function $w_j = w(x_j)$ and $\Delta T(x_j,\phi)Y^*_{\ell m}(x_j,\phi)$ are taken at points x_j which are the net of roots of the Legendre polynomial

$$P_N(x_j) = 0\,, \qquad (5)$$

where N is the maximal rank of the polynomial under consideration. It is well known that the equation (5) has N number of zeros in interval $-1 \le x \le 1$. For the Gaussian–Legendre method Eq.(??), the weighting coefficients are (Press et al. 1992)

$$w_j = \frac{2}{1-x_j^2}[P_N'(x_j)]^{-2}\,, \qquad (6)$$

where $'$ denotes a derivative. They can be calculated together with the set of x_j with the *"gauleg"* code (Press et al. 1992, Sec. 4.5).

In the GLESP approach are the trapezoidal pixels bordered by θ and ϕ coordinate lines with the pixel centers (in the θ direction) situated at points with $x_j = \cos\theta_j$. Thus, the interval $-1 \le x \le 1$ is covered by N rings of the pixels (details are given in Sec. 3). The angular resolution achieved in the measurement of the CMB data determines the upper limit of summation in Eq. (??), $\ell \le \ell_{max}$. To avoid the Nyquist restrictions we use a number of pixel rings, $N \ge 2\ell_{max}$. In order to make the pixels in the equatorial ring (along the ϕ coordinate) nearly squared, the number of pixels in this direction should be $N_\phi^{max} \approx 2N$. The number of pixels in other rings, N_ϕ^j, must be determined from the condition of making the pixel sizes as equal as possible with the equatorial ring of pixels.

Fig. **??** shows the weighting coefficients, w_j, and the position of pixel centers for the case $N = 31$. Fig.

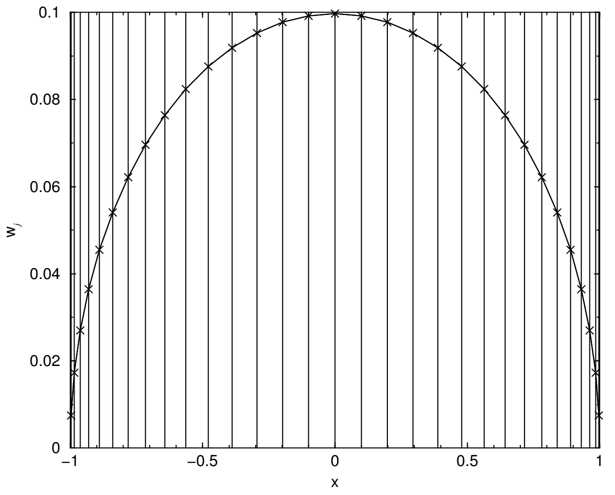

Figure 1: *Gauss-Legendre weighting coefficients (w_j) versus Legendre polynomial zeros ($x_j = \cos\theta_j$) being centers of rings used in GLESP for the case of $N = 31$. Positions of zeros are plotted by vertical lines.*

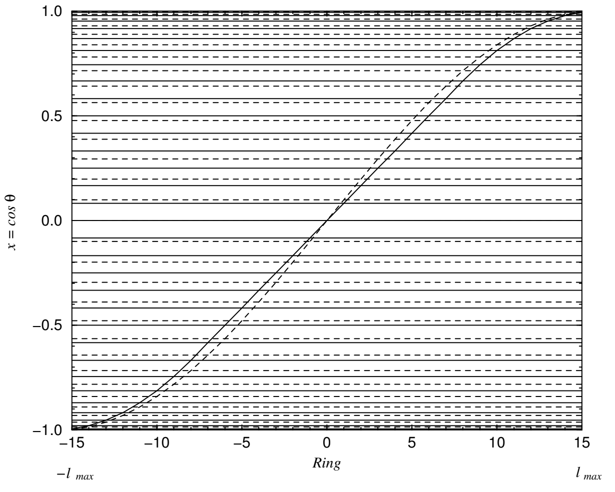

Figure 2: *Ring center position ($x_j = \cos\theta_j$) vs ring number for 2 pixelization schemes, HEALPix (solid) and GLESP (dashed). Figure demonstrates the case of $N = 31$.*

?? compares some features of the pixelization schemes used in HEALPix and GLESP (see Sec. 4). Fig. ?? compares pixel shapes and distribution on a sphere in a full sky Mollweide projections of HEALPix and GLESP maps.

In the definition (??) are the coefficients $a_{\ell m}$ complex quantities while ΔT is real. In the GLESP code started from the definition (??) we use the following representation of the ΔT

$$\Delta T(\theta,\phi) = \sum_{\ell=2}^{\ell_{max}} a_{\ell 0} Y_{\ell 0}(\theta,\phi) +$$

$$+ \sum_{\ell=2}^{\ell_{max}} \sum_{m=1}^{\ell} \left(a_{\ell m} Y_{\ell m}(\theta,\phi) + a_{l,-m} Y_{\ell,-m}(\theta,\phi)\right) , \qquad (7)$$

where

$$Y_{l,-m}(\theta,\phi) = (-1)^m Y^*_{l,m}(\theta,\phi), a_{\ell m} = (-1)^m a^*_{l,-m} \,.(8)$$

Thus,

$$\Delta T(\theta,\phi) = \frac{1}{\sqrt{2\pi}} \sum_{\ell=2}^{\ell_{max}} Re(a_{\ell,0}) P_\ell^0(\cos\theta) +$$

$$+ \sqrt{\frac{2}{\pi}} \sum_{\ell=2}^{\ell_{max}} \sum_{m=1}^{l} \sqrt{\frac{2\ell+1}{2}\frac{(\ell-m)!}{(\ell+m)!}} P_\ell^m(\cos\theta) \times$$

$$\times \left[Re(a_{\ell m})\cos(m\phi) - Im(a_{\ell m})\sin(m\phi)\right] , \qquad (9)$$

where $P_\ell^m(\cos\theta)$ are the well known associated Legendre polynomials (see Gradshteyn and Ryzhik 2000). In the GLESP code, we use normalized associated Legendre polynomials f_ℓ^m:

$$f_\ell^m(x) = \sqrt{\frac{2\ell+1}{2}\frac{(\ell-m)!}{(\ell+m)!}} P_\ell^m(x) , \qquad (10)$$

where $x = \cos\theta$, and θ is the polar angle. These polynomials, $f_\ell^m(x)$, can be calculated using two well known recurrence relations. The first of them gives $f_\ell^m(x)$ for a given m and all $\ell > m$:

$$f_\ell^m(x) = x\sqrt{\frac{4\ell^2-1}{\ell^2-m^2}} f_{\ell-1}^m -$$

$$-\sqrt{\frac{2\ell+1}{2\ell-3}\frac{(\ell-1)^2-m^2}{\ell^2-m^2}} f_{\ell-2}^m \,. \qquad (11)$$

This relation starts with

$$f_m^m(x) = \frac{(-1)^m}{\sqrt{2}} \sqrt{\frac{(2m+1)!!}{(2m-1)!!}} (1-x^2)^{m/2},$$

$$f_{m+1}^m = x\sqrt{2m+3} f_m^m \,.$$

The second recurrence relation gives $f_\ell^m(x)$ for a given ℓ and all $m \le l$:

$$\sqrt{(\ell-m-1)(\ell+m+2)} f_\ell^{m+2}(x) +$$

$$+ \frac{2x(m+1)}{\sqrt{1-x^2}} f_\ell^{m+1}(x) +$$

$$+ \sqrt{(\ell-m)(\ell+m+1)} f_\ell^m(x) = 0 \,. \qquad (12)$$

This relation is started with the same $f_\ell^\ell(x)$ and $f_\ell^0(x)$ which must be found with (??).

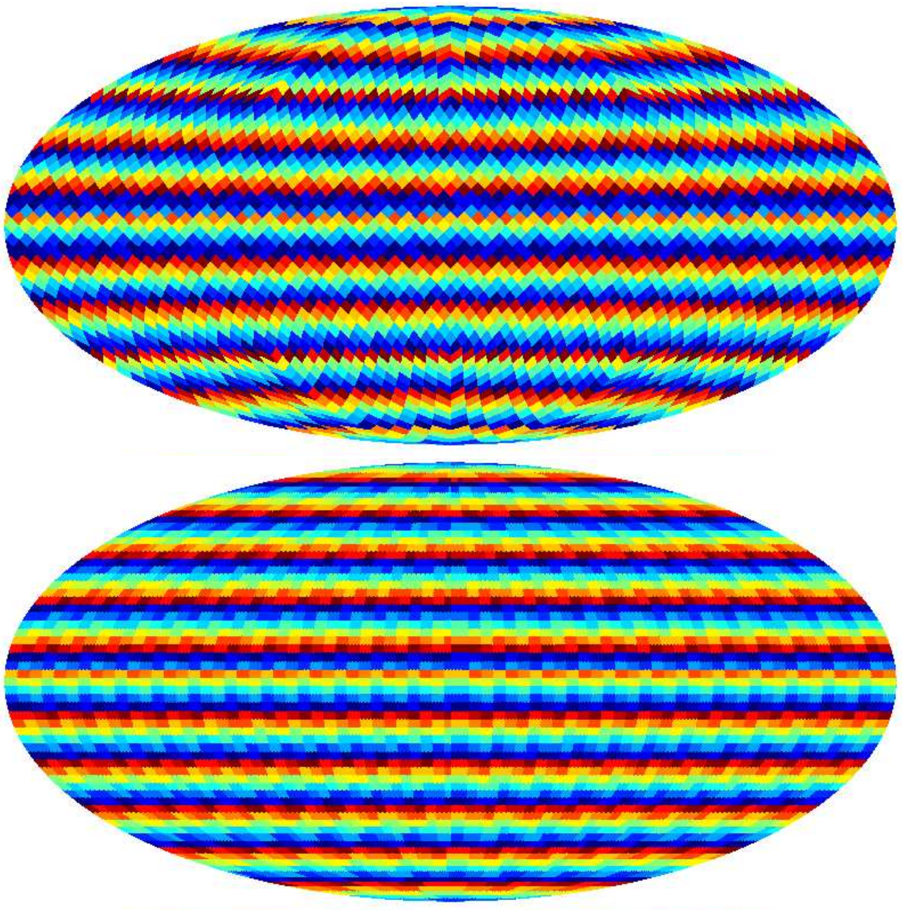

Figure 3: *Schematic representation of 2 types of pixelization on sphere: HEALPix (top) and GLESP (bottom). Various color of pixels is used to show their shape.*

As is discussed in Press et al. (1992, Sec. 5.5), the first recurrence relation (**??**) is formally unstable if the number of iteration tends to infinity. Unfortunately, there are no theoretical recommendations what the maximum iteration one can use in the quasi-stability area. However, it can be used because we are interested in the so-called *dominant* solution (Press et al. 1992, Sec. 5.5), which is approximately stable. The second recurrence relation (**??**) is stable for all ℓ and m.

3. Properties of GLESP

Following the previous discussion we define the new pixelization scheme GLESP as follows:

- In the polar direction $x = cos\theta$, we define $x_j, j = 1, 2, .., N$, as the net of roots of Eq. (**??**).
- Each root x_j determines the position of a ring with N_ϕ^j pixel centers with ϕ–coordinates ϕ_i.
- All the pixels have nearly equal area.
- Each pixel has weight w_j (see Eq. (**??**)).

In our numerical code which realizes the GLESP pixelization scheme we use the following conditions.

- Borders of all pixels are along the coordinate lines of θ and ϕ. Thus with a reasonable accuracy they are trapezoidal.
- The number of pixels along the azimuthal direction ϕ depends on the ring number. The code allows an arbitrary number of these pixels to be choosen. The number of pixels depends on the ℓ_{max} accepted for the CMB data reduction.
- To satisfy the Nyquist's theorem, the number N of the ring along the $x = \cos(\theta)$ axis must be taken as $N \geq 2\ell_{max} + 1$.
- To make equatorial pixels roughly square, the number of pixels along the azimuthal axis, ϕ, is taken as $N_\phi^{max} = \mathrm{int}(2\pi/d\theta_k + 0.5)$, where $k = \mathrm{int}(N+1)/2$, and $d\theta_k = 0.5(\theta_{k+1} - \theta_{k-1})$.
- The nominal size of each pixel is defined as $S_{pixel} = d\theta_k \times d\phi$, where $d\theta_k$ is the value on the equatorial ring and $d\phi = 2\pi/N_\phi^{mx}$ on equator.
- The number N_ϕ^j of pixels in the j^{th} ring at $x =$

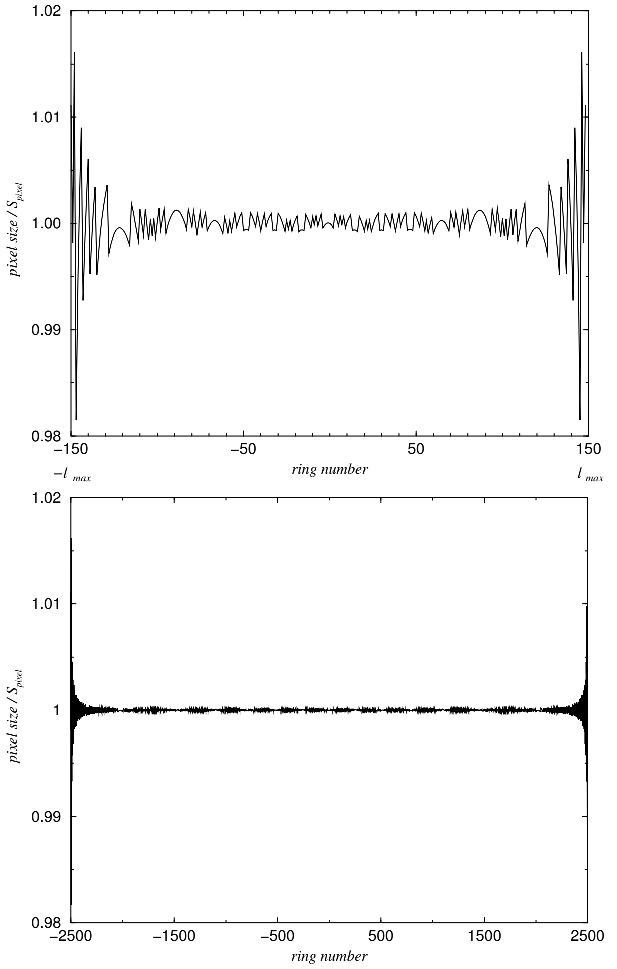

Figure 4: *Pixel size/equator pixel area vs ring-number for GLESP for number of rings $N = 300$ and $N = 5000$.*

x_j is calculated as $N_\phi^j = \mathrm{int}(2\pi\sqrt{1-x_j^2}/S_{pixel}+0.5)$;

- Polar pixels are triangular.
- Because the number N_ϕ^j differs from 2^k where k is integer, we use for the Fast Fourier transformation along the azimuthal direction the FFTW code (Frigo and Johnson 1997). This code permits one to use not only 2^n–approach, but other base-numbers too, and provide even higher speed.

With this scheme, the pixel sizes are equal inside each ring, and with a maximum deviation between the different rings of ∼1.5% close to the poles (Fig **??**). Increasing resolution decreases an absolute error of an area because of the in-equivalence of polar and equator pixels proportionally to N^{-2}.

Fig. **??** shows that this pixelization scheme for high resolution maps (e.g. $\ell_{max} > 500$) produces nearly equal thickness $d\theta$ for most rings.

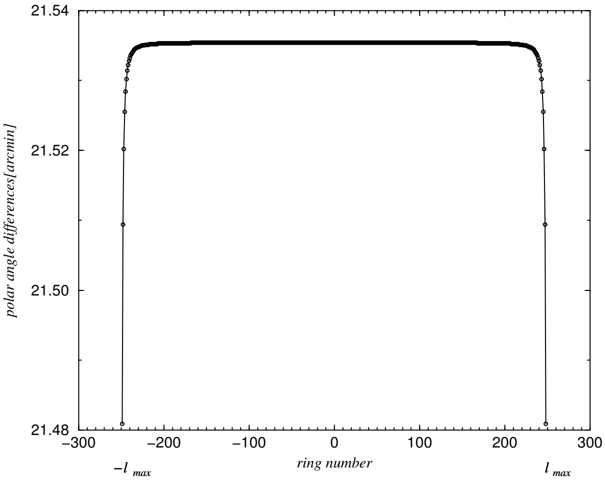

Figure 5: *Pixel size along polar angle ($\ell_{max} = 250$).*

GLESP has not the hierarchical structure, but the problem of the closest pixel selection is on the software level. Despite GLESP is close to the Igloo pixelization scheme in the azimuthal approach, there is a difference between the two schemes in connection with the θ-angle (latitude) pixel step selection. Therefore, we can not unify these two pixelizations. The Igloo scheme applied to the GLESP latitude step will give too different pixel areas. The pixels will be neither equally spaced in latitude, nor of uniform area, like Igloo requires.

4. GLESP pixel window function

For application of the GLESP scheme, we have to take into account the influence of the pixel size, shape and its location on the sphere on the signal in the pixel and its contribution to the power spectrum $C(\ell)$. The temperature in a pixel is (Górski et al. 1999; CT98)

$$\Delta T_p = \int_{\Delta\Omega_p} W_p(\theta,\phi)\Delta T(\theta,\phi)d\Omega\,, \qquad (13)$$

where $W_p(\theta,\phi)$ is the window function for the p-th pixel with the area $\Delta\Omega_p$. For the window function $W_p(\theta,\phi) = 1$ inside the pixel and $W_p(\theta,\phi) = 0$ outside (Górski et al. 1999), we have from Eq.(**??**) and Eq.(**??**):

$$\Delta T_p = \sum_{\ell,m} a_{\ell m} W_p(\ell,m)\,,$$

where

$$W_p(\ell,m) = \int d\Omega W_p(\theta,\phi) Y_{\ell m}(\theta,\phi)$$

and

$$W_p(\theta,\phi) = \sum_{\ell,m} W_p(\ell,m) Y^*_{\ell m}(\theta,\phi). \qquad (14)$$

The corresponding correlation function (CT98) for the pixelized signal is

$$\langle \Delta T_p \Delta T_q \rangle = \sum_{l,m} C(\ell) W_p(\ell,m) W_q^*(\ell,m)\,. \qquad (15)$$

4.1. Accuracy of the window function estimation

The discreteness of the pixelized map determines the properties of the signal for any pixels and restricts the precision achieved in any pixelization scheme. To estimate this precision, we can use the expansion (CT98)

$$\Delta T^{map}(\theta,\phi) = \sum_p S_p \Delta T_p W_p(\theta,\phi) = \\ = \sum_{\ell,m} a_{\ell m}^{map} Y_{\ell m}(\theta,\phi)\,, \qquad (16)$$

$$a_{\ell m}^{map} = \int d\Omega \Delta T^{map}(\theta,\phi) Y_{\ell m}^*(\theta,\phi) = \\ = \sum_p S_p \Delta T_p W_p^*(\ell,m)\,, \qquad (17)$$

where S_p is the area of the p-th pixel. These relations generalize Eq. (**??**), taking properties of the window function into account. The GLESP scheme uses the properties of Gauss–Legendre integration in the polar direction while azimuthal pixelization for each ring is similar to the Igloo scheme, and we get (see Eq.(4)):

$$W_p(\ell,m) = \frac{w_p}{\sqrt{2\pi}\Delta x_p} \exp\left(\frac{im\pi}{N_\phi^p}\right) \frac{\sin\left(\pi m/N_\phi^p\right)}{\left(\pi m/N_\phi^p\right)} \times \\ \times \int_{x_p-0.5\Delta x_p}^{x_p+0.5\Delta x_p} dx f_\ell^m(x)\,, \qquad (18)$$

where $\Delta x_p = (x_{p+1} - x_{p-1})/2$ with x_p the p-th Gauss–Legendre knot and N_ϕ^p the number of pixels in the azimuthal direction. This integral can be rewritten as follows:

$$\int_{x_p-0.5\Delta x_p}^{x_p+0.5\Delta x_p} dx f_\ell^m(x) \simeq \\ \simeq \sum_{k=0}^{\infty} \frac{1+(-1)^k}{(k+1)!} {f^{(k)}}_\ell^m(x_p) \left(\frac{\Delta x_p}{2}\right)^{k+1}\,, \qquad (19)$$

where ${f^{(k)}}_\ell^m(x_p)$ denotes the k-th derivatives at $x = x_p$. So, for $\Delta x_p \ll 1$ we get the expansion of (**??**):

$$W_p^{(2)}(\ell,m) = W_p^{(0)}(\ell,m)\left(1 + \frac{f_\ell^{(2)m}(\Delta x_p)^2}{24 f_\ell^m}\right)\,, \qquad (20)$$

where $W_p^{(0)}(\ell,m) \simeq$

$$\simeq \frac{w_p}{\sqrt{2\pi}} \exp\left(\frac{im\pi}{N_\phi^p}\right) \frac{\sin\left(\pi m/N_\phi^p\right)}{\left(\pi m/N_\phi^p\right)} f_\ell^m(x_p)\,, \qquad (21)$$

where $W_p^0(\ell,m)$ is independent of Δx_p. For the accuracy of this estimate we get

$$\frac{\delta W_p(\ell,m)}{W_p(\ell,m)} = \frac{W_p^{(2)}(\ell,m) - W_p^{(0)}(\ell,m)}{W_p^{(0)}(\ell,m)} \simeq \\ \simeq \left|\frac{(f'')_\ell^m(\Delta x_p)^2}{24 f_\ell^m}\right|\,. \qquad (22)$$

According to the last modification of the HEALPix, an accuracy of the window function reproduction is about 10^{-3}. To obtain the same accuracy for the $W_p(\ell,m)$, we need to have

$$\Delta x_p \le 0.15 \left|\left|\frac{f_\ell^m}{(f'')_\ell^m}\right|^{\frac{1}{2}}\right|_{x=x_p}\,. \qquad (23)$$

Using the approximate link between Legendre and Bessel functions for large ℓ (Gradshteyn and Ryzhik 2000) $f_\ell^m \propto J_m(\ell x)$ we get:

$$\Delta x_p \le 0.15 x_p/\sqrt{m(m+1)}\,, \qquad (24)$$

and for $\Delta x_p \sim \pi/N$ we have from Eq.(**??**)

$$\frac{\delta W_p(\ell,m)}{W_p(\ell,m)} \ge 10^{-2} \cdot \left(\frac{\ell_{max}}{N}\right)^2\,. \qquad (25)$$

For example, for $N = 2\ell_{max}$, we obtain $\delta W_p(\ell,m)/W_p(\ell,m) \simeq 2.3 \cdot 10^{-3}$, what is a quite reasonable accuracy for $\ell_{max} \sim 3000$–6000.

5. Structure of the GLESP code

The code is developed in two levels of organization. The first one, which unifies F77 FORTRAN and C functions, subroutines and wrappers for C routines to be used for FORTRAN calls, consists of the main procedures: '*signal*', which transforms given values of $a_{\ell m}$ to a map, '*alm*', which transforms a map to $a_{\ell m}$, '*cl2alm*', which creates a sample of $a_{\ell m}$ coefficients for a given C_ℓ and '*alm2cl*', which calculates C_ℓ for $a_{\ell m}$. Procedures for code testing, parameters control, Kolmogorov-Smirnov analysis for Gaussianity of $a_{\ell m}$ and homogeneity of phase distribution, and others, are also included. Operation of these routines is based on a block of procedures calculating the Gauss–Legendre pixelization for a given resolution parameter, transformation of angles to pixel numbers and back.

The second level of the package contains the programs which are convenient for the utilization of the

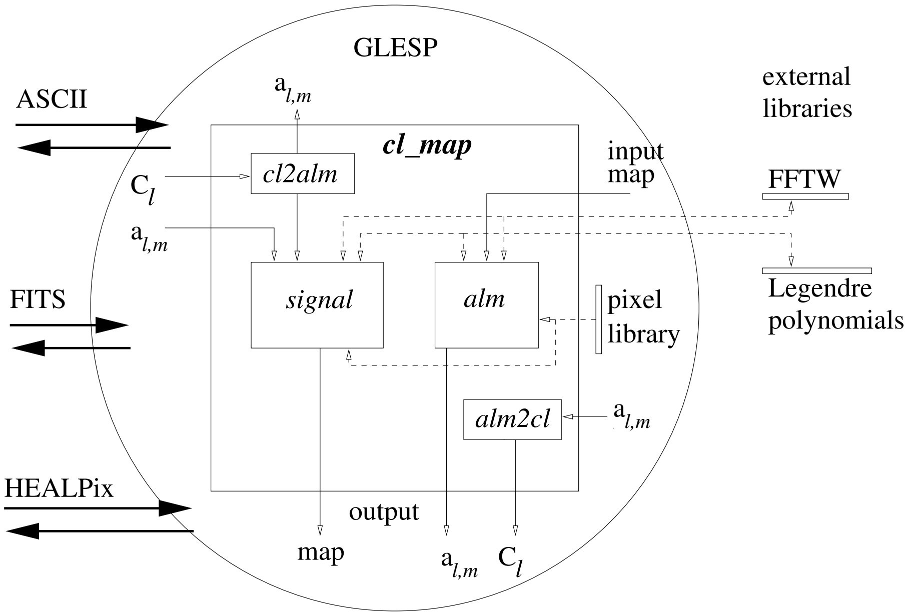

Figure 6: *Structure of the GLESP package.*

first level routines. In addition to the straight use of the already mentioned four main procedures, they also provide means to calculate map patterns generated by the Y_{20}, Y_{21} and Y_{22} spherical functions, to compare two sets of $a_{\ell m}$ coefficients, to convert a GLESP map to a HEALPix map, to convert a HEALPix map, or other maps, to a GLESP map.

Fig. **??** outlines the GLESP package. The circle defines the zone of the GLESP influence based on the pixelization library. It can include several subroutines and operating programs. The basic program *'cl_map'* of the second level, shown as a big rectangle, interacts with the first level subroutines. These subroutines are shown by small rectangles and call external libraries for the Fourier transform and Legendre polynomial calculations. The package reads and writes data both in ASCII table and FITS formats. More than 10 programs of the GLESP package operate in the GLESP zone.

The present development of the package has also parallel calculation implementation. Visualization procedures in OPEN GL have been developed at IaO, Cambridge.

5.1. Basic organization principles of the GLESP package

The package is realized using the FADPS ideology (Verkhodanov et al. 1993, Verkhodanov 1997). It satisfies to the following principles:

- Each program is designed to be easily joint with other modules of a package. It operates both with a given file and standard output.
- Each program can operate separately.
- Each program is accessible in a command string with external parameters. It has a dialogue mode and could be tuned with a resource file in some cases.
- Output format of resulting data is organized in the standard way and is prepared in the F-format (Verkhodanov & Kononov 2002) or ASCII table accessible for other packages.
- The package programs can interact with other FADPS procedures and CATS database.

5.2. Main operations

There are four types of operations accessible in the GLESP package:

- Operations related to maps:

1. Spherical harmonic decompostion of a map into $a_{\ell m}$ (*cl2map*).
2. Smooth a map with a Gaussian beam (*cl2map*).
3. Sum/difference/averaging between maps (*difmap*).
4. Scalar multipication/division (*difmap*).
5. Map rotation (*difmap*).
6. Conversion from Galactic to equatorial coordinates (*difmap*).
7. Cut temperature values in a map (*mapcut*).
8. Cut a zone in/from a map (*mapcut*).

9. Cut out cross-sections from a map (*mapcut*).
10. Produce simple patterns (*mappat*).
11. Read ASCII into binary (*mappat*).
12. Read point sources to binary map (*mappat*).
13. Print values in map (*mapcut*).
14. Find min/max values in map sample per pixel (*difmap*).
15. Simple statistic on a map (*difmap*).
16. Correlation coefficients of two maps (*difmap*).
17. Pixel size on a map (*ntot*).
18. Plot figures (*f2fig*).

- Operations related to $a_{\ell m}$:

1. Synthesise the map from given $a_{\ell m}$ (*cl2map*).
2. Sum/difference (*difalm*).
3. Scalar multipication/division (*difalm*).
4. Vector multipication/division (*difalm*).
5. Add phase to all harmonics (*difalm*).
6. Cut out given mode of harmonics (*difalm*).
7. Calculate angular power spectrum C_ℓ (*alm2dl*).
8. Calculate phases (*alm2dl*).
9. Select the harmonics with a given phase (*alm2dl*).
10. Compare two $a_{\ell m}$ samples (*checkalm*).
11. Produce $a_{\ell m}$ of map derivatives (*dalm*).

- Operations related to angular power spectrum C_ℓ :

1. Calculate power spectrum C_ℓ (*alm2dl*).
2. Simulate a map by a given C_ℓ (*cl2map*).
3. Simulate $a_{\ell m}$ by C_ℓ (*createalm*).

- Operations related to phases $\phi_{\ell m}$ and amplitudes $|a_{\ell m}|$:

1. Calculate phases $\phi_{\ell m}$ (*alm2dl*).
2. Calculate amplitudes $|a_{\ell m}|$ (*alm2dl*).
3. Simulate $a_{\ell m}$ by phases (*createalm*).
4. Select harmonics with a given phase (*alm2dl*).
5. Add a phase to all harmonics (*difalm*).

5.3. Main programs

The following procedures organized as separate programs in the pixel and harmonics domain are realized now:

alm2dl calculates spectra and phases by $a_{\ell m}$–coefficients.

checkalm compares different $a_{\ell m}$–samples.

cmap converts HEALPix format maps to the GLESP package format.

cl2map converts a map to $a_{\ell m}$–coefficients and $a_{\ell m}$–coefficients to a map, simulates a map by a given C_ℓ–spectrum.

createalm creates $a_{\ell m}$–coefficients by phases, amplitudes or/and C_ℓ–spectrum.

dalm calculates the 1-st and 2-nd derivatives by $a_{\ell m}$–coefficients

difalm calculates arithmetic operations over $a_{\ell m}$–samples.

difmap calculates arithmetic operations over maps, produces coordinates transformations.

f2fig produces color pictures in GIF–images.

f2map converts a GLESP map to a HEALPix format map.

fitstof converts the HEALPix $a_{\ell m}$–coefficients data format to the GLESP format $a_{\ell m}$–data.

ftofits converts the GLESP format $a_{\ell m}$–data to the HEALPix $a_{\ell m}$–coefficients data format.

mapcut cuts amplitude and coordinates in a GLESP map, produces one-dimensional cross-sections from a given maps allowing one to simulate RATAN–600 data for CMB study (Parijskij 2001).

mappat produces standard map patterns, reads ASCII data to produce a map, reads point sources position from ASCII files including CATS data (Verkhodanov et al. 1997b).

psep makes phase analysis separation (Naselsky et al. 2003a) for two input $a_{\ell m}$–samples.

5.4. Data format

The GLESP data are represented in two formats describing $a_{\ell m}$–coefficients and maps.

$a_{\ell m}$–coefficients data contains index describing number of ℓ and m modes corresponding to the HEALPix, real and imaginary parts of $a_{\ell m}$. These three parameters are described by three–fields records of the Binary Table of the F–format (Verkhodanov & Kononov 2002).

Map data are described by the three–fields Binary Table F–format containing a vector of $x_i = \cos\theta$ positions, a vector of numbers of pixels per each layer N_{ϕ_i}, and set of temperature values in each pixel recorded by layers from the North Pole.

6. Test and precision of the GLESP code

Three tests allow us to check the code. The first of them is from the analytical maps

$$Y_{2,0} = \sqrt{\frac{5}{16\pi}}(3x^2 - 1)\,,$$

$$Y_{2,1} = -\sqrt{\frac{15}{8\pi}}x\sqrt{1-x^2}\cos\phi,$$

$$Y_{2,-1} = -\sqrt{\frac{15}{8\pi}}x\sqrt{1-x^2}\sin\phi\,,$$

$$Y_{2,2} = \sqrt{\frac{15}{32\pi}}(1-x^2)\cos(2\phi),$$

$$Y_{2,-2} = -\sqrt{\frac{15}{32\pi}}(1-x^2)\sin(2\phi)\,.$$

to calculate $a_{\ell m}$. The code reproduces the theoretical $a_{\ell m}$ better than 10^{-7}.

The second test is to reproduce an analytical map $\Delta T(x,\phi) = Y_{\ell m}(x,\phi)$ from a given a_{lm}. These tests check the calculations of the map and spherical coefficients independently.

The third test is the reconstruction of $a_{\ell m}$ after the calculations of the map, $\Delta T(x,\phi)$, and back. This test allows one to check orthogonality. If the transformation is based on really orthogonal functions it has to return after forward and backward calculation the same $a_{\ell m}$ values.

Precision of the code can be estimated by introduction of a set of $a_{\ell m} = 1$ and reconstruction of them. This test showed that using relation (??) we can reconstruct the introduced $a_{\ell m}$ with a precision $\sim 10^{-7}$ limited only by single precision of float point data recording and with a precision $\sim 10^{-5}$ for relation (??).

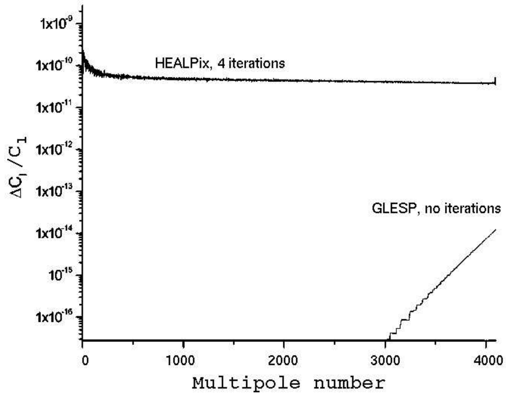

Figure 7: *Comparison of calculation accuracy in HEALPix (4 iterations) of the version 1.20 and in GLESP (no iterations) methods. Number of pixels is approximately the same ($\sim 6\times10^7$) and calculation time is proportional to the number of iterations.*

Fig. ?? demonstrates the accuracy of C_ℓ calculations using HEALPix and GLESP[2]

It should be noted that unlike the HEALPix code, the GLESP method does not need any iteration for calculation of the $a_{\ell m}$ coefficients and therefore is much faster. Our definition of the $a_{\ell m}$ coefficients is exactly the same as in HEALPix as an estimator of the anisotropy power spectrum:

$$C(\ell) = \frac{1}{2\ell+1}\left[|a_{\ell 0}|^2 + 2\sum_{m=1}^{\ell}|a_{\ell m}|^2\right]. \tag{26}$$

7. Re-pixelization

To transfer a sky distribution map from one pixel grid (e.g. from HEALPix to GLESP) to another one, we should use one of 2 ways:

1) to calculate $a_{\ell m}$–coefficients and after that to restore a map in a new grid;

2) or to use re-pixelization procedures on the current brightness distribution.

Any re-pixelization procedure will cause loss of information and thereby introduce uncertainties and errors. The GLESP code has procedures for map re-pixelization based on two different methods in the $\Delta T(\theta,\phi)$–domain: the first one consists in averaging input values in the corresponding pixel, the second one is connected with spline interpolation inside the pixel grid.

In the first method, we consider input pixels which fell in our pixel with values $\Delta T(\theta_i,\phi_i)$ to be averaged with a weighting function. The realized weighting function is a function of simple averaging with equal weights. This method is widely used in appropriation of a given values to the corresponding pixel number.

In the second method of re-pixelization, we use a spline interpolation approach. If we have a map $\Delta T(\theta_i,\phi_i)$ recorded in the knots different from the Gauss–Legendre grid, it is possible to repixelize it to our grid $\Delta T(\theta'_i,\phi'_i)$ using approximately the same number of pixels and the standard interpolation scheme based on the cubic spline approach for the map re-pixelization. This approach is sufficiently fast because the spline is calculated once for one vector of the tabulated data (e.g. in one ring), and values of interpolated function for any input argument are obtained by one call of separate routine (see routines "*spline*" to calculate second derivatives of interpolating function and "*splint*" to return a cubic spline interpolated value in Press et al. (1992)).

[2] Calculations were carried out and Fig.?? was produced by Vlad Stolyarov at IaO, Cambridge.

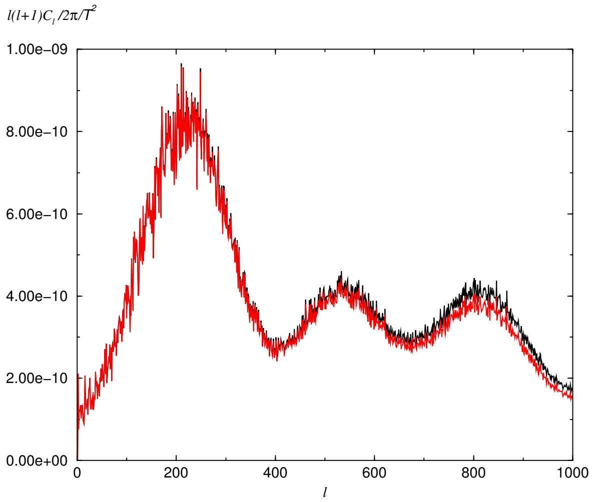

Figure 8: *Power spectra calculated for the initial HEALPix map (upper curve) with ℓ_{max} = 1000, N_{side} = 1024, pixel size = 11.8026□′, and N_{tot}=12 582 912, and for resulting re-pixelized GLESP map (lower curve) with the closest possible pixel size = 11.8038□′, N_{tot}=12 581 579. Deviations of the power spectra at high ℓ illustrate the ratio of the HEALPix and GLESP window functions.*

Our spline interpolation consists of the three steps:

- we set equidistant knots by the ϕ–axis to reproduce an equidistant grid;
- we change the grid by $x = cos(\theta)$–axis to the required GLESP grid;
- after that, we recalculate ϕ-knots to the rings corresponding to the GLESP x-points.

Fig. **??** demonstrates the deviation of accuracy of the power spectrum in a case of re-pixelization from a HEALPix map to a GLESP map with the same resolution. As one can see, for the range $\ell \leq \ell_{max}/2$, re-pixelization reproduces correctly all properties of the power spectra. For $\ell \geq \ell_{max}/2$ some additional investigations need to be done to take into account the pixel-window function. This work is in progress.

8. Summary

We suggest a new scheme "GLESP" for sky pixelization based on the Gauss–Legendre quadrature zeros. It has strict expansion by the orthogonal functions which gives accuracy for $a_{\ell m}$–coefficients calculations below 10^{-7} without any iterations. We realized two approaches for Legendre polynomials calculation using L– and M–methods of calculation schemes.

Among the main advantages of this scheme are

- a high accuracy in calculation $a_{\ell m}$,
- a high speed because of no iterations,
- an optimal selection of resolution for a given beam size, which means an optimal number of pixels and a pixel size.

A corresponding code has been designed in FORTRAN 77 and C languages for procedures of the CMB sky map analysis.

The $a_{\ell m}$ calculation is the main goal. $a_{\ell m}$-s are used in component separation methods and tests for non-Gaussianity (Chiang et al. 2003, Naselsky et al. 2003a,b). It is oriented on the fast and accurate calculation of the $a_{\ell m}$ for the given resolution specified by the beam size. Using accurately calculated $a_{\ell m}$-s, one can reproduce any pixelization scheme by the given pixel centers: GLESP, HEALPix, Igloo or Icosahedron.

Acknowledgements. This paper was supported in part by Danmark Grundforskningsfond through its support for the establishment of the Theoretical Astrophysics Center. Authors are thankful to Vladislav Stolyarov for testing parallel capabilities and OPEN GL visualization tool for current pre-release version of GLESP. OVV thanks the RFBR for partial support of the work through its grant 02–07–90038. Some of the results in this paper have been derived using the HEALPix (Górski et al. 1999) package.

References

Chan F.K., O'Neil E.M., 1976, Feasibility study of a quadrilateralized spherical cube Earth data base, Computer Sciences Corp., EPRF Technical Report

Chiang L.-Y., Naselsky P.D., Verkhodanov O.V., Way M.J, 2003, ApJ, **590**, L65 (astro-ph/0303643)

Coles P., Dineen P., Earl J., Wright D., 2004, MNRAS, **350**, 989 (astro-ph/0310252)

Crittenden R.G., Turok N.G., 1998, Exactly Azimuthal Pixelizations of the Sky, Report-no: DAMTP-1998-78 (astro-ph/9806374) (CT98)

Doroshkevich A.G., Naselsky P.D., Verkhodanov O.V., Novikov D.I., Turchaninov V.I., Novikov I.D., Christensen P.R., Chiang L.-Y., 2003, accepted to Internat. J. Modern Phys. D, **14**, No. 7 (astro-ph/0305537)

Frigo M., Johnson S.G., 1997, The Fastest Fourier Transform in the West, Technical Report MIT-LCS-TR-728 (http://www.fftw.org)

Górski K.M., Hivon E., & Wandelt B.D., 1999, in "Evolution of Large–Scale Structure: from Recombination to Garching" (http://www.eso.org/science/healpix)

Gradsteyn I.S., Ryzhik I.M., 2000, Tables of Intergrals, Series and Products, Sixth Edition, Academic Press

Greisen E.W., Calabretta M., 1993, Bull. American Astron. Soc., **182**, 09.01

Hivon, E., Górski, K.M., Netterfield, C.B., Crill, B.P., Prunet, S., & Hansen, F., 2002, ApJ, **567**, 2

Naselsky P.D., Verkhodanov O.V., Chiang L.-Y., Novikov I.D., 2003a, ApJ(submitted) (astro-ph/0310235)

Naselsky P.D., Doroshkevich A.G., Verkhodanov O.V., 2003b, ApJ, **599**, L53 (astro-ph/0310542)

Naselsky P.D., Doroshkevich A.G., Verkhodanov O.V., 2004, MNRAS, **349**, 695 (astro-ph/0310601)

O'Neil E.M., Laubscher R.E., 1976, Extended studies of a quadrilateralized spherical cube Earth data base, Computer Sciences Corp., EPRF Technical Report

Parijskij Yu N. 2001. Current Topics in Astrofundamental Physics: the Cosmic Microwave Background. Proceedings of the NATO Advanced Study Institute, held December 5–16, 1999, in Erice, Etta Majorana Centre, Italy. Edited by Norma G. Sanchez. Published by Kluwer Academic Publishers, P. O. Box 17, 3300 AA Dordrecht, ISBN 0-7923-6855-X, 219

Press W.H., Teukolsky S.A., Vetterling W.T., Flannery B.P., 1992, Numerical Recipes in FORTRAN, Second Edition, Cambridge University Press (http://www.nr.com)

Stolyarov V., Hobson M.P., Ashdown M.A.J., Lasenby A.N., 2002, MNRAS, **336**, 97

Tegmark M., 1996, ApJ, **470**, L81

Verkhodanov O.V., Erukhimov B.L., Monosov M.L., Chernenkov V.N., Shergin V.S., 1993, Astrofiz. Issled. (Izv. SAO), **36**, 132

Verkhodanov O.V., 1997, in: "Astronomical Data Analysis Software and Systems VI", eds.: G.Hunt & H.E.Payne, ASP Conf. Ser., **125**, 46

Verkhodanov O.V., Trushkin S.A., Andernach H., Chernenkov V.N., 1997, in: "Astronomical Data Analysis Software and Systems VI", eds.: G.Hunt & H.E.Payne, ASP Conf. Ser., **125**, 322

Verkhodanov O.V., Kononov V.K., 2002, Bull. Spec. Astrophys. Obs., **53**, 119

Verkhodanov O.V., Doroshkevich A.G., Naselsky P.D., Turchaninov V.I., Novikov I.D., Cristensen P.R., 2003, in: Book of Abstracts, Danish Phys. Soc. Ann. Meeting, 2003, Hotel Nyborg Strand, June 12-13, HCOe Tryk, Kobenhavn, AA30P

Verkhodanov O.V., Doroshkevich A.G., Naselsky P.D., Novikov D.I., Turchaninov V.I., Novikov I.D., Christensen P.R., 2004, in: Proc. Sternberg Astron. Instit., V.LXXV, Abstr. of All Russian Astron. Conf. VAC-2004 "Horizons of Universe" (in Russian), MSU, ISSN 0371-6769, p.184