Physics – Uspekhi **56** (8) 801 – 812 (2013)

INSTRUMENTS AND METHODS OF INVESTIGATION PACS numbers: **95.75. – z**, 95.85.Bh, **98.80. – k**, 98.80.Es

Sky pixelization for the analysis of extended emission

O V Verkhodanov, A G Doroshkevich

DOI: 10.3367/UFNe.0183.201308c.0849

Contents

1. Introduction **801**
1.1 Hierarchical triangular grid; 1.2 Sky mapping in the COBE experiment
2. Introduction to sky mapping **803**
3. Pixelization grids **804**
3.1 Icosahedron pixelization; 3.2 Igloo tilings; 3.3 HEALPix hierarchical pixelization
4. Sky pixelization using the Gauss–Legendre method. The GLESP scheme **806**
4.1 Pixelization scheme; 4.2 Repixelization problem
5. Conclusion **810**
6. Appendices **810**
A. Normalized joint Legendre polynomials; B. Description of the GLESP code
References **811**

Abstract. Spherical pixelization schemes are reviewed that allow analyzing extended emission and, in particular, the cosmic microwave background. Problems with implementing different schemes are considered. The nonhierarchical Gauss–Legendre sky pixelization (GLESP) approach is discussed in detail.

1. Introduction

In the last decade of cosmological research, both ground-based and ambitious space observations of the cosmic microwave background (CMB) have been carried out. Among these experiments, WMAP (Wilkinson Microwave Anisotropy Probe)[1] of NASA (National Aeronautic and Space Administration) [1, 2], which was completed in 2010, and Planck[2] of the ESA (European Space Agency) [3], which was completed in 2013, stand out. The meaning of these experiments is determined by the fact that they provide important insight into our understanding of the model of the Universe and accurately measure the main parameters of the Universe. The achieved progress is, first of all, due to new technologies and instruments in centimeter, millimeter, and submillimeter astronomy. Modern CMB data processing allows comparing these observations with theoretical descriptions of the physical processes that were underway in the early Universe.

The pipeline of CMB data processing and measurement of cosmological parameters includes several steps:

(1) receiving time-ordered data (TOD);

(2) CMP data pixelization and preparation of multi-frequency sky maps;

(3) component separation and cosmological signal rectification using observational data at different wavelengths (see, e.g., [4–9]);

(4) statistical analysis of CMB maps (see, e.g., [10–21]);

(5) harmonic transformations 'map–spherical harmonics';

(6) calculation and analysis of the angular power spectra $C(\ell)$ (see, e.g., [22, 23]);

(7) estimation of cosmological parameters (see, e.g., [24–26]).

The problem of representing and analyzing a sky map appears not only in microwave sky surveys but also in all other sky surveys from the gamma-ray to the low-frequency radio band (for example, BATSE (Burst And Transient Source Experiment) [27] in the gamma-ray band, ROSAT (ROentgenSATellite) [28] in the X-ray band, SDSS (Sloan Digital Sky Survey) [29] in the optical band, IRAS (InfraRed Astronomical Satellite) [30] and 2MASS (Two Micron All Sky Survey) [31] in the infrared band, NVSS (National Radio Astronomy Observatory (NRAO) Very Large Array Sky Survey) [32] in the radio (1.4 GHz) band, radio sky survey at 45 MHz [33]), as well as in neutrino astronomy (IceCube [34, 35]) and cosmic-ray experiments. In addition, sky pixelization systems turn out to be important in constructing databases requiring fast searches of sky objects (see Section 1.1).

In this paper, we deal with one point of the technological pipeline related to the sky pixelization procedure. Sky pixelization is the representation of a sphere as a collection

[1] http://lambda.gsfc.nasa.gov.
[2] http://www.rssd.esa.int/Planck/.

O V Verkhodanov Special Astrophysical Observatory, Russian Academy of Sciences, 369167 Nizhnii Arkhyz, Karachaevo-Cherkesia, Russian Federation
E-mail: vo@sao.ru
A G Doroshkevich Astro Space Center, Lebedev Physical Institute, Russian Academy of Sciences, Profsoyuznaya ul. 84/32, 117997 Moscow, Russian Federation
E-mail: dorr@asc.rssi.ru

Received 14 June 2012, revised 18 July 2012
Uspekhi Fizicheskikh Nauk **183** (8) 849 – 862 (2013)
DOI: 10.3367/UFNr.0183.201308c.0849
Translated by K A Postnov; edited by A M Semikhatov

of small areas (pixels[3]) with a given angular resolution. This problem has largely been solved.

The choice of a particular pixelization scheme depends on the features of the problem to be solved. Ordinary pixelization, which is used in CCD (charge-coupled device) matrices with equal-area square pixels, is now a sufficiently simple task, and in the limit case (only for visualization), it can be applied for partitioning a sphere as well. In the analysis of pixels in a rectangular (cylindrical) coordinate system on the 'unfolded' celestial sphere, it is necessary to take projective distortions and changes in the area of a pixel into account as a function of its distance from the equator.

Before describing pixelization schemes of extended signal maps, we also note one of the methods of partition of a sphere, which is used to search for objects in astronomical databases and allows the indexation of individual objects in the sky.

1.1 Hierarchical triangular grid

Fast access to a chosen area of the sky is the key problem in giant (terabyte) astronomical archives. A clear example is the public database of the Sloan Digital Sky Survey (SDSS),[4] containing information on about 200 mln objects (stars, galaxies, quasars, etc.) in five optical bands, which can be used both by professional astronomers and for educational purposes. To provide Internet access to the archive, the dedicated *Sky Server* [37] was designed, which uses a special pixelization scheme, the Hierarchical Triangular Mesh (HTM)[5] [38]. HTM pixelization is based on splitting the sky into elements of a spherical octahedron (Fig. 1), and each spherical triangle on its surface is recursively divided into smaller triangles (Fig. 2).

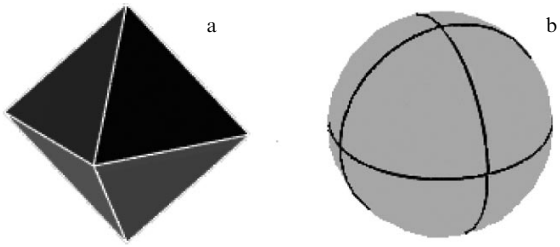

Figure 1. Octahedron partition of a sphere: (a) octahedron, (b) spherical triangles with octahedron vertexes.

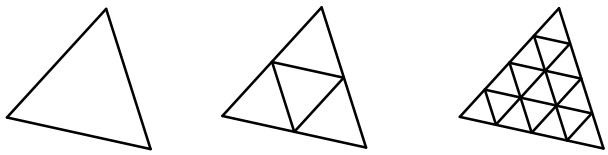

Figure 2. Hierarchical division of a triangle with increasing resolution.

[3] The word 'pixel' (from picture element) was introduced by Frederic Billingsley (NASA Jet Propulsion Laboratory) in 1965 to describe minimal elements of the Moon and Mars images taken from spacecraft [36]. The notion of a 'pixel' includes coordinates or the number of an image, as well as some values (measurements) corresponding to a given area: the number of detected photons, temperature, color, etc.

[4] http://www.sdss.org.

[5] http://www.skyserver.org/htm/.

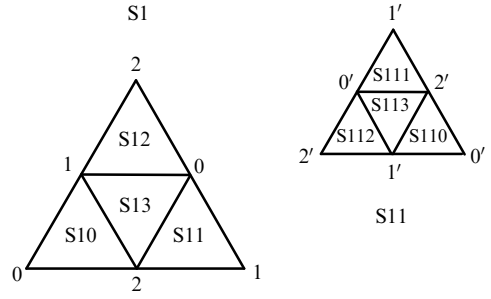

Figure 3. Trixel naming after numeration of angles.

The original partitioning into base-resolution pixels starts from eight equal-size pixels, produced by the intersection of three big circles on a sphere: the equator and two meridians with latitudes 0° and 90°. The hierarchical scheme starts at the next step of increasing the angular resolution: each pixel is divided into four approximately equal-area spherical triangles called trixels. The trixels are produced by medians of the previous-level triangle formed by big circle arcs. At a given hierarchical level, areas of trixels differ within ±70% from the mean area $2\pi/4^{d+1}$, where d is the trixel level depth (the recursive spitting step). For a given depth, the number of trixels is $N_{\rm pix} = 8 \times 4^d$, the minimal side length is $\pi/2^{d+1}$, and the maximal side length is approximately $\pi/2$ as large.

Trixel partitioning is a powerful tool for indexing fields on sky maps and significantly increases the speed of searching for sources in astronomical databases. The indexation system is based on eight base trixels (or zero-level trixels), which are denoted N0 – N3 and S0 – S3 for North (N) and South (S). Each trixel has three vertices numbered 0, 1, and 2. The midpoints of the opposite sides are respectively marked as 0, 1, and 2. At the next levels, the numbers of trixels are composed by adding one of the digits 0, 1, and 2, which indicate the vertex common with the parent trixel (Fig. 3). The central trixel is marked as 3. The resulting length of the trixel name also shows the hierarchy level. In such a decomposition, the number of a trixel consecutively includes the number of the zero level (from 0 to 7), followed by the number (from 0 to 3) of next-level trixels. This numbering allows associating the center of a trixel with a unique 64-bit identifier (called HtmID) representing the corresponding split hierarchy. In this representation, the indexing can be used up to the 31st level. For example, the 25th level yields a resolution of about 60 cm on Earth or 0.02 arc second on the celestial sphere. We note that not every integer number from a given interval corresponds to some trixel. Nor is there any unique trixel corresponding to a given point in the sky, because any point on a sphere can fall within trixels with different hierarchies.

1.2 Sky mapping in the COBE experiment

In problems related to harmonic expansion of signals, special schemes for partitioning a sphere are applied. The relevance of sky pixelization increased significantly after the COBE (COsmic Background Explorer)[6] experiment [39]. The COBE team applied a so-called quadrilaterized (i.e., represented as six squares) QSC projection (Quadrilaterized Sky Cube Projection) [40–42]. For example, Fig. 4 shows a quadrilaterized projection of a map of Earth.

[6] http://lambda.gsfc.nasa.gov/product/cobe.

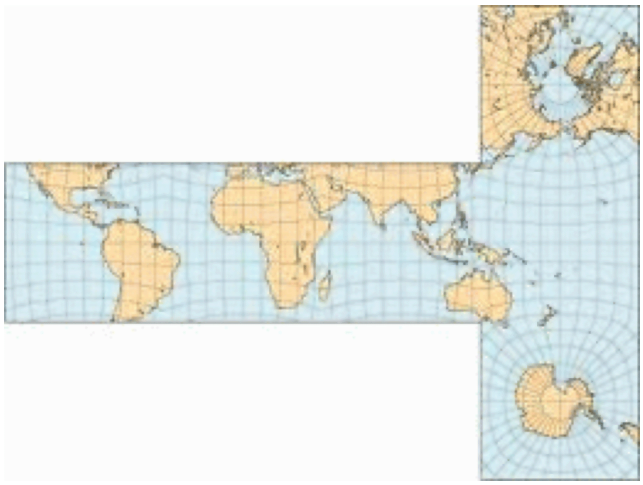

Figure 4. Quadrilateral projection of a map of Earth.

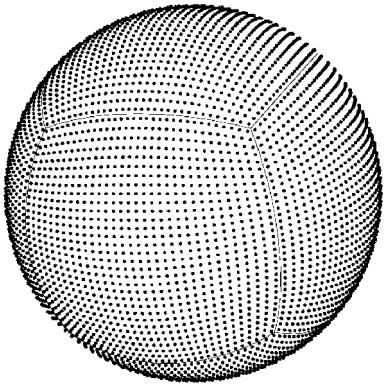

Figure 5. Cubic pixelization. (From [43].)

Based on such a projection, the following pixelization scheme has been proposed (Fig. 5).

(1) A cube is inscribed into a sphere, and the sides of the sphere corresponding to the cube faces are split into pixels using a regular square grid. A pixel includes a set of points that are closer to a given node of the grid than other nodes.

(2) The emission from solid angles subtended by pixels is integrated.

(3) To make pixels equal in area, their centers are somewhat shifted along a circle.

In the QSC grid, a pixel has a nearly quadratic shape with the side $\sqrt{4\pi/N}$, where N is the number of pixels. Points that are farthest from pixel centers lie in the corners of the squares, at the corresponding distance $d_{\rm cube} \approx \sqrt{2\pi/N}$. To minimize the distance d, it is convenient to use a pixelization based on platonic solids, but with a different mesh form. This is discussed in Section 3.1 below.

The application of QSC pixelization to the sky led to the recognition that the pixelization scheme significantly affects the data analysis of extended sky emissions. But it should be noted from the very beginning that the choice of the pixelization scheme, although important, is not the final step in the CMB data analysis. The expansion of measured temperature anisotropy in spherical harmonics and analysis of the angular power spectrum play important roles in this problem. However, some pixelization schemes can describe sky maps very precisely while being poorly suited to harmonic expansion. This shows that the choice of the optimal scheme for sky map pixelization depends on the purposes of a particular study. We return to this question in Section 4.

In this paper, we consider several schemes proposed for sky map pixelization and relate them to the CMB harmonic expansion. In Section 2, we start by describing sky mapping from time series. In Section 3, we briefly describe several different pixelization schemes. In Section 4, we consider the GLESP pixelization scheme in more detail, which offers the most effective harmonic expansion of CMB temperature and polarization anisotropy maps and a description of space–frequency CMB properties. Section 5 contains our conclusions. Some technical details are presented in Appendices A and B.

2. Introduction to sky mapping

Sky pixelization is closely related to map-making, which allows constructing a two-dimensional map of emission distribution on a sphere. For completeness, we outline the map-making algorithm for one-horned experiments [44].

We assume that the time-ordered data (TOD) output $\mathbf{d}$ and a map in pixels $\mathbf{m}$ are linearly dependent:

$$\mathbf{d} = P\mathbf{m} + \mathbf{n}\,, \tag{1}$$

where $\mathbf{n}$ is a vector of random noise and P is some known matrix relating the signals $\mathbf{d}$ and $\mathbf{m}$. The rectangular $N_{\rm d} \times N_{\rm p}$ matrix P is called the pointing matrix; here, $N_{\rm d}$ is the number of sky observations and $N_{\rm p}$ is the number of sky pixels of the size $\sim$ FWHM/3, where FWHM is the full width at half maximum of the antenna beam. That is, applying P to a map 'unfolds' the latter on TOD according to a given scanning strategy. Conversely, applying $P^{\rm T}$ to the TOD 'collects' them into a map. The value of a pixel of this map is the sum of all the observations of that pixel made at different times according to a given scanning strategy.

The structure of the matrix P depends on what we assume for $\mathbf{m}$. If $\mathbf{m}$ contains a pixelized but unsmeared image of the sky, then P must account for beam smearing. That is a very general assumption, which, for example, allows properly treating an asymmetric beam profile, such as quasi-Gaussian beams in the WMAP experiment (Fig. 6) or possible effects of horn degradation during the experiment [45]. In this case, applying the matrix P to the vector $\mathbf{m}$ implies both convolving the sky pattern with the detector beam response and unfolding $\mathbf{m}$ into a 'signal-only' time stream. If, on the other hand, the beam is at least approximately symmetric, then it is possible and certainly more convenient to regard P as a beam-smeared pixelized sky. The structure of P for a one-horned experiment would then be very simple. Only one element per row would differ from zero: the one connecting the observation of the jth pixel to the ith element of the TOD. Many methods have been proposed to estimate $\mathbf{m}$ in Eqn (1) (see, e.g., [46] for a review). Because the problem is linear in $\mathbf{m}$, the generalized least-squares (GLS) method can be used. This involves minimizing the quantity

$$\chi^2 = \mathbf{n}^{\rm T} V \mathbf{n} = (\mathbf{d}^{\rm T} - \mathbf{m}^{\rm T} P^{\rm T})\, V (\mathbf{d} - P\mathbf{m}) \tag{2}$$

for some nonsingular symmetric matrix V. In order to have a 'low noise' algorithm of the χ^2 estimation (a so-called estimator), we have to find a V that minimizes the variance of $\bar{\mathbf{m}}$. This is attained if we take V to be the noise inverse covariance matrix, i.e., $V^{-1} = N \equiv \langle \mathbf{n}\mathbf{n}^{\rm T} \rangle$. We can then

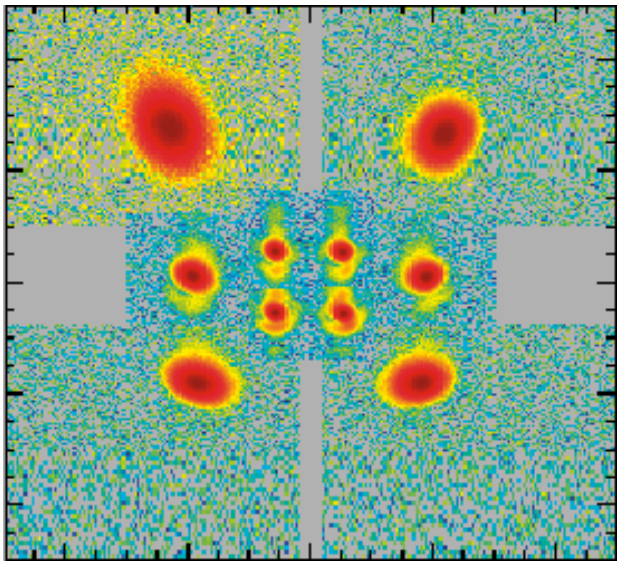

Figure 6. Examples of quasi-Gaussian beams. Shown are beams in the WMAP mission for channels K (0.88°), Ka (0.66°), Q (0.51°), V (0.35°), and W (0.22°). (Figure taken from http://lambda.gsfc.nasa.gov.)

construct the estimator $\tilde{\mathbf{m}}$ for the map:

$$\tilde{\mathbf{m}} = (P^{\mathrm{T}} V P)^{-1} P^{\mathrm{T}} V \mathbf{d}\,. \tag{3}$$

The GLS solution of the map-making problem is then

$$\bar{\mathbf{m}} = \Sigma^{-1} P^{\mathrm{T}} N^{-1} \mathbf{d}\,, \tag{4}$$

where

$$\Sigma = P^{\mathrm{T}} N^{-1} P\,.$$

The map-making uses the chosen sky pixelization, which also provides the integration accuracy when doing harmonic expansion of the signal [see Eqn (7) in Section 4.1]. The problem of integration on a sphere was already discussed in the 1970s (see, e.g., [47–50]). This problem became relevant in astrophysics when all-sky surveys appeared, and it became necessary to expand extended signals in spherical harmonics. Presently, sky map pixelization algorithms, namely, the decomposition of the sky into areas in which observational data are integrated using certain rules, are important ingredients of the CMB data processing [51].

3. Pixelization grids

In this section, we consider several proposed and/or realized schemes of sky pixelization for analyzing extended emission.

3.1 Icosahedron pixelization

When thinking of possible sky pixelization schemes, Max Tegmark [43] was the first to ask the question: which sky map pixelization should be considered *good*? Tegmark focused on two criteria that can be used in choosing the location of N points (or pixel centers) on a sphere:

(1) it is necessary to minimize the largest distance d to the nearest pixel (for example, from corner points to the center of a pixel in a square grid);

(2) the grid must provide precise integration at the nodes.

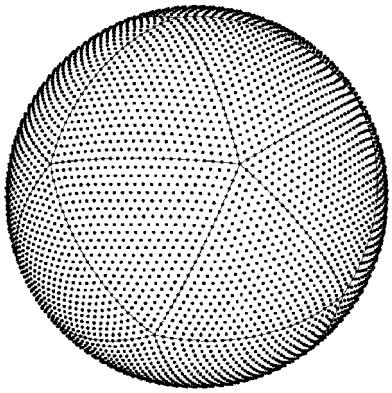

Figure 7. Icosahedron pixelization scheme. (From [43].)

The minimum-d criterion leads to the choice of a pixel shape that is preferable to a square. For example, for the meshed grid in which a pixel is hexagonal, we can easily calculate the value $d_{\mathrm{icosa}} \approx [8\pi/(3\sqrt{3}N)]^{1/2}$, which turns out to be about 12% smaller than in the square-grid case. To use this advantage, it is necessary to change a cube on a sphere by a platonic solid with triangular faces, for example, by a tetrahedron, an octahedron, or an icosahedron. Thus, the choice of d determines not only the grid resolution and pixel locations but also their shape. We note that obtaining equal-area pixels is one of the main problems in constructing the grid. This condition becomes important for pixel convolution and integration because equal weights allow using the same differentials $\mathrm{d}\Omega$ in the integrands.

For a quasilaterized projection, as in the COBE data analysis, when pixels are located on a rectangular grid on faces of a cube (in the tangent plane), the number of 'stretches' increases in going to the face edges. Both the tangential and radial projections somewhat deform the pixel form, such that the further away from the center of an initial face, the more pixels on the sphere differ from those for a regular grid. In this case, it is highly desirable to make faces as small (and hence flat) as possible. A platonic solid with the smallest faces and hence the maximum number of such faces is an icosahedron, consisting of 20 triangles (Fig. 7). Such a partition is advantageous in having the number of faces three times as large as in the cube case, which in turn increases the calculation accuracy due to almost the same pixel areas. In addition, the triangles cover the sky in a more regular way (with less area distortion) (Fig. 8) [43].

The icosahedron pixelization scheme is similar to that used in the COBE experiment:

(1) an icosahedron is inscribed into a sphere, and its faces are divided into a regular triangular grid;

(2) the points (centers of pixels) are radially projected onto the sphere;

(3) the points are somewhat shifted to make all pixels approximately equal in area.

The icosahedron pixelization is advantageous compared with the quadrilaterized scheme in having a larger number of rotational symmetries. The corresponding rotational matrices can be included in the software in advance and can later be called on whenever necessary in the data analysis.

3.2 Igloo tilings

Crittenden and Turok [52] suggested another pixelization scheme, called igloo tilings (because of the similarity with

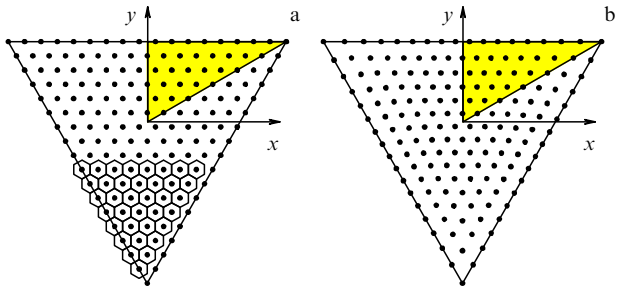

Figure 8. (a) Regular triangular grid is modified to obtain equal-area pixels. The pixels have a hexagonal form. (b) The face of a triangular icosahedron can be symmetrically divided into six equal regular triangles (one of them is marked with a dark color), and the area is equalized for symmetry. A close inspection of the figure reveals how the centers of pixels are displaced [shifted toward the center of the big triangle in Fig. b with respect to that in Fig. a (cf. also the location of pixels along the x axis)] in order to obtain equal-area pixels in the radial projection onto a sphere. (Figure reproduced from [43].)

the Eskimo house). In this scheme, a sphere is divided into rows with constant-latitude edges and each row is divided into identical pixels by lines of constant longitude (Fig. 9). The pixels are roughly trapezoidal, becoming nearly rectangular away from the poles. The northern and southern hemispheres are tiled identically. Igloo tilings have many advantages: (1) they are quite simple; (2) they are naturally azimuthal and can be easily made of equal areas, with most pixels nearly square; (3) most of all, the pixel edges are drawn along constant lines of the spherical polar coordinates θ and ϕ, allowing fast and exact spherical harmonic transformations using pixel values. This is essential in constructing exact simulated skies and in optimally recovering the angular power spectrum from real data. To evaluate the igloo scheme, the authors of [52] consider the following three criteria:

I. *Lack of pixel distortion*. Pixelization of the data suppresses modes with wavelengths shorter than the pixel size in any given direction. To minimize this effect, the largest pixel diameter is to be taken as small as possible. The ideal (unachievable) limit would be to have circular pixels with the largest diameter $D = 4/\sqrt{N_{\rm tot}}$, where $N_{\rm tot}$ is the total number of pixels on the sphere. For square pixels, the best achievable value is $D = \sqrt{8\pi/N_{\rm tot}}$, about 25% larger than for circular pixels. The authors of [52] introduce the concept of the amount of pixel distortion as the maximum diameter normalized by the ideal (circular) case.

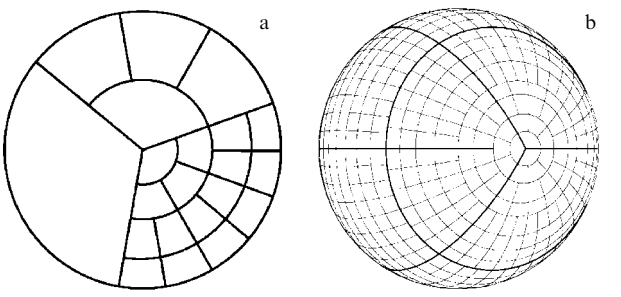

Figure 9. Igloo pixelization. (a) Picture of the polar cap region in the igloo schemes. This hierarchical prescription is designed in order to minimize the distortion in the highest-resolution pixels. (b) Picture of a 3:6:3 equal-area pixelization, which divides the sky into twelve base patches. Here, each base pixel is broken up into 64 small pixels equal in area. (Figure taken from [52].)

II. *Equal-area pixels*. Desirable in order to obtain the best resolution for a fixed number of pixels, in the first approximation, having equal statistic weights.

III. *Built-in hierarchy*. Allows coarse graining of the data at different resolutions, which is sometimes required in data analysis. Igloo pixelization requires that the hierarchy be nested, i.e., that each higher-resolution pixel fit perfectly into a single lower-resolution pixel. Ideally, a minimal set of pixels can be fixed that have minimal distortions and approximately the same area at each level.

The authors of [52] consider four possible igloo pixelizations:

- the simplest example is the ECP (Equidistant Cylindrical Protection) pixelization: the pixel edges are defined along lines of the spherical polar coordinates θ and ϕ. In the ECP scheme, pixels become quite narrow near the poles.
- a twelve-pixel scheme (3:6:3) involving division into equal-area pixels. The divisions between the layers lie at $\theta = \pm 30^\circ$;
- a twelve-pixel scheme (3:6:3) involving division into pixels with equal-latitude spacing;
- an equal-area scheme with 12,116 base equal-area pixels.

When using the scheme with pixels that are of nearly (or exactly) equal area, the number of pixels in each row must decrease in approaching the poles (see Fig. 9). Igloo pixelizations with either rows equally spaced in latitude, like the ECP, or pixels of uniform area can be constructed. The advantage of equal-latitude spacing is that the pixelization can be created by a simple rebinning of an ECP pixelization, if the latter is chosen to have an appropriate number of pixels. In addition, by letting the pixel areas vary, we can make them less distorted. An equal-area pixelization is not exactly equally spaced in latitude, but has the advantage that all pixels have the same statistical weight.

Igloo pixelization can also be made hierarchical. For this, we first divide the sphere into a base pixelization with relatively few pixels, optimized to minimize pixel distortion. Each of these coarse pixels is then divided into four by bisecting it in longitude and latitude. This division is chosen either to keep the pixels of the same area or to maintain a constant-latitude spacing of the rows. This results in a finer-grained pixelization with four times the number of pixels. This procedure can be repeated until we reach the required resolution. Reducing the number of base pixels tends to increase the level of pixel distortions, and therefore some compromise must be found. While there are clear advantages to have fewer base pixels, computing their weights is a complicated task compared with the case of uniform pixels.

The hierarchical partition causes some subpixels to become more distorted than the coarser pixels, especially near the poles. Away from the poles, there is a limit of how distorted the pixels become. However, if the polar cap were simply bisected in θ and ϕ, the pixels would become more and more distorted, as occurs in the ECP.

In standard igloo tilings, the authors of [52] have chosen to initially divide each polar cap into three equal wedges. Higher-resolution pixelizations are found by dividing each wedge into four pieces, one central wedge and three pieces surrounding it (see Fig. 9). This process is iterated.

3.3 HEALPix hierarchical pixelization

HEALPix (Hierarchical Equal Area isoLatitude Pixelization)[7] was in fact the first pixelization scheme supplied with a well-documented computer code [53, 54].

The authors of [53] formulated a list of desiderata for the mathematical structure of discrete whole sky maps, arranged into three points.

(1) *Hierarchical structure of the database*. This is recognized as essential for very large databases, and was already postulated in constructing the Quadrilateralized Spherical Cube. A simple argument in favor of this states that the data elements that are neighbors in a multi-dimensional configuration space (here, on the surface of a sphere) are also neighbors in the tree structure of the database. This property facilitates various topological methods of analysis, and allows easy construction of wavelet transforms on triangular and quadrilateral grids with a fast choice of neighboring pixels.

(2) *Equal areas of discrete elements of the partition*. This is advantageous because white noise at the sampling frequency of the instrument is integrated exactly into white noise in the pixel space, and sky signals are sampled without regional dependence, although care must still be taken to choose a pixel size sufficiently small compared to the instrumental resolution to avoid excessive and pixel-shape-dependent signal smoothing.

(3) *Iso-latitude distribution of discrete area elements on the sphere*. This property is essential for computation speed in all operations involving evaluations of spherical harmonics. Because the associated Legendre polynomials are evaluated via slow recursion, any sampling grid deviations from an iso-latitude distribution result in a prohibitive loss of computational performance with the growing number of sampling points.

All requirements formulated above are satisfied by a construction with the HEALPix of the sphere, which is shown in Fig. 10.

The HEALPix base-resolution comprises twelve pixels in three rings around the poles and the equator. The next hierarchical level is formed from the previous one by dividing each pixel of the previous-resolution level into four equal parts. The resolution of the grid is expressed by the parameter $N_{\rm side}$ (see Fig. 10), which defines the number of divisions along the side of a base-resolution pixel that is needed to reach the desired high-resolution partition, for example, due to a horn beam. All iso-latitude rings located between the upper and lower corners of the equatorial base-resolution pixels ($-2/3 < \cos\theta_* < 2/3$) or in the equatorial zone are divided into the same number of pixels: $N_{\rm eq} = 4N_{\rm side}$. The remaining rings are located within the polar cap regions ($|\cos\theta_*| > 2/3$) and contain a varying number of pixels, increasing from ring to ring by one pixel within each quadrant with increasing the distance from the poles. Hence, a HELAPix-pixelized map consists of $N_{\rm pix} = 12N_{\rm side}^2$ pixels with the same area $\Omega_{\rm pix} = \pi/(3N_{\rm side}^2)$.

The authors of [54] use two systems of indexation, which are applied to process and store maps in the form of FITS-files (Flexible Image Transport System): with isolatitude, or ring, index and nested index (Fig. 11). In the first case, we can simply count the pixels moving from the north to the south pole along each isolatitude ring. In the second case, we can replicate the tree structure of pixel numbering by using a certain algorithm [54]. The nested indexation allows applying the HEALPix scheme inside a database, as was described above for the HTM grid. Special procedures of the HEALPix package allow using alternative indexation schemes for the analyzed map.

[7] http://healpix.jpl.nasa.gov.

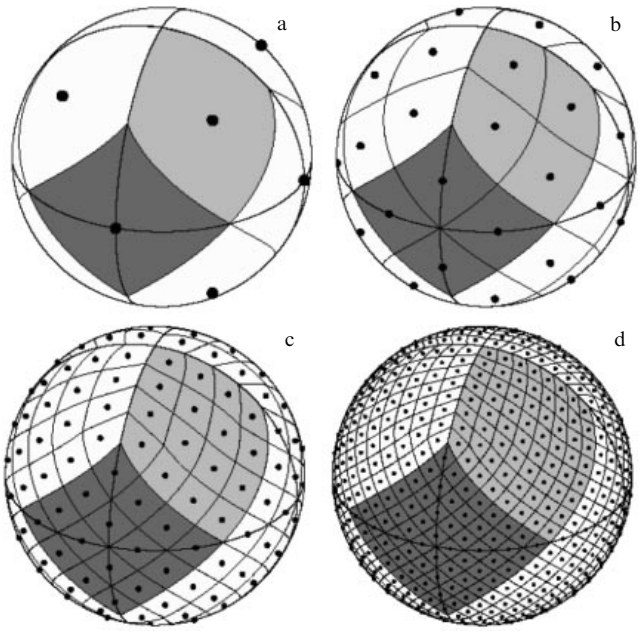

Figure 10. Orthographic view of a HEALPix division of a sphere. Overplot of equator and meridians illustrates the octahedral symmetry of the HEALPix construction. The lowest resolution corresponds to twelve base-resolution pixels. The sphere is hierarchically mosaically partitioned into curvilinear quadrangles. Light-gray shading shows one of eight (four north and four south) identical polar base-resolution pixels. Dark gray shading shows one of four identical equatorial base-resolution pixels. The grid resolution of the mosaic increases upon dividing each pixel into four new ones. The grid resolution increases in three steps from the base level (from Fig. a consecutively to Figs b, c, and d). The grid resolution is characterized by the parameter $N_{\rm side}$ [equal to 1 (a), 2 (b), 4 (c), and 8 (d)], which determines the total number of pixels $N_{\rm pix} = 12N_{\rm side}^2$, i.e., the sphere is partitioned into 12 (a), 48 (b), 192 (c), and 768 (d) pixels. Pixels are equal in area for each grid resolution. All pixel centers are located on rings of constant latitude, which is important for harmonic analysis and calculation of spherical harmonics.

The HEALPix software package, consisting of individual programs and program libraries, includes procedures for harmonic expansion on a sphere of the temperature and polarization anisotropy maps. Among the most frequently used procedures are *synfast*, for the construction (and modeling) of maps; *anafast*, for multipole expansion and calculation of the angular power spectrum (with an option of data masking); and *map2gif*, for imaging maps on a sphere.

4. Sky pixelization using the Gauss–Legendre method. The GLESP scheme

The pixelization procedure discussed above was mainly devoted to the optimal choice of pixel shape or filling a sphere, and the accuracy of calculating harmonic-analysis integrals was addressed after the choice of the pixelization scheme. We recall that the CMB pixelization on the sphere is only one part of the main problem of determining the coefficients $a_{\ell m}$ in the CMB signal expansion in spherical harmonics.

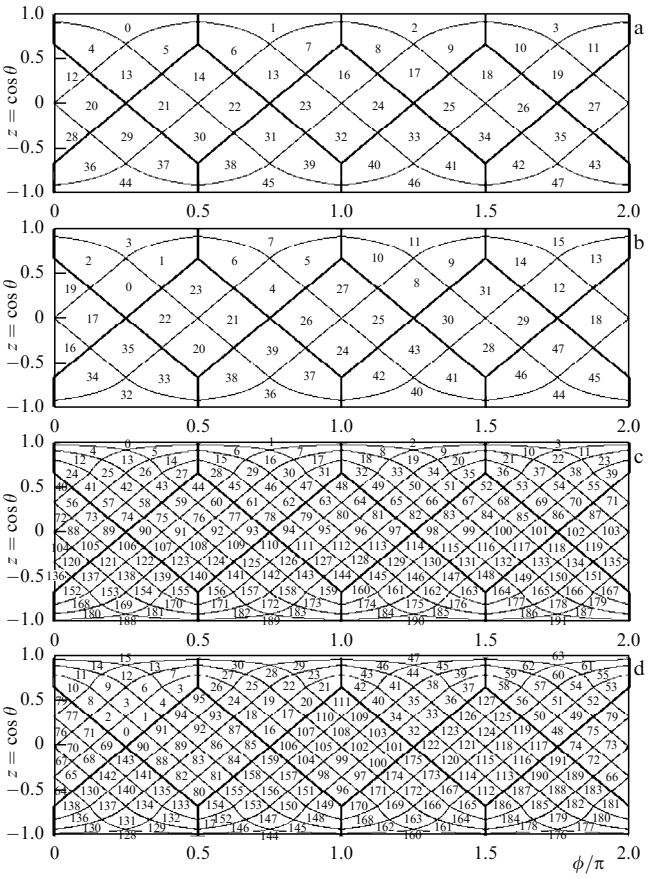

Figure 11. Two systems of pixel indexation on a sphere for HEALPix grids with the grid resolution parameter (a, b) $N_{\rm side} = 2$ and (c, d) $N_{\rm side} = 4$. Shown is numbering along isolatitude rings (a, c) and in the nested tree case (b, d).

4.1 Pixelization scheme

The coefficients $a_{\ell m}$ are calculated using the standard expansion of the measured temperature variations in the sky, $\Delta T(\theta, \phi)$, in spherical harmonics:

$$\Delta T(\theta,\phi) = \sum_{\ell=2}^{\infty} \sum_{m=-\ell}^{m=\ell} a_{\ell m} Y_{\ell m}(\theta,\phi)\,, \tag{5}$$

$$Y_{\ell m}(\theta,\phi) = \sqrt{\frac{2\ell+1}{4\pi}\,\frac{(\ell-m)!}{(\ell+m)!}}\,P_\ell^m(x)\exp(\mathrm{i}m\phi)\,, \quad x=\cos\theta\,, \tag{6}$$

where $P_\ell^m(x)$ are the associated Legendre polynomials (see, e.g., [55] and Appendix A), ℓ is the number of a spherical harmonic (multipole), and m is the number of the multipole mode. For a continuous function $\Delta T(x, \phi)$, the expansion coefficients $a_{\ell m}$ are calculated as

$$a_{\ell m} = \int_{-1}^{1} \mathrm{d}x \int_0^{2\pi} \mathrm{d}\phi\, \Delta T(x,\phi)\; Y^*_{\ell m}(x,\phi)\,, \tag{7}$$

where $Y^*_{\ell m}$ denotes the complex conjugation of $Y_{\ell m}$. A somewhat more complicated scheme is used for the analysis of CMB polarization.

In 2003, the GLESP (Gauss–Legendre Sky Pixelization) scheme, primarily designed for harmonic analysis, was worked out [56–59]. In such an analysis, we need to evaluate integral (7) over the coordinate θ with very high accuracy. Bearing this in mind when constructing the grid allows changing the approach to data processing on a sphere, which, in particular, determines the pixelization scheme. For precise numerical evaluation of integral (7), we use Gaussian quadratures, a method proposed by Gauss already in 1814 and developed later by Christoffel in 1877. The key point of the method is that the integral over x in Eqn (7) is an integral of a polynomial, and we can therefore use weight functions $w(x)$ [60] to achieve the exact equality:

$$\int_{-1}^{1} \mathrm{d}x\, \Delta T(x,\phi)\; Y^*_{\ell,m}(x,\phi) = \sum_{j=1}^{N} w_j\, \Delta T(x_j,\phi)\; Y^*_{\ell,m}(x_j,\phi)\,, \tag{8}$$

where N is the maximal rank of the Legendre polynomial. Here, both the weight function $w_j = w(x_j)$ and $\Delta T(x_j,\phi)\, Y^*_{\ell,m}(x_j,\phi)$ are taken at points that are the net roots,

$$P_N(x_j) = 0\,. \tag{9}$$

It is well known that the equation $P_N(x_j) = 0$ has exactly N zeros in the interval $-1 \leqslant x \leqslant 1$. For the Gauss–Legendre method [Eqn (8)], the weight coefficients are [60]

$$w_j = \frac{2}{1-x_j^2}\left[P'_N(x_j)\right]^{-2}, \tag{10}$$

and they can be calculated together with the set of x_j with the 'gauleg' code [60]. The use of such a method for calculating integral (7), based on Gauss quadrature (7), allows calculations to within the machine epsilon of floating point arithmetic. This harmonic expansion of temperature and polarization anisotropy maps is most effective from the standpoint of computational error minimization in comparison with other methods.

We refer to the pixelization grid in which pixel centers coincide with nodes in the Gauss–Legendre quadrature and pixels are quasi-equal in area as GLESP (Gauss–Legendre Sky Pixelization) [56].[8] To compare the HEALPIx and GLESP grids, Fig. 12 shows the location of centers of rings

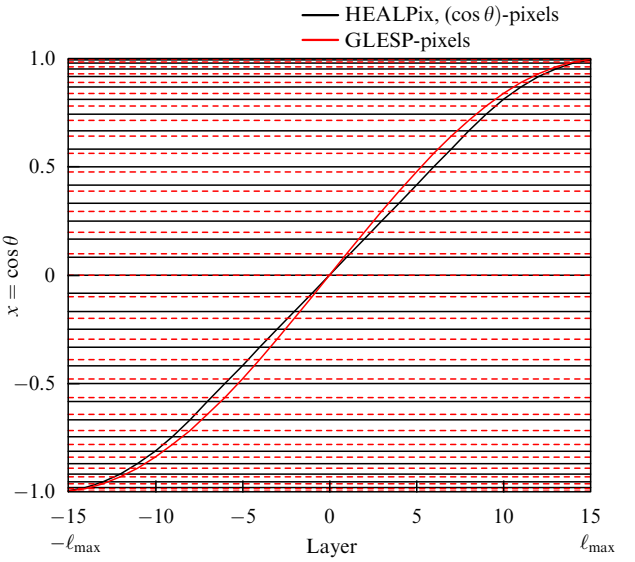

Figure 12. Two schemes of sky pixelization: nodes along the $\cos\theta$ axis HEALPix (solid line) and GLESP (dashed line).

[8] http://www.glesp.nbi.dk.

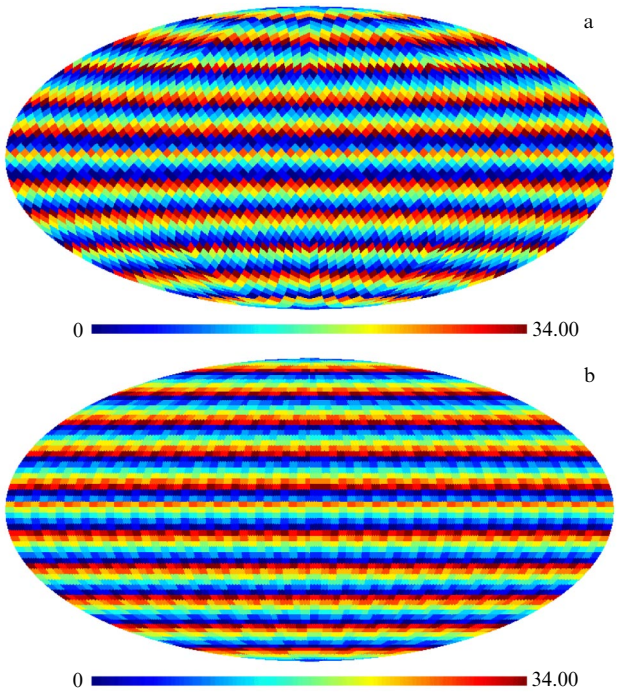

Figure 13. Two types of pixelization of a sphere: pixels (a) in the Mollweide projection (HEALPix) and (b) in the GLESP scheme.

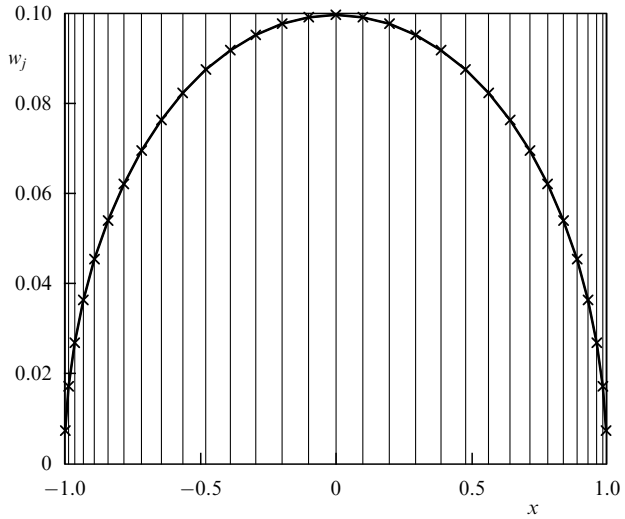

Figure 14. Gauss–Legendre weight coefficients w_j as functions of the zeros of Legendre polynomials ($x_j = \cos\theta_j$) at which rings are centered in the GLESP scheme (for $N = 31$). Vertical lines show the location of the zeros.

$x = \cos\theta$ on the sphere, which are used in both pixelization schemes. We note that the calculation of integral (7) can be done by different means, but the pixelization scheme is constructed using fixed nodes in which time series averaging is performed.

In fact, all CMB data have some resolution limit and hence the upper limit in the summation over ℓ in Eqn (8) $\ell \leqslant \ell_{\max}$, where $\ell_{\max}$ is the highest multipole adopted for data processing on a sphere. This means that when constructing the grid of zeros at points x_j and the weights w_j ($j = 1, 2, \ldots, N$), we can set $N = N_{\max}$, where $N_{\max}$ is connected with $\ell_{\max}$.

In the GLESP approach, trapezoidal pixels are 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 $-1 \leqslant x \leqslant 1$ is covered by N rings of pixels. The angular resolution achieved in the measurement of the CMB data determines the upper summation limit in Eqn (5), $\ell \leqslant \ell_{\max}$. To avoid the Nyquist restrictions, we use the number of pixel rings $N \geqslant 2\ell_{\max}$. In order to make the pixels in the equatorial ring (along the ϕ coordinate) nearly square, 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 to the equatorial ring of pixels.

Figure 12 compares some features of the pixelization schemes used in HEALPix and GLESP. Figure 13 compares pixel distribution and shapes on a sphere in the mollview projections of HEALPix and in GLESP. Figure 14 shows the weight coefficients w_j and the position of pixel centers in the case $N = 31$.

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

• in the polar direction $x = \cos\theta$, we define x_j, $j = 1, 2, \ldots, N$ as the grid of roots of Eqn (9);

• each root x_j determines the position of a ring with N_ϕ^j pixel centers with ϕ-coordinates ϕ_i;

• all pixels have nearly equal area;

• each pixel has the weight w_j [see (10)].

In our numerical code implementing the GLESP pixelization scheme, we use the following conditions.

— All pixels are bordered by θ and ϕ coordinate lines. Hence, all pixels are nearly trapezoidal.

— The number of pixels along the azimuthal direction ϕ depends on the ring number. Their number in the ring can be chosen randomly. The number of pixels depends on $\ell_{\max}$ adopted for data processing of an extended source.

— To satisfy the Nyquist theorem requirement, the number of rings must be $N \geqslant 2\ell_{\max} + 1$ along the axis $x = \cos\theta$.

— In order to make equatorial pixels nearly square, the number of pixels along the azimuthal direction ϕ is $N_\phi^{\max} = \lfloor (2\pi/\mathrm{d}\theta_k + 0.5) \rfloor$, where $\lfloor \ldots \rfloor$ denotes the integer part of a number and $k = \lfloor (N+1)/2 \rfloor$, $\mathrm{d}\theta_k = 0.5(\theta_{k+1} - \theta_{k-1})$.

— The nominal size of each pixel is determined as $S_{\mathrm{pixel}} = \mathrm{d}\theta_k\, \mathrm{d}\phi$, where $\mathrm{d}\theta_k$ and $\mathrm{d}\phi = 2\pi/N_\phi^{\max}$ are pixel sizes in the equatorial ring.

— The number of pixels N_ϕ^j in the jth ring at $x = x_j$ is calculated as $N_\phi^j = \lfloor (2\pi(1 - x_j^2)^{1/2}/S_{\mathrm{pixel}} + 0.5) \rfloor$.

— Polar pixels are triangular.

— Because the number N_ϕ^j differs from 2^k, where k is an integer number, to make a fast Fourier transform along the azimuthal direction, we use the FFTW (Fast Fourier Transform in the West) code [61]. This code allows using not only 2^n points but also arbitrary base powers. This is the fastest code among the currently available ones.

In the proposed scheme, all pixels are the same inside each ring, and the maximal difference between pixels from different rings is $\approx 1.5\%$ near the poles (Fig. 15). Away from the poles, most pixels have an approximately equal polar-angle size (Fig. 16). Increasing the resolution decreases the absolute error in the area difference, because nonequivalence of polar and equatorial pixels is proportional to N^{-2}.

GLESP is not a hierarchical structure, but problems with the choice of pixels nearest to a given point can be simply

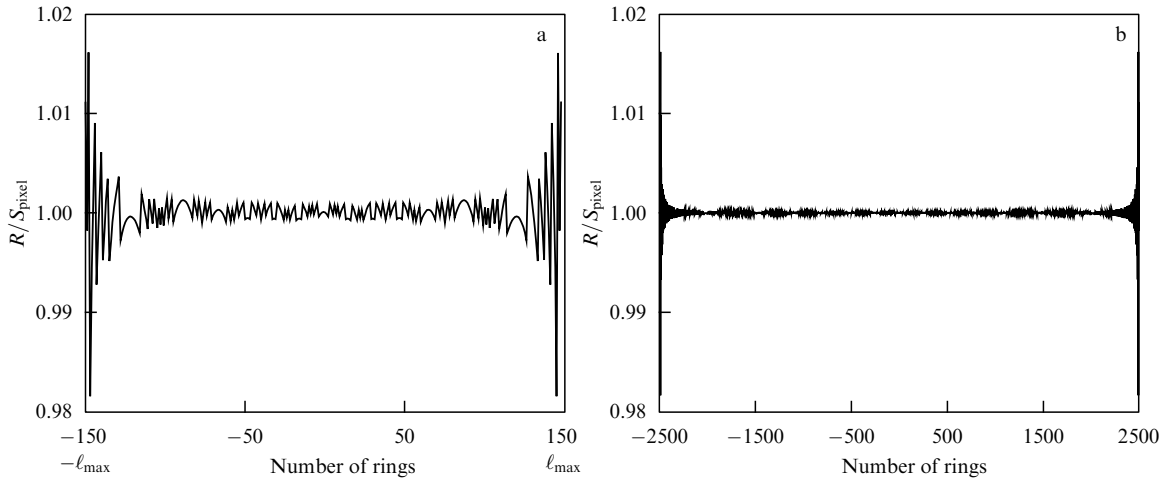

Figure 15. The ratio R of the pixel size to its square S_{pixel} on the equator as a function of the ring number in the GLESP scheme for the total number of rings (a) $N = 300$ and (b) $N = 5000$.

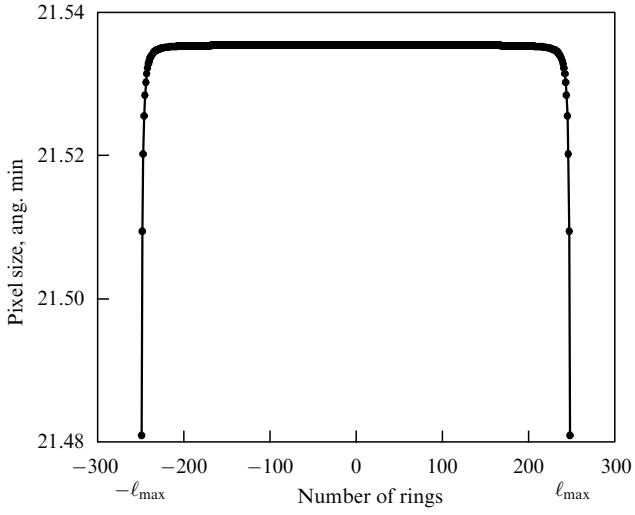

Figure 16. Pixel size along the polar angle ($\ell_{\max} = 250$).

solved at the software level. In addition, using GLESP also allows performing a hierarchical partition similar to that used in the igloo pixelization [52] by dividing each pixel into four parts using meridians and parallels. Although such a procedure distorts the location of polar angles of nodes of the GLESP grid at the next hierarchical level, it offers the possibility of storing and searching for objects in nested structures, similarly to the case of HTM. If a harmonic analysis must be performed at the upper level, where the GLESP grid works very well, the nested pixels can be integrated, but with some loss in resolution. Generally, if some standard set of given map resolutions is fixed, then it is possible, as in some other pixelization schemes, to index the pixels and to use GLESP for searching for sources in the sky and in databases. In GLESP, the choice of the angular resolution for the source list allows using methods of harmonic analysis to search for structures and their correlations on the sphere [62–65].

We note that in spite of the similarity of the GLESP and igloo schemes in the azimuthal division, there is a significant difference in pixel spacing along the polar angle θ. Therefore, we cannot unite these two pixelization schemes. The igloo scheme, when applied to discrete GLESP latitudes, gives pixel sizes that are too different. The pixels have neither equal hight nor equal areas, as the igloo tiling requires.

4.2 Repixelization problem

A nonhierarchical pixelization structure was realized in the GLESP package [58] (see also Appendix B), which is mainly aimed at precisely calculating the harmonic expansion coefficients. The accuracy of calculations in the package procedures is limited only by the machine epsilon of data representation. Nevertheless, there are, of course, natural accuracy limits that restrict the quality of generated maps:

- the noise level of the system;
- the level of knowledge about local, galactic, and extragalactic components from which the sought background signal should be cleaned (see, for example, [9]);
- the instability of methods of separation of the measured signal components and of the CMB signal extraction;
- introduced systematic errors.

A certain type of introduced error is related to the repixelization problem. To transform a map with an extended signal with one pixel grid into another map with another pixel grid (for example, when changing from HEALPix to GLESP or turning the sphere to study a specific projection) without the map-making procedure (see Section 2), we should use one of two methods:

(1) calculate the corresponding coefficients $a_{\ell m}$ and after that reconstruct a new map;

(2) use the repixelization procedure to measure the brightness distribution.

Any of the repixelization procedures leads to a loss of information and introduces new errors and residuals. Nevertheless, the use of this procedure is justified in some cases. The loss of accuracy due to this operation can be demonstrated by reproducing the angular power spectrum $C(\ell)$:

$$C(\ell) = \frac{1}{2\ell + 1}\left(|a_{\ell 0}|^2 + 2\sum_{m=1}^{\ell}|a_{\ell m}|^2\right). \qquad (11)$$

The result of spectral reconstruction by map repixelization is shown in Fig. 17. The spectra are very similar only for ℓ smaller than $\ell_{\max}/4$.

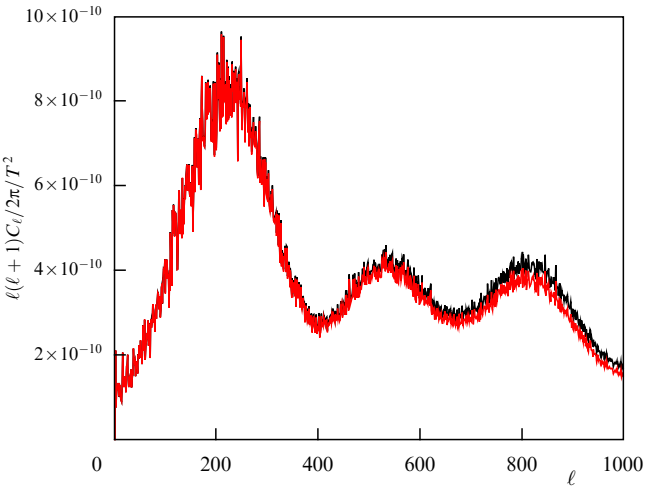

Figure 17. Angular power spectrum calculated for the original HEALPix map (black curve) with $\ell_{\max} = 1000$, $N_{\rm side} = 1024$, pixel size $= 11.8026'$ and the total number of pixels $N_{\rm tot} = 12{,}582{,}912$, and the resulting repixelized GLESP map (gray curve) with the nearest possible pixel size $= 11.8038'$, and $N_{\rm tot} = 12{,}581{,}579$. The angular power spectrum deviations at large ℓ show the ratio of window functions in HEALPix and GLESP.

The GLESP package offers two methods of CMB map repixelization in the pixel region $\Delta T(\theta, \phi)$: first, by averaging input values in a given pixel and second, by spline interpolating over the pixel grid. In the first method, we consider the input pixels that fall within the new pixel with values $\Delta T(\theta_i, \phi_i)$ and then average them using a weight function. The weight function implemented in the code performs simple averaging with equal weight. This method is widespread when assigning given values to a pixel with the corresponding number.

In the second repixelization method, we use a spline interpolation. If there is a map $\Delta T(\theta_i, \phi_i)$ written at nodes different from the Gauss–Legendre ones, we can recalculate it for our new grid $\Delta T(\theta_i', \phi_i')$ by using approximately the same number of pixels and the standard cubic spline interpolation scheme. In this approach, calculations are sufficiently fast because the spline is calculated once for one data vector (for example, in one ring), and obtaining the value of the interpolated function for any input argument requires only one call of an external procedure.

We note that in the GLESP package, it is possible to avoid repixelization problems when transforming maps from HEALPix to GLESP and back, as well as when rotating the maps. In the first case, intermediate calculations of the coefficients $a_{\ell m}$ are used to reconstruct the map, and in the second case, map rotation can be done inside GLESP by the implemented turning procedure and using recursive methods at the level of spherical harmonics [66, 67].

5. Conclusion

We considered several pixelization schemes on a sphere that were proposed to analyze extended sky emissions. Two of them, the hierarchical HEALPix and the nonhierarchical GLESP, are available with the corresponding source codes. Presently, the use of one pixelization scheme or another is a matter of taste and preference in choosing the user software package. The speed of calculations on modern laptops is approximately the same for both packages (for example, the multipole expansion of a sky map or its reconstruction from spherical harmonics take a few seconds for the number of multipoles up to $\ell_{\max} = 2000$). With four calculation iterations, the latest versions of HEALPix (the current version is N2.15) have approximately the same accuracy as GLESP. Both schemes can be used for analyzing CMB polarization. GLESP continues to be developed, and the main advantages of its pixelization grid are exploited:

- high-precision calculations of the coefficients $a_{\ell m}$;
- the possibility of choosing the optimal resolution for a given horn beam, i.e., the optimal size and number of pixels.

The most important features of the GLESP pixelization include its flexibility in filling rings on a sphere and the possibility of projecting the signal on a cylinder for subsequent harmonic analysis. Notably, GLESP includes codes for phase and statistical analysis of multiple events generated in nuclear collisions in the Large Hadron Collider, as well as for identification and studies of phenomena like jets and asymmetric outflows or their combinations [68].

To conclude, we note that if the coefficients $a_{\ell m}$ of the spherical harmonic decomposition are known, they can be used to construct a map with arbitrary pixelization for given pixel centers: GESP, HEALPix, igloo, icosahedrons, etc.

Acknowledgements
The authors thank their colleagues P R Christensen, P D Naselsky, I D Novikov, and V I Turchaninov for the helpful discussions during the development of the GLESP scheme. We thank the referee for valuable notes that improved the text. The HEAPix package [54] was used for analyzing data and some results. The GLESP package [56, 59] was used for analyzing extended emission on a sphere. The work is supported by the RFBR grants 11-02-00244 and 13-02-00027. OVV also acknowledges the Dmitry Zimin Dynasty Foundation for support.

6. Appendices

A. Normalized joint Legendre polynomials

For harmonic transformations in the GLESP code, we use the normalized associated Legendre polynomials

$$f_\ell^m(x) = \sqrt{\frac{2\ell+1}{2}\,\frac{(\ell-m)!}{(\ell+m)!}}\,P_\ell^m(x)\,, \tag{12}$$

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

$$f_\ell^m(x) = x\sqrt{\frac{4\ell^2-1}{\ell^2-m^2}}\,f_{\ell-1}^m - \sqrt{\frac{2\ell+1}{2\ell-3}\,\frac{(\ell-1)^2-m^2}{\ell^2-m^2}}\,f_{\ell-2}^m\,. \tag{13}$$

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 $m \leqslant \ell$:

$$\sqrt{(\ell-m-1)(\ell+m+2)}\; f_\ell^{m+2}(x) + \frac{2x(m+1)}{\sqrt{1-x^2}}\; f_\ell^{m+1}(x) + \sqrt{(\ell-m)(\ell+m+1)}\; f_\ell^m(x) = 0\,. \quad (14)$$

This relation starts with the same $f_\ell^\ell(x)$ and $f_\ell^0(x)$, which must be found with formula (13).

As discussed in [69], the first recurrence relation (13) is formally unstable if the number of iterations goes to infinity. Unfortunately, there are no theoretical recommendations on the maximum iteration number that can be used in the quasi-stability area. But this relation can be used because we are interested in the so-called *dominant* solution [69], which is approximately stable. The second recurrence relation (12) is stable for all ℓ and m.

B. Description of the GLESP code

The GLESP code, based on the GLESP pixelization, is a free Unix-based software compiled with the GNU C compiler. For operation with GLESP, the Fast Fourier Transform FFTW library [61] is required. GLESP has four types of procedures to work with maps on a sphere and the harmonics (the corresponding utilities of GLESP are given in parentheses):

• *Operations related to maps*.

(1) Spherical harmonic decomposition of a map into coefficients $a_{\ell m}$ (*c2lmap*).

(2) Synthesizing CMB temperature maps from a power spectrum $C(\ell)$ (*c2lmap*).

(3) Spherical decomposition of the temperature anisotropy and (Q,U)-polarization maps into $a_{\ell m}$ and $e, b_{\ell m}$ coefficients (*polamp*).

(4) Smoothing a map with a Gaussian beam (*c2lmap*).

(5) Adding/subtracting/averaging maps (*difmap*).

(6) Scalar multiplication/division (*difmap*).

(7) Map rotation (*difmap*).

(8) Repixelization with conversion between Galactic/ecliptic/equatorial coordinates (*difmap*).

(9) Truncating a map to a given temperature range (*mapcut*).

(10) Cutting zones in a map (*mapcut*).

(11) Constructing map cross sections (*mapcut*).

(12) Preparing map patterns using standard samples and analytic formulas (*mappat*).

(13) Converting/creating FITS binary from ASCII (*mappart*).

(14) Setting point-like and elliptical sources on a map (*mappat*).

(15) Converting FITS binary to ASCII (*mapcut*).

(16) Calculating the simplest statistics of a map: mean, variance, extrema (*difmap*).

(17) Displaying the cross-correlation and mosaic coefficients of two maps (*difmap*).

(18) Masking the map and screen pixels on the map (*difmap*, *mapcut*, *f2fig*).

(19) Choosing the required map resolution and giving information on the pixel number and size (*ntot*).

(20) Plotting the map and putting markers on a figure (*f2fig*).

• *Operations with coefficients* $a_{\ell m}$.

(1) Synthesizing CMB temperature maps from coefficients $a_{\ell m}$ (*c2lmap*).

(2) Synthesizing T,Q,U maps from harmonic coefficients $a_{\ell m}$ and $e, b_{\ell m}$ (*polmap*).

(3) Calculating sums and differences of two sets of $a_{\ell m}$ (*difalm*).

(4) Dot product/division of $a_{\ell m}$ (*difalm*).

(5) Cross product/division of $a_{\ell m}$ (*difalm*).

(6) Adding a given phase to all harmonics (*difalm*).

(7) Map rotation in the harmonic representation (*difalm*).

(8) Mode selection from harmonics (*difalm*).

(9) Calculation of the angular power spectrum C_ℓ (*alm2dl*).

(10) Calculation of harmonic phases (*alm2dl*).

(11) Harmonic selection by given phases (*alm2dl*).

(12) Calculation of derivatives of maps from harmonics (*dalm*).

• *Operations with angular power spectrum* C_ℓ.

(1) Calculation of the angular power spectrum C_ℓ (*alm2dl*).

(2) Map modeling from the given angular power spectrum C_ℓ (*cl2map*).

(3) Modeling the coefficients $a_{\ell m}$ from a given angular power spectrum C_ℓ (*createalm*).

(4) Adding/subtracting/averaging the angular power spectrum (*difcl*).

• *Operations with phases* $\varphi_{\ell m}$ *determined from the complex representation of coefficients* $a_{\ell m} = |a_{\ell m}| \exp(\mathrm{i}\varphi_{\ell m})$ *and spherical harmonic amplitudes* $|a_{\ell m}|$.

(1) Calculating phases $\varphi_{\ell m}$ (*alm2dl*).

(2) Calculating amplitudes $|a_{\ell m}|$ (*alm2dl*).

(3) Modeling $a_{\ell m}$ from given phases (*createalm*).

(4) Selecting harmonics with given phases (*alm2dl*).

(5) Adding phase to all harmonics (*difalm*).

(6) Plotting two-dimensional color phase and amplitude diagrams (*f2fig*).

The GLESP map, written as a FITS Binary Table [70], contains a self-documented ASCII header describing data and three binary tables that list Legendre polynomial zeros along the axis $\cos\theta$, the number of pixels in rings on a sphere, and the value of the signal in pixels in consecutive rings from north to south according to nodes of $\cos\theta$.

References

1. Bennett C L et al. *Astrophys. J. Suppl.* **148** 1 (2003); astro-ph/0302207
2. Jarosik N et al. *Astrophys. J. Suppl.* **192** 14 (2011); arXiv:1001.4744
3. Tauber J A et al. *Astron. Astrophys.* **520** A1 (2010)
4. Hinshaw G et al. *Astrophys. J. Suppl.* **170** 288 (2007); astro-ph/0603451
5. Tegmark M, de Oliveira-Costa A, Hamilton A J S *Phys. Rev. D* **68** 123523 (2003)
6. Leach S M et al. *Astron. Astrophys.* **491** 597 (2008); arXiv:0805.0269
7. Stolyarov V et al. *Mon. Not. R. Astron. Soc.* **336** 97 (2002)
8. Kim J, Naselsky P, Christensen P R *Phys. Rev. D* **79** 023003 (2009)
9. Doroshkevich A, Verkhodanov O *Phys. Rev. D* **83** 043002 (2011); arXiv:1008.4094
10. Komatsu E et al. *Astrophys. J. Suppl.* **148** 119 (2003); astro-ph/0302223
11. Chiang L-Y, Naselsky P D, Verkhodanov O V, Way M J *Astrophys. J.* **590** L65 (2003); astro-ph/0303643
12. Naselsky P D, Doroshkevich A G, Verkhodanov O V *Astrophys. J.* **599** L53 (2003); astro-ph/0310542
13. Coles P et al. *Mon. Not. R. Astron. Soc.* **350** 989 (2004)
14. Vielva P et al. *Astrophys. J.* **609** 22 (2004); astro-ph/0310273

15. Park C-G *Mon. Not. R. Astron. Soc.* **349** 313 (2004)
16. Eriksen H K et al. *Astrophys. J.* **612** 64 (2004); astro-ph/0401276
17. Eriksen H K et al. *Astrophys. J.* **605** 14 (2004)
18. Hansen F K et al. *Astrophys. J.* **607** L67 (2004)
19. Larson D L, Wandelt B D *Astrophys. J.* **613** L85 (2004)
20. Schwarz D J et al. *Phys. Rev. Lett.* **93** 221301 (2004)
21. Verkhodanov O V *Phys. Usp.* **55** 1098 (2012) [*Usp. Fiz. Nauk* **182** 1177 (2012)]
22. Larson D et al. *Astrophys. J. Suppl.* **192** 16 (2011); arXiv:1001.4635
23. Kim J, Naselsky P *Astrophys. J.* **714** L265 (2010)
24. Spergel D N et al. *Astrophys. J. Suppl.* **170** 377 (2007); astro-ph/0603449
25. Komatsu E et al. *Astrophys. J. Suppl.* **180** 330 (2009); arXiv:0803.0547
26. Komatsu E et al. *Astrophys. J. Suppl.* **192** 18 (2011); arXiv:1001.4538
27. Paciesas W S et al. *Astrophys. J. Suppl.* **122** 465 (1999); astro-ph/9903205
28. Voges W et al. *Astron. Astrophys.* **349** 389 (1999); astro-ph/9909315
29. Abazajian K et al. (SDSS Collab.) *Astron. J.* **126** 2081 (2003)
30. Beichman C A et al. *Infrared Astronomical Satellite* (*IRAS*) *Catalogs and Atlases* Vol. 1 *Explanatory Supplement* (Washington, DC: Scientific and Technical Information Division, National Aeronautics and Space Administration,1988)
31. Cutri R M et al. "Explanatory supplement to the 2MASS Second Incremental Data Release" (2002), http://www.ipac.caltech.edu/2mass/releases/second/doc/explsup.html
32. Condon J J et al. *Astron. J.* **115** 1693 (1998)
33. Guzmán A E et al. *Astron. Astrophys.* **525** A138 (2011)
34. Abbasi R et al. (IceCube Collab.) *Phys. Rev. D* **82** 072003 (2010)
35. Böser S (for the IceCube Collab.), in *Proc. of the XLVIIth Rencontres de Moriond "Electroweak Interactions and Unified Theories", 4 – 10 March 2012*; arXiv:1205.6405
36. Lyon R F "A brief history of 'pixel'" *Proc SPIE* **6069** 606901 (2006); in *Digital Photography II — Invited Paper. IS&T/SPIE Symp. on Electronic Imaging, 15 – 19 January 2006, San Jose, Calif., USA*; http://www.foveon.com/files/ABriefHistoryofPixel2.pdf
37. Szalay A et al., cs/0111015
38. Kunszt P Z, Szalay A S, Thakar A R, in *Mining the Sky. Proc. of the MPA/ESO/MPE Workshop, Garching, Germany, July 31 – August 4, 2000* (ESO Astrophysics Symp., Eds A J Banday, S Zaroubi, M L Bartelmann) (Heidelberg: Springer, 2001) p. 631
39. Smoot G F et al. *Astrophys. J. Lett.* **396** L1 (1992)
40. Chan F K, O'Neil E M, Computer Sciences Corp. EPRF Technical Report (1976)
41. O'Neill E M, Laubscher R E, Computer Sciences Corp. EPRF Technical Report (1976)
42. Greisen E W, Calabretta M *Bull. Am. Astron. Soc.* **25** 808 (1993); American Astronomical Society, 182nd AAS Meeting (1993) 09.01
43. Tegmark M *Astrophys. J.* **470** L81 (1996)
44. Natoli P et al. *Astron. Astrophys.* **372** 346 (2001); astro-ph/0101252
45. Naselsky P D, Verkhodanov O V, Christensen P R, Chiang L-Y *Astrophys. Bull.* **62** 285 (2007) [*Astrofiz. Byull.* **62** 301 (2007)]; astro-ph/0211093
46. Tegmark M *Astrophys. J.* **480** L87 (1997); astro-ph/9611130
47. Stroud A H *Approximate Calculation of Multiple Integrals* (Englewood Cliffs, N.J.: Prentice-Hall, 1971)
48. Sobolev S L *Cubature Formulas and Modern Analysis: an Introduction* (Philadelphia, Pa.: Gordon and Breach, 1992) [Translated from Russian: *Vvedenie v Teoriyu Kubaturnykh Formul* (Moscow: Nauka, 1974)]
49. Mysovskikh I P, in *Quantitative Approximation. Proc. of a Symp. on Quantitative Approximation, Bonn, West Germany, August 20 – 24, 1979* (Eds R A DeVore, K Scherer) (New York: Academic Press, 1980) p. 217
50. Konyaev S I *Math. Notes Acad. Sci. USSR* **25** 326 (1979) [*Mat. Zametki* **25** 629 (1979)]
51. Verkhodanov O V, Doroshkevich A G, in *Advances in Machine Learning and Data Mining for Astronomy* (Eds M J Way et al.) (Boca Raton: Chapman and Hall/CRC Press, 2012) Ch. 8, p. 133
52. Crittenden R G, Turok N G "Exactly azimuthal pixelizations of the sky", Report DAMTP-1998-78 (1998); astro-ph/9806374
53. Górski K M, Hivon E, Wandelt B D, in *Evolution of Large–Scale Structure: From Recombination to Garching, August 2 – 7 1998. MPA/ESO Cosmology Conf.* (Eds A J Banday, R S Sheth, L Da Costa) (Garshing, 1999) p. 37
54. Górski K M et al. *Astrophys. J.* **622** 759 (2005)
55. Gradshteyn I S, Ryzhik I M *Tables of Integrals, Series, and Products* 6th ed. (San Diego: Academic Press, 2000)
56. Doroshkevich A G et al. *Int. J. Mod. Phys. D* **14** 275 (2005); astro-ph/0305537
57. Doroshkevich A G et al., astro-ph/0501494
58. Verkhodanov O V et al. *Bull. Spec. Astrophys. Obs.* **58** 40 (2005) [*Byull. Spets. Astrofiz. Observ.* **58** 40 (2005)]
59. Doroshkevich A G et al. *Int. J. Mod. Phys. D* **20** 1053 (2011); arXiv:0904.2517
60. Press W H et al. *Numerical Recipes in FORTRAN* 2nd ed. (Cambridge: Cambridge Univ. Press, 1992); http://www.nr.com
61. Frigo M, Johnson S G, Technical Report MIT-LCS-TR-728 (1997)
62. Verkhodanov O V, Khabibullina M L, Majorova E K *Astrophys. Bull.* **64** 263 (2009) [*Astrofiz. Byull.* **64** 272 (2009)]; arXiv:0912.3073
63. Verkhodanov O V et al. *Astrophys. Bull.* **65** 238 (2010) [*Astrofiz. Byull.* **65** 252 (2010)]; arXiv:1009.3720
64. Verkhodanov O V, Khabibullina M L *Astrophys. Bull.* **65** 390 (2010) [*Astrofiz. Byull.* **64** 413 (2010)]; arXiv:1108.4128
65. Verkhodanov O V, Naiden Ya V *Astrophys. Bull.* **67** 1 (2012) *Astrofiz. Byull.* **67** 1 (2012)]
66. Varshalovich D A, Moskalev A N, Khersonskii V K *Quantum Theory of Angular Momentum* (Singapore: World Scientific, 1988)
67. Risbo T *J. Geodesy* **70** 383 (1996)
68. Naselsky P et al. *Phys. Rev. C* **86** 024916 (2012); arXiv:1204.0387
69. Press W H et al. *Numerical Recipes in FORTRAN* 2nd ed. (Cambridge: Cambridge Univ. Press, 1992) Sect. 5.5
70. Hanisch R J et al. *Astron. Astrophys.* **376** 359 (2001)