All about needlets

Joëlle-Marie Bégin

November 24, 2020

1 The needlets

The needlet functions on the sphere are defined

$$\psi_{jk}(\mathbf{r}) = \sqrt{\lambda_j} \sum_{\ell} b\left(\frac{\ell}{B^j}\right) \sum_{m=-\ell}^{\ell} \overline{Y}_{\ell m}(\boldsymbol{r}) Y_{\ell m}(\boldsymbol{\xi}_{jk}) \tag{1}$$

where $\boldsymbol{r}$ and $\boldsymbol{\xi}_{jk}$ are vectors pointing to a position on the sphere. Something that people rarely do in the literature, for some reason, is to simplify this equation using the following identity:

$$\sum_{m=-\ell}^{\ell} Y_{\ell m}(\boldsymbol{x}) \overline{Y}_{\ell m}(\boldsymbol{y}) = \frac{2\ell+1}{4\pi} P_\ell(\boldsymbol{x} \cdot \boldsymbol{y})$$

where P_ℓ are the Legendre polynomials. Then, equation 1 becomes

$$\psi_{jk}(\mathbf{r}) = \sqrt{\lambda_j} \sum_{\ell} \frac{2\ell+1}{4\pi} b\left(\frac{\ell}{B^j}\right) P_\ell(\boldsymbol{r} \cdot \boldsymbol{\xi}_{jk}) \tag{2}$$

which is a *much* nicer expression, in my humble opinion. Why don't people ever write it this way? I don't know, maybe it's supposed to be obvious or something. Anyways, there's a lot to unpack here, so let's take it step by step. λ_j is a normalization factor, which we'll talk about later. $b(\cdot)$ here is a kind of window function, j is a kind of resolution parameter, and $B \in \mathbb{R} > 1$ is another parameter that you choose and fix for the whole transform once you've settled on a good value. Values in literature for B range from 1 to 2, usually. The values of j and k uniquely define a needlet once we've settled on a value for B. As we'll see later, j affects the spatial localization of the needlet, and k in turn defines the position on the sphere where the needlet peaks. Before we can plot some needlets, we have to describe in detail what b, B and j do.

$b(\cdot)$ is the function that makes this whole thing work for many mathy reasons that I don't fully understand. Its most important quality is compact support, though. Generating the function is actually quite complicated (because of all of its other mathy properties), but luckily I found some code that does it! Then, let's take a look at some $b(\cdot)$ so we can start getting an intuition for the j and B parameters.

First, what does B do? The left panel of figure 1 has b plotted for $j = 2$, and different values of B. We can see that the most important thing that B does is change the interval over which b is compactly supported. Basically, since the function b is like a bandpass filter, by changing B we're changing the range of frequencies (or ℓ) we're letting through for a given j.

On the right panel of figure 1, we set $B = 2$ and vary j; this has the effect of sliding the filter along the ℓ axis. For higher values of j, we're letting through higher ℓ.

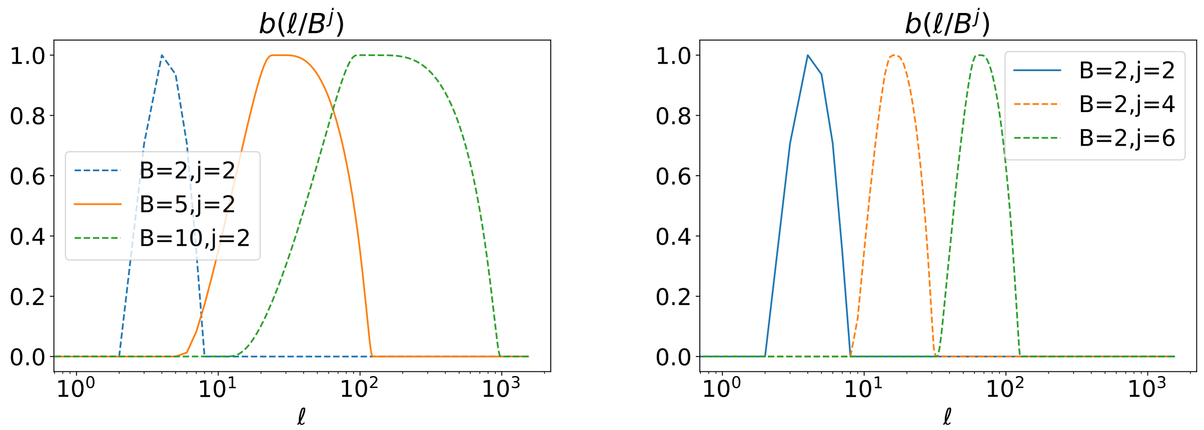

Figure 1: Filter functions plotted for a variety of B and j values.

The maximum value of j that we can use is set by the maximum ℓ that we're considering. $j_{\max}$ is the maximum j that satisfies

$$\lceil B^{j-1} \rceil \leq \ell_{\max} \tag{3}$$

Last thing to explain before we can plot the needlets is $\boldsymbol{\xi}_{jk}$. The set

$$\mathcal{X}_j = \{\boldsymbol{\xi}_{jk}\}_{k=1,2,3,..}$$

contains the cubature points for the resolution j. That is, given the resolution parameter j, this corresponds to a certain discretization of the sphere of some N_{side}^j. For a given resolution j, the corresponding N_{side} is determined by

$$\frac{1}{2}\lfloor B^{j+1} \rfloor \leq N_{\text{side}}^j \tag{4}$$

That is, N_{side}^j is the lowest power of two greater than $\frac{1}{2}\lfloor B^{j+1} \rfloor$. Finally, λ_j is a normalization factor corresponding to the area of each pixel in the discretization[1]. It is given by

$$\lambda_j = \frac{4\pi}{N_{\text{pix}}^j} \tag{5}$$

where N_{pix}^j is the number of cubature points for N_{side}^j. Now we know enough to plot some needlets! First, a pseudo-codey algorithm for computing needlets.

1. Choose a value of j and k. This defines the width and location of your needlet!

2. Your choice of j defines a set of cubature points with N_{side} as given in equation 4. This set limits the possible values of k that you can choose. Your choice of k defines the element of this set where your needlet peaks. So, for example, for $B = 2$ and $j = 2$, we have $N_{\text{side}}^j = 4$. This corresponds to 192 cubature points, so we can have $k \in \{0, 1, .., 191\}$. Then, if we take $k = 2$, this corresponds to the position $(\theta, \phi) = (0.204480, 3.926991)$.

3. Figure out the vector that $\boldsymbol{\xi}_{jk}$ corresponds to, using for example hp.ang2vec of the angle given in the step above. This value of $\boldsymbol{\xi}_{jk}$ is fixed for a given needlet.

4. Take some vector $\boldsymbol{r}$ corresponding to a position on the sphere.

5. Compute the sum
$$\sum_\ell \frac{2\ell+1}{4\pi} b\left(\frac{\ell}{B^j}\right) P_\ell(\boldsymbol{r} \cdot \boldsymbol{\xi}_{jk})$$

6. Normalize by λ_j as given in equation 5, where the N_{pix}^j is given by your choice of j.

7. Repeat Steps 4-6 for a set of positions on the sphere $\boldsymbol{r}$, and you've got your needlet as a function of position on the sphere!

Very important note: the cubature points of your output map, of which $\boldsymbol{r}$ is an element, have *nothing* to do with the cubature points $\mathcal{X}_j$.

In figure 2 we have plotted $\psi_{jk}(\boldsymbol{r})$ for $j = 2, 3$, and for a choice of k that gives $\boldsymbol{\xi}_{jk} = (3\pi/4, \pi/6)$. We can see that the maximum value of the needlets correspond to the angle defined by $\boldsymbol{\xi}_{jk}$; this makes sense when we look at equation 2, because this is the coordinate $\boldsymbol{r}$ where $\boldsymbol{r} \cdot \boldsymbol{\xi}_{jk}$ is maximized. j, as expected, changes the extent of the needlet's localization.

In figure 3, we plot $\psi_{jk}(\boldsymbol{r})$ for $j = 2$, and choices of k that correspond to $(\pi/4, \pi/2)$ and $(\pi/2, 0)$.

[1] All papers in the literature write λ_{jk}, but I'm pretty sure that is just generalizing for discretizations of the sphere that don't have equal pixel area. Since everyone pretty much uses HEALPix, I'm going to drop the k subscript since it's just confusing.

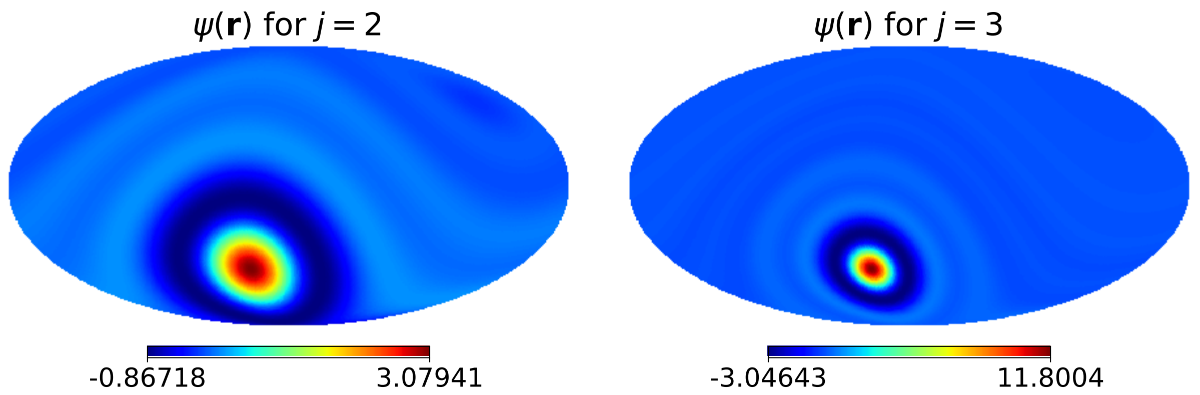

Figure 2: Needlets for $j = 2, 3$, for a choice of k that makes the needlets peak at $(3\pi/4, \pi/6)$.

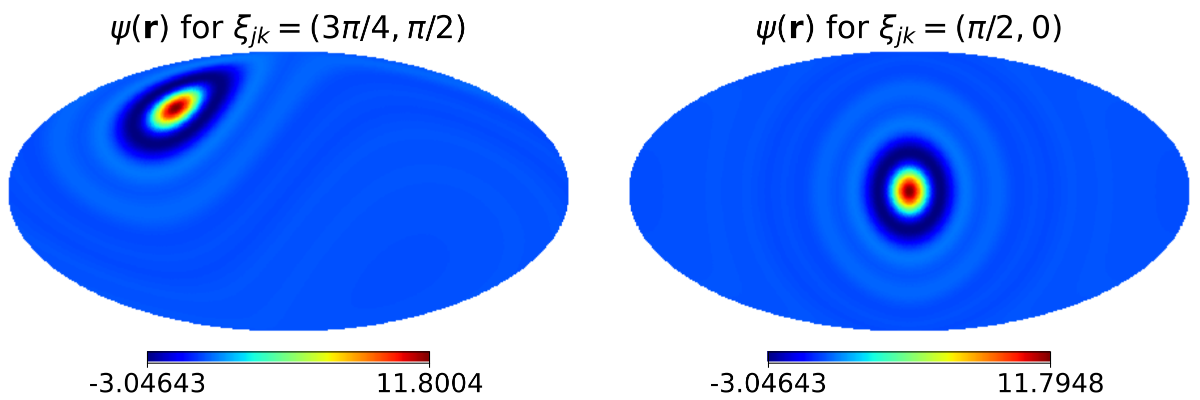

Figure 3: Needlets for $j = 2$, and choices of k that makes the needlets peak at $(\pi/4, \pi/2)$ and $(\pi/2, 0)$.

2 The needlet transform

The needlets form a basis for the sphere, and so any function on the sphere $T(\boldsymbol{r})$ can be written as a linear combination of the needlets:

$$T(\boldsymbol{r}) = \sum_{jk} \beta_{jk} \psi_{jk}(\boldsymbol{r}) \tag{6}$$

2.1 The coefficients

The coefficients of the linear combination are given by

$$\beta_{jk} = \sqrt{\lambda_j} \sum_{\ell} \sum_{m=-\ell}^{\ell} b\left(\frac{\ell}{B^j}\right) a_{\ell m} Y_{\ell m}(\boldsymbol{\xi}_{jk}) \tag{7}$$

This is just an inverse spherical harmonic transform, where we've applied the filter b to the $a_{\ell m}$! So let's take a look at some β_{jk} for a test map.

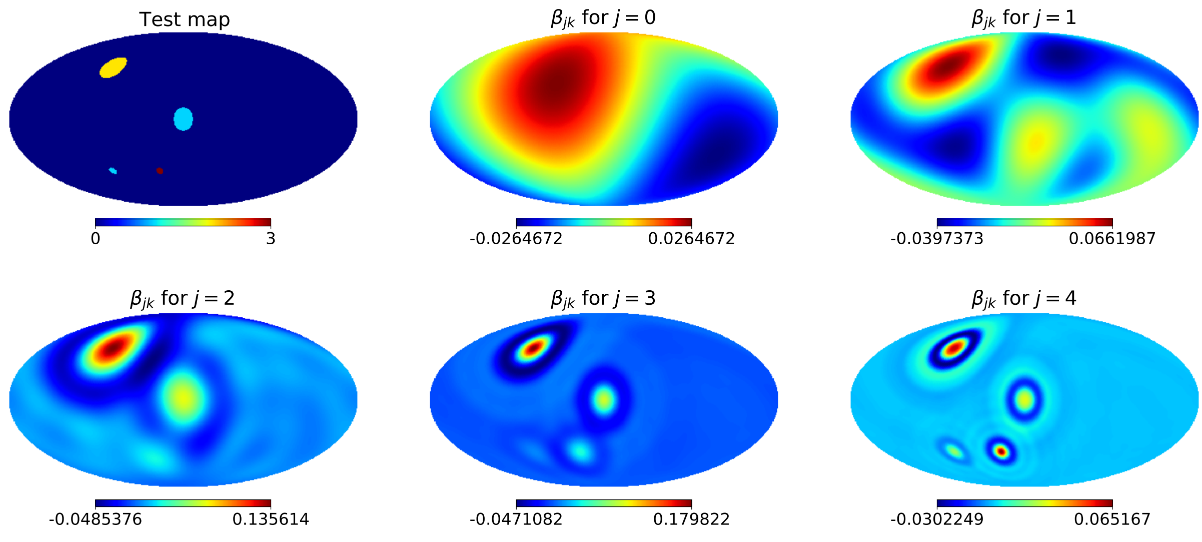

Figure 4: Needlet coefficients for the given test map.

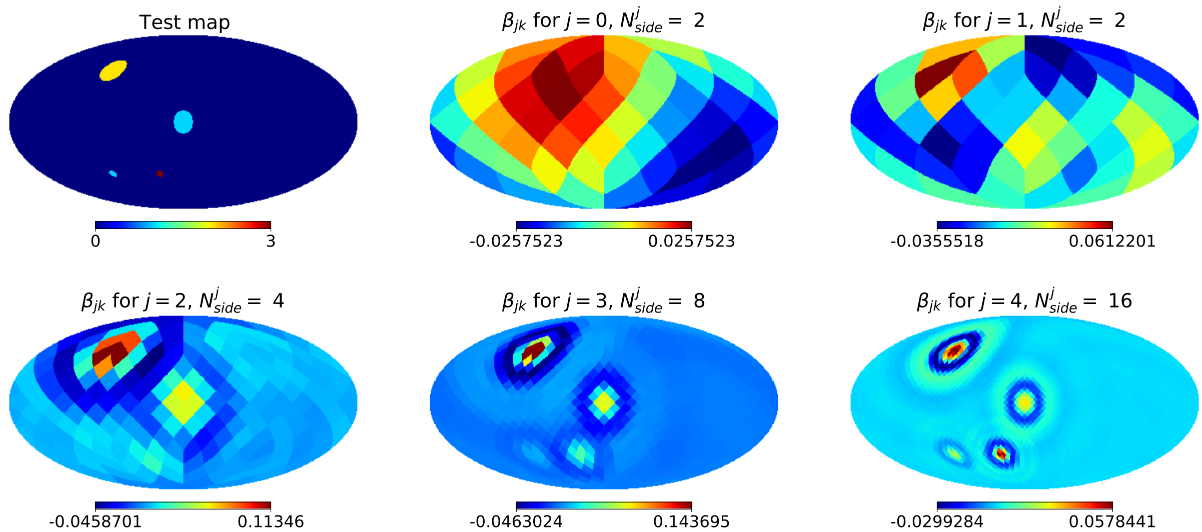

Figure 5: Coefficients plotted for the N^j_{side} given by equation 4. We can see that the resolution increases for higher j.

In figure 4, we've chosen to plot the needlet coefficients for the same N_{side} as the input map. This agrees well with our intuition; since computing the coefficients for small values of j corresponds to letting through low ℓ, the β_{jk} map contains the large scale features of the input map. As we evaluate β_{jk} for higher j, we are sliding the filter up to higher ℓ and so see the small scale features of the map.

We can also choose to plot the needlets for the N^j_{side} given by equation 4, as in figure 5. When we plot it this way, it's pretty clear exactly what $\mathcal{X}_j$ represents. In figure 6, we've plotted the $j = 0$ coefficient map with its corresponding N_{side^j}. Then, the k parameter sets which of these pixels the coefficient β_{jk}

corresponds to. The pixels corresponding to each k are labeled in the figure.

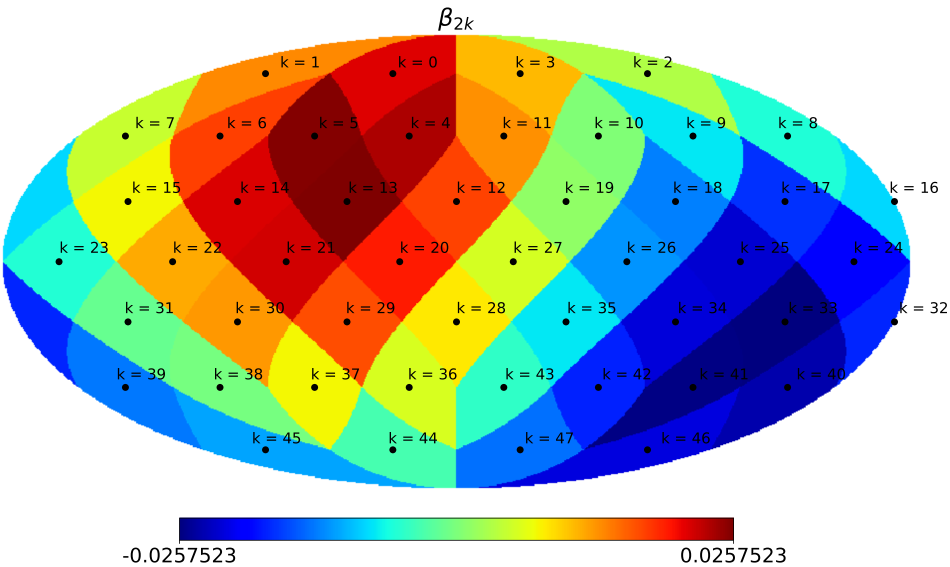

Figure 6: The relationship between k and the position on the grid, for $j = 2$.

2.2 The transform back into real space

Just to illustrate, let's say we're trying to get out a map of $N_{\text{side}} = 1$. If we use the HEALPix conventions, this corresponds to $\ell_{max} = 3N_{\text{side}} - 1 = 2$. Then, using equation 3, this corresponds to $j_{\max} = 2$ for $B = 2$. Using equation 4:

j	N^j_{side}	$\dim \mathcal{X}_j$
0	2	48
1	2	48
2	4	192

So, expanding equation 6,

$$
\begin{aligned}
T(\mathbf{r}) &= \sum_j \sum_k \beta_{jk} \psi_{jk}(\mathbf{r}) \\
&= \sum_{k=0}^{47} \beta_{0k} \psi_{0k}(\mathbf{r}) + \sum_{k=0}^{47} \beta_{1k} \psi_{1k}(\mathbf{r}) + \sum_{k=0}^{191} \beta_{2k} \psi_{2k}(\mathbf{r})
\end{aligned}
$$

Let's look at the $j = 0$ term only:

$$
\begin{aligned}
\sum_{k=0}^{47} \beta_{0k}\psi_{0k}(\mathbf{r}) &= \beta_{00}\psi_{00} + \beta_{01}\psi_{01} + \beta_{02}\psi_{02} + ... + \beta_{0,47}\psi_{0,47} \\
&= \beta_{00}\sqrt{\lambda_0}\sum_\ell \frac{2\ell+1}{4\pi} b\left(\frac{\ell}{B^0}\right) P_\ell(\boldsymbol{r}\cdot\boldsymbol{\xi}_{00}) + \beta_{01}\sqrt{\lambda_0}\sum_\ell \frac{2\ell+1}{4\pi} b\left(\frac{\ell}{B^0}\right) P_\ell(\boldsymbol{r}\cdot\boldsymbol{\xi}_{01}) + ...
\end{aligned}
$$

But this is kind of a terrible expression. Let's see if we can do better. What happens if we insert equations 2 and 7 into equation 6? Then,

$$
\begin{aligned}
T(\boldsymbol{r}) &= \sum_{jk} \beta_{jk}\psi_{jk}(\boldsymbol{r}) \\
&= \lambda_j \left[\sum_\ell b\left(\frac{\ell}{B^j}\right)\sum_m a_{\ell m} Y_{\ell m}(\boldsymbol{\xi}_{jk})\right]\left[\sum_\ell \frac{2\ell+1}{4\pi} b\left(\frac{\ell}{B^j}\right) P_\ell(\boldsymbol{r}\cdot\boldsymbol{\xi}_{jk})\right] \\
&= \lambda_j \sum_\ell \sum_m \frac{2\ell+1}{4\pi} b^2\left(\frac{\ell}{B^j}\right) P_\ell(\boldsymbol{r}\cdot\boldsymbol{\xi}_{jk}) a_{\ell m} Y_{\ell m}(\boldsymbol{\xi}_{jk})
\end{aligned}
$$

Now if we note that all the other stuff inside the sum apart from $Y_{\ell m}$ and $a_{\ell m}$ is just a function of ℓ, we can write

$$
T(\boldsymbol{r}) = \sum_{jk} \lambda_j \underbrace{\sum_{\ell m} f_{jk}(\ell) a_{\ell m} Y_{\ell m}(\boldsymbol{\xi}_{jk})}_{\text{inverse spherical harmonic transform}} \tag{8}
$$

and this is just an inverse spherical harmonic transform, with the $a_{\ell m}$ multiplied by

$$
f_{jk}(\ell) \equiv \frac{2\ell+1}{4\pi} b^2\left(\frac{\ell}{B^j}\right) P_\ell(\boldsymbol{r}\cdot\boldsymbol{\xi}_{jk}) \tag{9}
$$

an operation that can be done quickly with the healpy function almxfl. Although we still have to sum over j and k to do the transform back into real space, this is quite speedier than the brute force way.