

Searching for isolated pulsars using Einstein@Home: Setting up S4R2

Badri Krishnan, Ben Owen, Reinhard Prix, and Alicia Sintes

(Dated: 01 Dec, 2008)

(LIGO-T080340-00-Z)

This is a working document for figuring out how best to use Einstein@Home to search for gravitational waves from unknown isolated pulsars. This study formed the basis of the workunit setup implemented in the E@H run 'S4R2'.

I. NUMBER OF TEMPLATES USED BY LALAPPS_COMPUTEFSSTATISTIC

In order to optimize an E@H search using the current `lalapps_ComputeFStatistic` code (CFS), we need an expression for the *actual* number of templates as a function of the search parameters. We will base our numbers on the case where a metric-grid is used (`--gridType=2`).

The method employed by the current CFS is not covering the full parameter-space by a metric grid, but instead it only covers the sky by a metric grid (using `TwoDMesh()`) with the metric projected on the surface $f = f_{\max}$. In the spin-directions $f, \dot{f}, \ddot{f}, \dots$, on the other hand, only the (constant) metric-spacing in this coordinate-direction is used to determine a constant step-size.

This allows us to obtain an analytic expression for the number of templates in all directions except for the sky-grid, for which an empirical fit to the number of sky-points will be determined.

The (constant) stepsizes df and $d\dot{f}$ at a given mismatch argument m (`--metricMismatch=m`, **Note:** this is different from μ_{\max} !) in CFS are determined as follows. First, the frequency-axis is scanned at a constant stepsize of

$$df = 2\sqrt{\frac{m}{g_{ff}}}, \quad (1)$$

in terms of the parameter-space metric g_{ab} .

Because no correlations with the remaining parameters have been taken into account in this scan, we search the remaining subspace $\{\alpha, \delta, \dot{f}\}$ using the projected metric γ_{ab} onto the subspace of constant f , i.e.

$$\gamma_{ab} = g_{ab} - \frac{g_{af}g_{bf}}{g_{ff}}. \quad (2)$$

This metric is used in the sky-covering (using `TwoDMesh()`), and also in the independently covered spindown-direction, i.e.

$$d\dot{f} = 2\sqrt{\frac{m}{\gamma_{\dot{f}\dot{f}}}}. \quad (3)$$

The spin-components of the metric can be obtained as

$$g_{ss'} = (2\pi)^2 \frac{T^{s+s'+2}}{s!s'!(s+2)(s'+2)(s+s'+3)}, \quad (4)$$

and so the relevant diagonal elements are

$$g_{ff} = (2\pi)^2 \frac{T^2}{12} \quad \gamma_{\dot{f}\dot{f}} = (2\pi)^2 \frac{T^4}{16 \times 45}. \quad (5)$$

The resulting spacings are found as

$$df = \frac{2\sqrt{3m}}{\pi} \frac{1}{T}, \quad d\dot{f} = \frac{12\sqrt{5m}}{\pi} \frac{1}{T^2}. \quad (6)$$

We are using constant upper and lower boundaries for the frequency- and spindown ranges, i.e. $f \in [f_{\min}, f_{\min} + \Delta f]$ and $\dot{f} \in [\dot{f}_{\min}, \dot{f}_{\min} + \Delta \dot{f}]$, and so the total number of spin-templates N_{spin} is

$$N_{\text{spin}} = \frac{\pi^2 \Delta f \Delta \dot{f}}{24\sqrt{15}m} T^3. \quad (7)$$

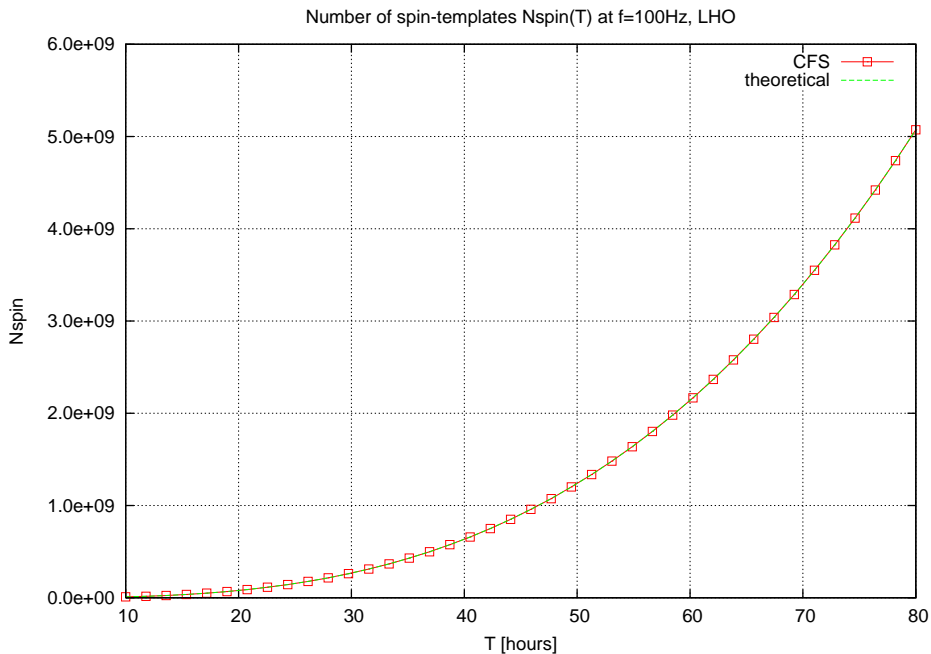


FIG. 1: Comparison between theoretically predicted and measured number of spin-templates.

In order to check this analytic relation, we compare it to the actual number of spin-templates produced by the grid-code using in CFS, the result of which is shown in Fig. 1.

The remaining unknown is the number of sky-templates $N_{\text{sky}}(T; f, m)$ (for the whole sky) at a fixed f, m . The dependencies of N_{sky} on f, m and can be made explicit based on the theoretical scalings, namely

$$N_{\text{sky}}(T; f, m) = \left(\frac{f}{f_0}\right)^2 \left(\frac{m_0}{m}\right) N_{\text{sky}}(T; f_0, m_0), \quad (8)$$

so we only need to determine an empirical fit to N_{sky} at fixed frequency f_0 and mismatch-parameter m_0 for an all-sky search.

We checked these theoretical scaling against the actual numbers, as shown in Fig. 2.

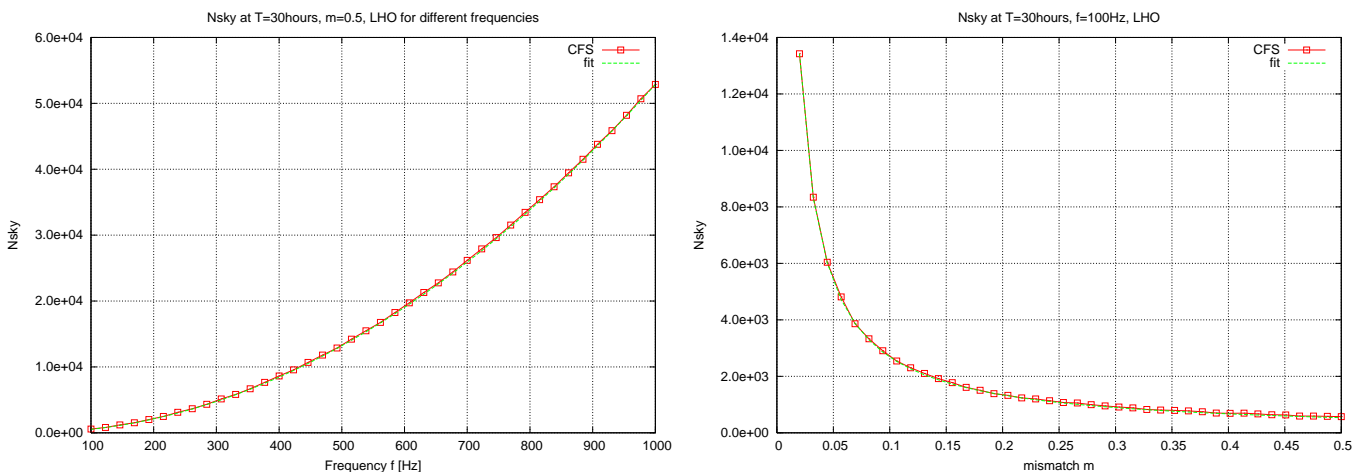


FIG. 2: Scaling of N_{sky} with frequency f and with mismatch m .

Unfortunately, the corresponding number of sky-templates produced by `TwoDMesh()` turns out to depend significantly on the start-time of the observation, see Fig. 3.

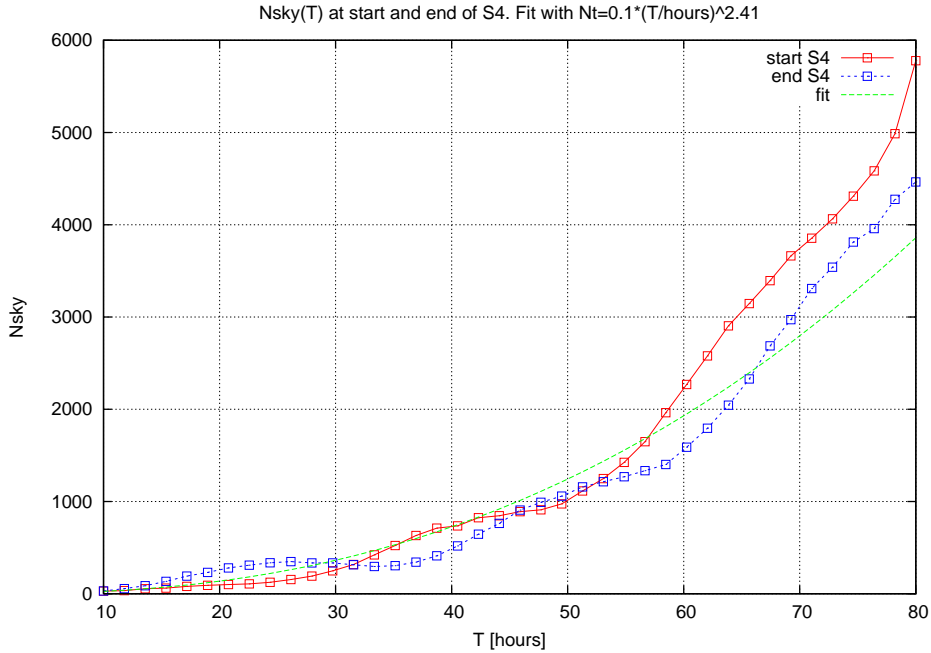


FIG. 3: Number of sky-templates N_{sky} as a function of observation-time (in hours) for startTime at the beginning of S4 ($t_0 = 792576013$) and at the end ($t_0 = 795830413$). The green line shows a common fit of the data by a power law.

However, in the following we will work with a rough power-law fit to provide a representative description of the overall behaviour of $N_{\text{sky}}(T)$, but obviously the exact number will depend on the start-time of the observation, so this fit should be considered only as an approximation to within some 10% errors or so up to $T \leq 55$ hours. Thus, we try a fit of the form $N_{\text{sky}} = AT^\lambda$. The best least-squares fit is shown in figure 3, and is found as:

$$N_{\text{sky}}(T; f_0 = 100 \text{ Hz}, m = 0.5) = AT^\lambda = 0.1 \left(\frac{T}{T_{\text{hour}}} \right)^{2.41}, \quad (9)$$

where $T_{\text{hour}} = 1$ hour, and so

$$N_{\text{sky}}(T; f, m) = \frac{f^2}{m} \frac{m_0 A}{f_0^2} T^\lambda, \quad (10)$$

with $f_0 = 100$ Hz, $m_0 = 0.5$ and $\lambda \approx 2.41$, $A \approx 0.1/(T_{\text{hour}})^\lambda$.

Putting the pieces together, and using the fact that the sky-grid is to be evaluated at the highest frequency f_{max} in the frequency-interval Δf , we obtain the total number of templates N_p as

$$N_p = N_{\text{spin}} N_{\text{sky}}(f_{\text{max}}) = \frac{\pi^2}{24\sqrt{15}} \frac{m_0 A}{f_0^2} \Delta f \Delta f f_{\text{max}}^2 m^{-2} T^{3+\lambda}. \quad (11)$$

Note that in the case of orthogonal search-axes we would expect the maximal mismatch μ_{max} occurring to be roughly given by $\mu_{\text{max}} \approx D \times m$, where D is the number of dimensions and m is the maximal mismatch along each axis. Due to the practical implementation of the search-grid, however, it turns out that (see http://www.aei.mpg.de/~repr/EnoteEntries/MetricGrids_Report/Report_CFS4D.pdf) the maximal mismatch seems roughly given by

$$\mu_{\text{max}} \approx 1.25 m, \quad (12)$$

where m corresponds to the parameter specified to CFS as `--metricMismatch=m`.

II. THE LARGEST POSSIBLE COHERENT INTEGRATION

To run a pulsar search on Einstein@home, we need to break up the parameter space suitably. Let us first consider the simplest way of doing this, i.e. only divide the frequency range into sub-bands. Thus, each work-unit gets the

entire sky, and the entire range of spin-down parameters $\Delta\dot{f}$, but gets frequency bands of possibly different size. We want to check whether such a procedure is practical in the sense that the following constraints must be satisfied:

- Each work unit must take roughly the same time to finish.
- The amount of data to be transferred with each work unit must not be too large.
- Each work unit must analyse at least 100 frequency and spindown values for every sky-location – this is when `ComputeFStatistic` is computationally efficient.
- The break up of the parameter space should be as simple as possible!

Let N_{CPU} be the *effective* number of CPUs available, i.e. the average equivalent number of CPU's running E@H all the time. τ_{tot} denotes the total amount of time the project can run, and τ_{WU} the time for each work unit to finish on a CPU. Then, the number of work units is given by

$$N_{WU} = \frac{N_{CPU} \tau_{tot}}{\tau_{WU}}. \quad (13)$$

Let Δf_i ($i = 1 \dots N_{WU}$) be the frequency bands given to each work-unit and let f_i be the highest frequency value of the i^{th} work unit. Thus, the i^{th} work unit gets the frequency band $(f_i - \Delta f_i, f_i)$. Let f_{min} be the smallest frequency considered in the whole search and f_{max} the largest. Let $\Delta\dot{f}_i$ be the range of first spindown parameter values considered in the i^{th} work unit. We are considering spindowns in the range $(-\Delta\dot{f}_i, \Delta\dot{f}_i)$. We shall consider two cases:

- $\Delta\dot{f}_i$ is independent of i and equal to $2f_{max}/\tau_{min}$ for a given minimum spindown age τ_{min} .
- $\Delta\dot{f}_i = 2f_i/\tau_{min}$.

Let us first focus on case i. Let T be the total amount of data available. We are assuming that the duration spanned by the data T_{span} and the actual amount of data T_{data} are the same. This requirement can be relaxed but for simplicity, let us work with this for now.

From the template counting formulae (11) shown earlier, we have that the number of templates analysed by a single work unit is

$$N_p = k_1 T^{3+\lambda} f_i^2 \Delta f_i \Delta\dot{f} m^{-2} \quad (14)$$

where

$$k_1 = \frac{\pi^2}{24\sqrt{15}} \frac{m_0 A}{f_0^2}. \quad (15)$$

We would like to keep the time spent on each work unit to be the same. Thus, we must require that

$$f_i^2 \Delta f_i = C \quad \text{for} \quad i = 1 \dots N_{WU}, \quad (16)$$

where C is a constant independent of i . If the frequency bands are small enough, this can be rewritten as

$$\Delta \left(\frac{f_i^3}{3} \right) = C. \quad (17)$$

Summing over $i = 1 \dots N_{WU}$, we get

$$f_{max}^3 - f_{min}^3 = 3CN_{WU}. \quad (18)$$

In the limit where $N_{spin} \gtrsim 100$, the time taken to complete one template asymptotes to the approximate expression

$$\tau_p = k_2 T, \quad (19)$$

where k_2 is a constant. Since τ_p is proportional to the number of SFTs M , we must have

$$k_2 = \frac{\tau_{p0}}{M_0 T_{SFT}} \quad (20)$$

where τ_{p0} and M_0 are reference values for τ_p and M . The time τ_{WU} to complete one workunit is therefore $\tau_{\text{WU}} = \tau_p N_p$, and substituting into equation (13) using (14) we get

$$\Delta f \frac{f_{\text{max}}^3 - f_{\text{min}}^3}{3} = \frac{N_{\text{CPU}} \tau_{\text{tot}}}{k_1 k_2 T^{4+\lambda}} m^2 \quad \implies \quad T = \left(\frac{3 N_{\text{CPU}} \tau_{\text{tot}} \tau_{\text{min}} m^2}{2 k_1 k_2 f_{\text{max}} (f_{\text{max}}^3 - f_{\text{min}}^3)} \right)^{1/(4+\lambda)}. \quad (21)$$

We need to check if this solution is consistent with the constraints listed at the start of this section. To do this, assume that τ_{WU} is some reasonable number, say 12 h. Then, from (13), we can calculate N_{WU} , and thus

$$C = (f_{\text{max}}^3 - f_{\text{min}}^3) / 3 N_{\text{WU}} \quad (22)$$

Thus, the smallest and largest frequency bands are respectively

$$\Delta f_{N_{\text{WU}}} = C / f_{\text{max}}^2 \quad \text{and} \quad \Delta f_1 = C / f_{\text{min}}^2. \quad (23)$$

Thus, the maximum number of bins is TC / f_{min}^2 and the minimum is TC / f_{max}^2 . Both of these must satisfy the constraints listed earlier.

Let us now consider case ii. when the spindown range corresponds to a fixed τ_{min} and thus changes with frequency. In this case, $\Delta \dot{f}_i = 2f_i / \tau_{\text{min}}$ so that $N_p^i \propto f_i^3 \Delta f_i$. To require each work unit to analyse the same amount of templates, we would require:

$$f_i^3 \Delta f_1 = C' \quad \implies \quad f_{\text{max}}^4 - f_{\text{min}}^4 = 4C' N_{\text{WU}}. \quad (24)$$

Following the same steps as before, we arrive at the solution for T :

$$T = \left(\frac{2 N_{\text{CPU}} \tau_{\text{tot}} \tau_{\text{min}} m^2}{k_1 k_2 (f_{\text{max}}^4 - f_{\text{min}}^4)} \right)^{1/(4+\lambda)}. \quad (25)$$

The frequency bands decrease as $1/f^3$ in this case:

$$\Delta f_{N_{\text{WU}}} = C' / f_{\text{max}}^3 \quad \text{and} \quad \Delta f_1 = C' / f_{\text{min}}^3. \quad (26)$$

These results are made concrete below.

A. Illustrative example

As an illustrative example, we consider an all-sky search and we take

- $N_{\text{CPU}} = 50,000$
- $f_{\text{min}} = 100 \text{ Hz}$, $f_{\text{max}} = 600 \text{ Hz}$
- $\tau_{\text{min}} = 1000 \text{ yr}$ (so that at 600 Hz, $\dot{f}_{\text{max}} = f / \tau_{\text{min}} = 1.9 \times 10^{-8} \text{ Hz/s}$ and we search over the range $(-\dot{f}_{\text{max}}, \dot{f}_{\text{max}})$)
- $m = 0.3$
- $\tau_{\text{tot}} = 3 \text{ months}$
- $\tau_{\text{WU}} = 12 \text{ h}$.
- $\tau_{p0} = 2.5 \times 10^{-5} \text{ sec}$ ($k_2 = \tau_{p0} / (20 * 1800)$)

With these parameters, solving eq. (21), we see that Einstein@Home can search over $T \approx 56.9$ hours of data. Each work unit analyses $\sim 3 \times 10^8$ templates. The other features of this search are the following.

As mentioned earlier, we only break up the parameter space in frequency with the frequency spacing decreasing quadratically with increasing frequency. This ensures that all work-units take approximately the same time to finish. The total number of work units is 9×10^6 (thus the frequency band from 100-600 Hz has been broken up into 9 million bands). The smallest frequency band is at 600 Hz and it has a width of $2.2 \times 10^{-5} \text{ Hz}$ corresponding to ~ 5 bins at the frequency resolution of $1/T$. The largest frequency band is at 100 Hz and it has a width of $7.96 \times 10^{-4} \text{ Hz}$ corresponding to ~ 163 bins.

Are the constraints satisfied in this solution? First, we want that the frequency and spindown bands are large enough so that we do at least ~ 100 frequency/spindown values for each sky position. This means that N_{spin} defined in 7 must be at least 100. It is easy to calculate that at 600 Hz, we get $N_{\text{spin}} = 3571$, which is comfortably larger than 100. The other constraint is that the data which must be sent with each work unit is not too large. In this case, the largest work units are at 100 Hz. The frequency band, as we saw, is $\Delta f_1 = 7.96 \times 10^{-4}$ Hz. This corresponds to $1800\Delta f_1 = 1.43$, i.e. 2 frequency bins in every SFT. Assuming each SFT bin is 8 bytes, the amount of data which must be sent in the form of SFTs is $8 \times 2 \times N_{\text{SFT}} \sim 16T/T_{\text{SFT}} \sim 1800$ bytes. Each SFT must also contain ‘‘wings’’ corresponding to the doppler width, i.e. $2f_iv/c$. This is ~ 0.12 Hz at 600 Hz, and 0.02 Hz at 100 Hz.

If we perform the same search as above, except that we search over a fixed $\tau_{\text{min}} = 1000$ y instead of a constant f range, the answers do not change qualitatively. We get $T = 59.6$ hours, each work unit analysing 2.9×10^8 templates. Also, we have $\Delta f_1 = 3.6 \times 10^{-3}$ Hz ($N_{\text{spin}} \sim 8 \times 10^4$), and at the maximum frequency: $\Delta f_{N_{\text{WU}}} = 1.67 \times 10^{-5}$ Hz ($N_{\text{spin}} = 2218$). The constraints are still satisfied.

As discussed in section III below, it is possible to choose an optimal value for the mismatch m based on optimizing the sensitivity of the search. This leads to a range of values and 0.3, as chosen above, is on the conservative side. If we were to choose instead $m = 0.7$, then we would get $T = 74.7$ h with the constraints still satisfied. More generally, $T \propto m^{2/(4+\lambda)}$.

III. THE OPTIMAL THRESHOLD F_{th} AND MAXIMAL MISMATCH μ

A. False alarm and minimal threshold \mathcal{F}_{th}

Another practical constraint limiting the sensitivity of our search is the maximum number of candidates we can allow to be sent back from each node/CPU. This is limited on one hand by the maximum reasonable data upload-rate Γ_{up} , either determined by the rate a user can send back via a standard DSL-connection, and on the other hand by the maximum data-flow our servers can handle. A related secondary question might also be how many candidates in total we want/can deal with after this whole coherent step of the first stage.

This maximum number of candidates per WU (giving the false-alarm rate) determines the minimum threshold on \mathcal{F} we can set, which in turn limits the loudest possible signal which can pass this first stage, and thereby the maximum sensitivity of the whole pipeline.

The maximum average number of candidates N_c per WU is therefore given by

$$N_c \sim \frac{\Gamma_{\text{up}}}{m_c} \tau_{\text{WU}}, \quad (27)$$

where m_c is the amount of memory needed for one candidate. On the other hand, this maximum number N_c determines the lowest false-alarm probability α we can allow, namely

$$\alpha = \frac{N_c}{N_p}. \quad (28)$$

This upper limit on α is an important difference from the analysis of [1]. There it was assumed that one would set the threshold based on the false dismissal rate $\beta(\mathcal{F}_{\text{th}}, h_0)$, such we don’t lose more than, say, 10% of signals with a certain amplitude h_0 . The threshold was then determined by inverting the equation $\beta(\mathcal{F}_{\text{th}}, h_0) = 0.1$. This was used to determine the false alarm rate $\alpha(\mathcal{F}_{\text{th}})$. Now it is different: we have to accept a false alarm rate based on the above constraint and this determines the sensitivity of the search.

On the other hand, for Gaussian noise the false alarm α is related to the threshold \mathcal{F}_{th} by

$$\alpha = (1 + \mathcal{F}_{\text{th}}) e^{-\mathcal{F}_{\text{th}}}. \quad (29)$$

Using Eqs. (27) and (28) together with the relation $N_p = \tau_{\text{WU}}/\tau_p(T)$, we obtain

$$\tau_p(T) \frac{\Gamma_{\text{up}}}{m_c} = (1 + \mathcal{F}_{\text{th}}) e^{-\mathcal{F}_{\text{th}}}, \quad (30)$$

which determines the lowest feasible threshold in terms of the maximal upload-rate and the time per template $\tau_p(T)$. It is interesting to note that this threshold does not depend on any other details of the search.

Taking the above example, and requiring the maximum upload-rate $\Gamma_{\text{up}} = 300$ kB/24hours, assuming the memory per candidate $m_c = 30$ Bytes, then we get from the above: $\alpha = 1.65 \times 10^{-5}$ and $\mathcal{F}_{\text{th}} = 11.0$.

B. Optimal mismatch μ_{\max}

In the next step we want to optimize the sensitivity of the search in terms of the maximal mismatch μ_{\max} of the search-grid. First we fix a false-dismissal rate β for our search. For gaussian noise the pdf for \mathcal{F} in the presence of a signal-to-noise ratio (squared) $\eta = (h|h)$ is given by $p_1(\mathcal{F}; \eta) = 2\chi_{(4)}^2(2\mathcal{F}; \eta)$, which determines the false-dismissal rate

$$\beta(\mathcal{F}_{\text{th}}, \eta) = \int_0^{\mathcal{F}_{\text{th}}} p_1(\mathcal{F}; \eta) d\mathcal{F}. \quad (31)$$

One can show generally that for $\chi_{(n)}^2(z; \eta)$ -distributions the corresponding false-dismissal rate β is a monotonically decreasing function of the signal-to-noise ratio η , i.e. $\partial_\eta \beta < 0$, and therefore fixing a maximal β implies a smallest η which can be detected with the required confidence. On the other other hand, using the expression derived in JKS for the *averaged* SNR, we can write η in terms of the signal-amplitude h_0 as

$$\eta = (1 - \mu_{\max}) \frac{4}{25} \frac{h_0^2 T(\mu_{\max})}{S_n}, \quad (32)$$

where the prefactor $1 - \mu_{\max}$ accounts for the signal-to-noise ratio loss at the “worst” parameter-space points with maximal distance from the grid-points. Note that we could also (consistently with the rest of the SNR-expression) chose an “average” mismatch-loss (which for a cubic lattice would be $(1 - \mu_{\max}/3)$), and ideally we should evaluate the false-dismissal rate over the search-grid.

Using Eq. (21), which determines the longest-possible observation time T , and using $m \sim \mu_{\max}$, we see that

$$T \propto \mu_{\max}^{2/(4+\lambda)}, \quad (33)$$

and so we can express the smallest signal-amplitude h_0 , i.e. the sensitivity, as

$$h_0^{-2} \propto (1 - \mu_{\max}) \mu_{\max}^{2/(4+\lambda)}, \quad (34)$$

which has its maximum at

$$\mu_{\max} = (3 + \lambda/2)^{-1} \approx 0.24. \quad (35)$$

Note that if we had used the approximate *average* loss in signal-to-noise η due to template-mismatch $(1 - \mu_{\max}/3)$, we would find $\mu_{\max} = (1 + \lambda/6)^{-1} \approx 0.71$.

[1] C. Cutler, I. Gholami, and B. Krishnan. Improved stack-slide searches for gravitational-wave pulsars. *Phys. Rev. D.*, 72(4): 042004, Aug. 2005. doi: 10.1103/PhysRevD.72.042004.