Resampling FFT-based implementation of the $\mathcal{F}$-statistic

Reinhard Prix and others ...[*]
Max-Planck-Institut für Gravitationsphysik, Albert-Einstein-Institut, D-30167 Hannover, Germany
(Dated: 2017-04-04 18:31:48 +0200; commitID: 84927ee-CLEAN; LIGO-T1600531-v3)

Working notes on Resampling FFT-based Fstat calculation. Previous work in this [1, 2]

I. CHARACTERIZING THE RESAMPLING-FFT $\mathcal{F}$-STATISTIC IMPLEMENTATION

A. Algorithm

There are essentially four different vector lengths over which various operations are performed in the Resampling-FFT $\mathcal{F}$-statistic code:

N_{Fbin}: The user-requested number of output frequency bins, with spacing df, for which the $\mathcal{F}$-statistic is computed.

$N_{\mathrm{samp}}^{\mathrm{det}}$: The (maximum over detectors) number of samples in the input timeseries (in the detector frame.

$N_{\mathrm{samp}}^{\mathrm{src}}$: The (maximum over detectors) number of samples of the input timeseries, interpolated into the source-frame.

$N_{\mathrm{samp}}^{\mathrm{FFT}}$: The length of the (zero-padded) timeseries on which the FFT is performed.

which are given by the following relations:

The sampling step dt_{det} of the input timeseries in the detector frame is given in terms of the frequency bandwidth Δf_{sft} of the input SFTs, namely

$$dt_{\mathrm{det}} \equiv \frac{1}{\Delta f_{\mathrm{sft}}}\,, \tag{1}$$

n and the number of detector-frame time-samples are therefore

$$N_{\mathrm{samp}}^{\mathrm{det}} = \frac{T_{\mathrm{span}}^{\mathrm{data}}}{dt_{\mathrm{det}}} = T_{\mathrm{span}}^{\mathrm{data}}\,\Delta f_{\mathrm{sft}}\,, \tag{2}$$

where $T_{\mathrm{span}}^{\mathrm{data}}$ is the (maximum over IFOs) maximum data span, which is determined by the SFTs actually used and will therefore be generally $T_{\mathrm{span}}^{\mathrm{data}} \leq T_{\mathrm{span}}$, where T_{span} is the user-requested (or via segment list) length of the data-stretch being analyzed. The length in time T_{FFT} of the final timeseries to be FFT'ed is determined by the user-requested frequency resolution df, namely

$$T_{\mathrm{FFT}} = \frac{1}{df/D}\,, \quad \text{with} \quad D \equiv \lceil T_{\mathrm{span}}\, df \rceil\,, \tag{3}$$

where $D \in \mathbb{N}^+$ ensures that $T_{\mathrm{FFT}} \geq T_{\mathrm{span}}^{\mathrm{data}}$, which is easily achieved by zero-padding. This is required in order to make sure we use all the data (zero-padding rather than truncating the timeseries) and therefore not lose SNR: in the 'standard case' of a fine frequency resolution $df < 1/T_{\mathrm{span}}$ we have $D = 1$. But in cases where the user requests a very coarse frequency resolution $df > 1/T_{\mathrm{span}}^{\mathrm{data}}$ we effectively increase the internal frequency resolution by an integer factor $D > 1$, which allows us to easily return the requested frequency bins by only returning every D'th output bin.

With the original detector-frame sampling step dt_{det} this would correspond to

$$N_{\mathrm{samp}}^{\mathrm{FFT}(0)} = \frac{T_{\mathrm{FFT}}}{dt_{\mathrm{det}}} = \frac{\Delta f_{\mathrm{sft}}}{df/D}\,, \tag{4}$$

time samples. However, in order to ensure the most efficient (and relatively most consistent) FFT performance, we round the number of FFT bins to the next power of 2, namely

$$N_{\mathrm{samp}}^{\mathrm{FFT}} = 2^{\lceil \log_2 N_{\mathrm{samp}}^{\mathrm{FFT}(0)} \rceil}\,, \tag{5}$$

[*] reinhard.prix@aei.mpg.de

which can be achieved by effectively decreasing the sampling step (increasing the frequency band) of the timeseries in the source frame, i.e.

$$dt_{\mathrm{src}} \equiv \frac{T_{\mathrm{FFT}}}{N_{\mathrm{samp}}^{\mathrm{FFT}}} \,. \tag{6}$$

The original time-series (without zero-padding) interpolated into the source-frame will therefore have $N_{\mathrm{samp}}^{\mathrm{src}}$ samples, which can be expressed as

$$N_{\mathrm{samp}}^{\mathrm{src}} \equiv \frac{T_{\mathrm{span}}^{\mathrm{data}}}{dt_{\mathrm{src}}} = \mathcal{R}\, N_{\mathrm{samp}}^{\mathrm{FFT}} \,, \tag{7}$$

where we defined the frequency resolution $\mathcal{R}$ in terms of the the natural units $1/T_{\mathrm{span}}$ as

$$\mathcal{R} \equiv \frac{T_{\mathrm{span}}^{\mathrm{data}}}{T_{\mathrm{FFT}}} = T_{\mathrm{span}}^{\mathrm{data}} \frac{df}{D} \leq 1 \,, \tag{8}$$

where $D = 1$ whenever $df < 1/T_{\mathrm{span}}^{\mathrm{data}}$, see Eq. (3). From Eq. (2) and Eq. (4) we find as similar relation to Eq. (7) for $N_{\mathrm{samp}}^{\mathrm{det}}$:

$$N_{\mathrm{samp}}^{\mathrm{det}} = \mathcal{R}\, N_{\mathrm{samp}}^{\mathrm{FFT}(0)} \,. \tag{9}$$

The input SFT frequency band contains extra "frequency drift" sidebands $\Delta f_{\mathrm{drift}}$ to account for Doppler shifts and spindowns (from `XLALCWSignalCoveringBand()`), and an extra "transition band" $\Delta f_{\mathrm{rolloff}}$ to allow for the roll-off from the Hamming-windowed sinc-interpolation used in Barycentric resampling. 16 extra SFT bins (8 on either side) to account for frequency "leakage" near the SFT edges. We can express this as

$$\Delta f_{\mathrm{sft}} = \Delta f_{\mathrm{load}} \left(1 + \frac{4}{2\,\mathrm{Dterms} + 1} \right) \,, \tag{10}$$

where

$$\Delta f_{\mathrm{load}} = \Delta f + \Delta f_{\mathrm{drift}} + \frac{16}{T_{\mathrm{sft}}} \,, \tag{11}$$

$$\Delta f_{\mathrm{drift}} = 2.12 \times 10^{-4} \left(f_{\mathrm{max}} + |\dot{f}|_{\mathrm{max}} \frac{T_{\mathrm{span}}}{2} + |\ddot{f}|_{\mathrm{max}} \frac{T_{\mathrm{span}}^2}{8} + \ldots \right) \,, \tag{12}$$

and Dterms is the (user-specified) number of sinc-kernel terms to use (on either side, therefore the window-size is $2\,\mathrm{Dterms} + 1$) in the barycentric resampling interpolation.

Note that these expressions are only approximative of what *exactly* happens in the F-stat codes, and due to the power-of-two rounding of Eq. (5), this can in some cases lead to relatively large deviations from the actual measured runtimes. Use the octapps function `predictResampTimeAndMemory()` for the most accurate (still not perfect, due to numerical rounding issues) runtime prediction as a function of physical search parameters.

The "physical" number of requested output frequency bins is simply

$$N_{\mathrm{Fbin}} = \frac{\Delta f}{df} \,. \tag{13}$$

Note that for the GCT code Δf is not just the user-input search frequency band Δf_0, but includes additional GCT sideband bins (called `extraBinsFstat` in the GCT code) as well, namely

$$\Delta f = \Delta f_0 + \frac{1}{2} \left(T_{\mathrm{span}}\, d\dot{f} + T_{\mathrm{span}}^2\, d\ddot{f} \right) \,. \tag{14}$$

B. Timing model

We can break the implementation of the Resampling $\mathcal{F}$-statistic calculation into the following contributions with respective scalings (referring to the time to compute an output vector of N_{Fbin} $\mathcal{F}$-statistic values *from a single detector*:

1. Barycenter and interpolate the input detector-frame time-series into the source frame (accounting for Doppler-shifts due to the Earth's motion and any binary orbital motion of the source). This contribution contains terms that scale with both input samples of the actual SFTs (in case of gaps), as well as $N^{\mathrm{src}}_{\mathrm{samp}}$ and $N^{\mathrm{det}}_{\mathrm{samp}}$, but we can postulate a rough scaling relation as

$$\mathcal{T}_{\mathrm{bary}} \sim N^{\mathrm{src}}_{\mathrm{samp}}\, \tau_{\mathrm{bary}} = \mathcal{R}\, N^{\mathrm{FFT}}_{\mathrm{samp}}\, \tau_{\mathrm{bary}}\,. \tag{15}$$

Note that barycentering will only be performed once per sky-position and binary-orbital parameters, while re-using the results for all different $\dot{f}$ and $\ddot{f}$ bins via buffering, and so its average contribution will scale with

$$b = 1/N_{\{\dot{f},\ddot{f},\ldots\}} \in (0,1]\,,$$

where $N_{\{\dot{f},\ddot{f},\ldots\}}$ is the total number of spindown-templates (of any order) per sky- and binary-orbital template. For many cases of practical interest it will be possible to achieve $b \ll 1$ and in this case one can neglect the barycentering contribution.

2. Apply spindown-correction and frequency shift to source-frame timeseries, which clearly scales with $N^{\mathrm{src}}_{\mathrm{samp}}$, and therefore

$$\mathcal{T}_{\mathrm{spin}} = N^{\mathrm{src}}_{\mathrm{samp}}\, \tau_{\mathrm{spin}} = \mathcal{R}\, N^{\mathrm{FFT}}_{\mathrm{samp}}\, \tau_{\mathrm{spin}}\,. \tag{16}$$

3. Apply the FFT, which operates on the zero-padded timeseries of length $N^{\mathrm{FFT}}_{\mathrm{samp}}$, therefore

$$\mathcal{T}_{\mathrm{FFT}} = N^{\mathrm{FFT}}_{\mathrm{samp}}\, \tau_{\mathrm{FFT}}\,. \tag{17}$$

4. various operations on N_{Fbin} output bins, such as copying the bins, normalizing them, computing $\mathcal{F}$ from F_a, F_b and summing them over detectors $F_{a,b} = \sum_X F^X_{a,b}$ (here considering time per detector), which can be summarized as a time contribution per output frequency bin τ_{Fbin},

$$\tau_{\mathrm{Fbin}} = \left(\mathcal{T}_{\mathrm{copy}} + \mathcal{T}_{\mathrm{norm}} + \mathcal{T}_{\mathrm{sumFabX}} + \mathcal{T}_{\mathrm{Fab2F}}\right)/N_{\mathrm{Fbin}}\,. \tag{18}$$

Let us denote the total time spent computing N_{Fbin} output bins as $\mathcal{T}_{\mathrm{total}}$, and the pure "resampling" contribution $\mathcal{T}_{\mathrm{RS}}$ excluding barycentering as

$$\mathcal{T}_{\mathrm{RS}} \equiv \mathcal{T}_{\mathrm{total}} - b\,\mathcal{T}_{\mathrm{bary}}\,. \tag{19}$$

We can combine this to yield the resampling time τ_{RS} per detector per output frequency bin *assuming perfect buffering* as

$$\tau_{\mathrm{RS}} \equiv \frac{\mathcal{T}_{\mathrm{RS}}}{N_{\mathrm{Fbin}}} = \tau_{\mathrm{Fbin}} + \frac{N^{\mathrm{FFT}}_{\mathrm{samp}}}{N_{\mathrm{Fbin}}}\left[\mathcal{R}\tau_{\mathrm{spin}} + \tau_{\mathrm{FFT}}\right]\,, \tag{20}$$

which will only be constant for different search setups if the ratio $N^{\mathrm{FFT}}_{\mathrm{samp}}/N_{\mathrm{Fbin}}$ and the natural resolution $\mathcal{R}$ are approximately constant. The total time to compute the $\mathcal{F}$-statistic for one detector and N_{Fbin} output frequency bins is

$$\mathcal{T}_{\mathrm{total}} = N_{\mathrm{Fbin}}\, \tau^{\mathrm{eff}}_{\mathrm{RS}} = N_{\mathrm{Fbin}}\left(\tau_{\mathrm{RS}} + b\,\Delta\tau^{\mathrm{bary}}_{\mathrm{RS}}\right)\,, \tag{21}$$

where $\Delta\tau^{\mathrm{bary}}_{\mathrm{RS}}$ is the time per frequency bin spent in barycentering, which is

$$\Delta\tau^{\mathrm{bary}}_{\mathrm{RS}} \equiv \frac{\mathcal{T}_{\mathrm{bary}}}{N_{\mathrm{Fbin}}} = \frac{N^{\mathrm{FFT}}_{\mathrm{samp}}}{N_{\mathrm{Fbin}}}\,\mathcal{R}\,\tau_{\mathrm{bary}}\,, \tag{22}$$

and so using this together with Eq. (20) we can express the effective resampling time $\tau^{\mathrm{eff}}_{\mathrm{RS}}$ per output frequency bin (including the barycentering cost) as

$$\tau^{\mathrm{eff}}_{\mathrm{RS}} \equiv \frac{\mathcal{T}_{\mathrm{total}}}{N_{\mathrm{Fbin}}} = \tau_{\mathrm{RS}} + b\,\Delta\tau^{\mathrm{bary}}_{\mathrm{RS}} = \tau_{\mathrm{Fbin}} + \frac{N^{\mathrm{FFT}}_{\mathrm{samp}}}{N_{\mathrm{Fbin}}}\left[\mathcal{R}\left(\tau_{\mathrm{spin}} + b\,\tau_{\mathrm{bary}}\right) + \tau_{\mathrm{FFT}}\right]\,, \tag{23}$$

which asymptotes to the "fully-buffered" value τ_{RS} of Eq. (20) for $b \to 0$, i.e. for $N_{\{\dot{f},\ddot{f},\ldots\}} \gg 1$.

C. Memory model for a multi-segment search

In a similar way to the timing model we can enumerate the amount of memory required to perform the resampling+FFT calculation of the $\mathcal{F}$-statistic over N_{seg} coherent segments.

- **Objects stored for every segment of a semi-coherent search:**
  - the original SFTs turned into a detector-frame `COMPLEX8` timeseries:

$$\text{mem}\left[\texttt{multiCOMPLEX8TimeSeries-DET}\right] = N_{\text{det}}\, N_{\text{samp}}^{\text{det}} \times \text{mem}\left[\texttt{C8}\right]\ ,$$

    where for simplicity we assumes the timeseries of all detectors to be of similar length $N_{\text{samp}}^{\text{det}}$.
  - two timeseries interpolated and barycentered into the source frame, multiplied by $a(t)$ or $b(t)$, respectively:

$$\text{mem}\left[\texttt{multiCOMPLEX8TimeSeries-SRC-[a|b]}\right] = 2\, N_{\text{det}}\, N_{\text{samp}}^{\text{src}} \times \text{mem}\left[\texttt{C8}\right]\ .$$

- **Objects stored only once for all segments, using a shared "workspace":**
  - Two `COMPLEX8` vectors for temporary storage of SRC-frame timeseries:

$$\text{mem}\left[\texttt{TStmp[1|2]-SRC}\right] = 2N_{\text{samp}}^{\text{src}} \times \text{mem}\left[\texttt{C8}\right]\ ,$$

  - One `REAL8` timeseries holding time-differences between SRC and DET frames (only used for barycentering:

$$\text{mem}\left[\texttt{SRCtimes-DET}\right] = N_{\text{samp}}^{\text{src}} \times \text{mem}\left[\texttt{R8}\right]\ .$$

  - the FFTW plan: I couldn't find a clear statement on the memory size of this, but I'm assuming it should be roughly $N_{\text{samp}}^{\text{FFT}}$ `COMPLEX8` numbers?

$$\text{mem}\left[\texttt{FFT-plan}\right] \approx N_{\text{samp}}^{\text{FFT}} \times \text{mem}\left[\texttt{C8}\right]\ .$$

  - the input and output vectors for the FFT (currently *not* using in-place transform):

$$\text{mem}\left[\texttt{FFT-input+output-vectors}\right] = 2\, N_{\text{samp}}^{\text{FFT}} \times \text{mem}\left[\texttt{C8}\right]\ .$$

  - temporary storage of $F_{a,b}^{X}$ and $F_{a,b}$ until no longer needed:

$$\text{mem}\left[\texttt{Fab}\right] = 4\, N_{\text{Fbin}} \times \text{mem}\left[\texttt{C8}\right]\ .$$

- **Objects returned from each $\mathcal{F}$-stat call:** (depending on user-request)

$$\text{mem}\left[\mathcal{F}\texttt{-stats return}\right] = N_{\text{Fbin}} \times \text{mem}\left[\texttt{R4}\right]\ ,$$

$$\text{mem}\left[\texttt{single-IFO }\mathcal{F}^{X}\texttt{ return}\right] = N_{\text{det}} N_{\text{Fbin}} \times \text{mem}\left[\texttt{R4}\right]\ .$$

where $\text{mem}\left[\texttt{C8}\right] = \text{mem}\left[\texttt{R8}\right] = 8\,\text{bytes}$ and $\text{mem}\left[\texttt{R4}\right] = 4\,\text{bytes}$.
We can therefore express the different memory blocks as

$$\text{mem}\left[\texttt{ResampWorkspace}\right] = \left[3N_{\text{samp}}^{\text{FFT}}\left(1+\mathcal{R}\right) + 4N_{\text{Fbin}}\right] \times 8\,\text{bytes}\,, \tag{24}$$

$$\text{mem}\left[\texttt{ResampMethodData-all}\right] = N_{\text{seg}}\, N_{\text{det}} \mathcal{R}\left[N_{\text{samp}}^{\text{FFT}(0)} + 2N_{\text{samp}}^{\text{FFT}}\right] \times 8\text{bytes}\,, \tag{25}$$

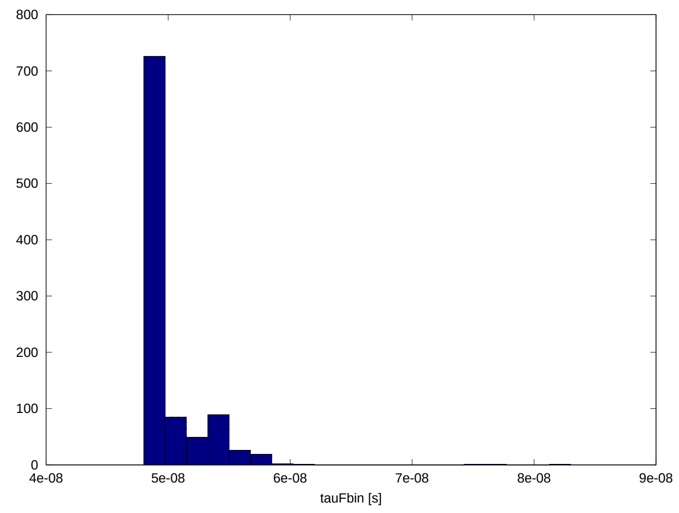
$$\text{mem}\left[\mathcal{F}\texttt{+}\mathcal{F}^{X}\texttt{ return}\right] = \left(1+N_{\text{det}}\right) N_{\text{Fbin}} \times 4\,\text{bytes}\,. \tag{26}$$

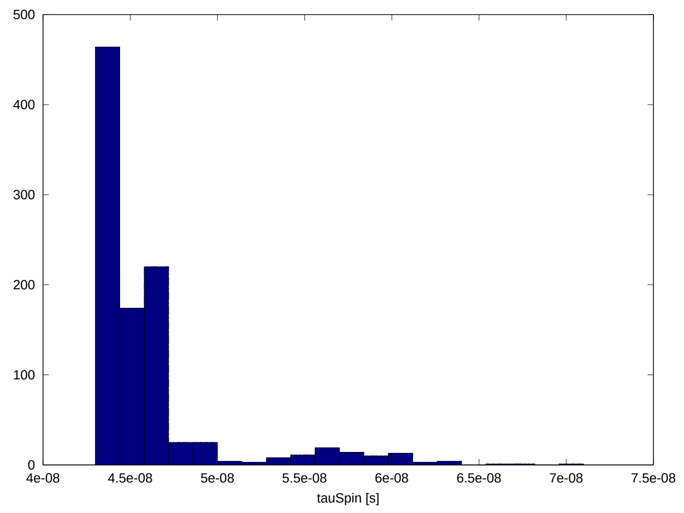
1. *Timing results based on `ComputeFstatBenchmark`[outdated]*

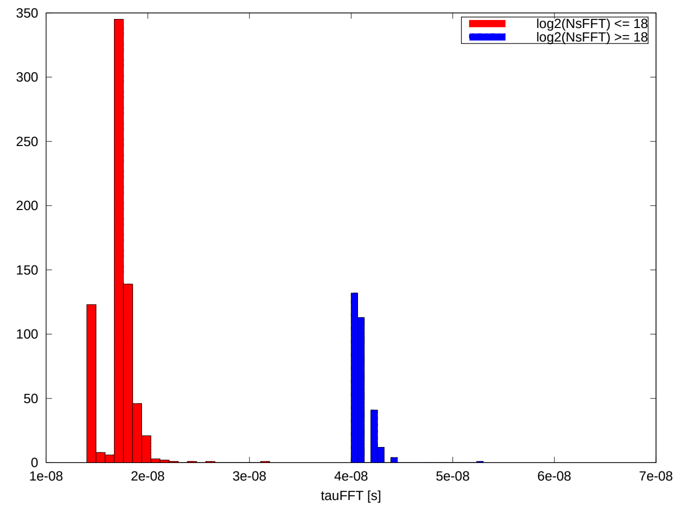
Running 1000 trials of `ComputeFstatBenchmark` on a Thinkpad T520, using commandline options

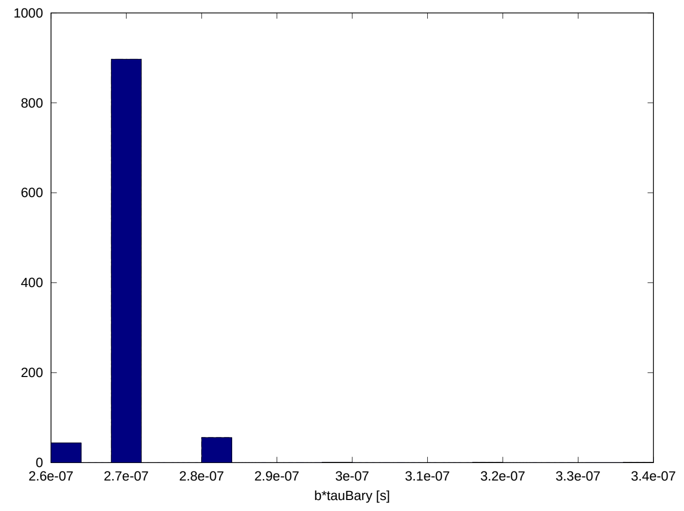
```
$ ./ComputeFstatBenchmark --numSegments=1 --Tseg=216000 --Freq=1500 --f1dot=-1e-7 --numTrials=1000
```

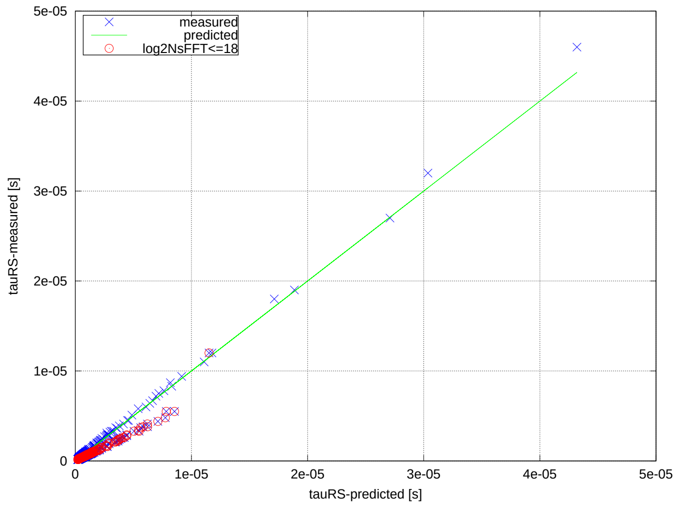
yields the following results (where $b = 1$)

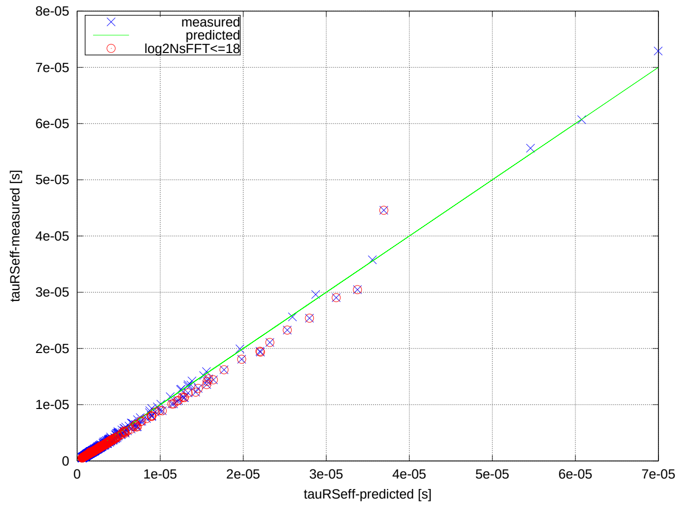








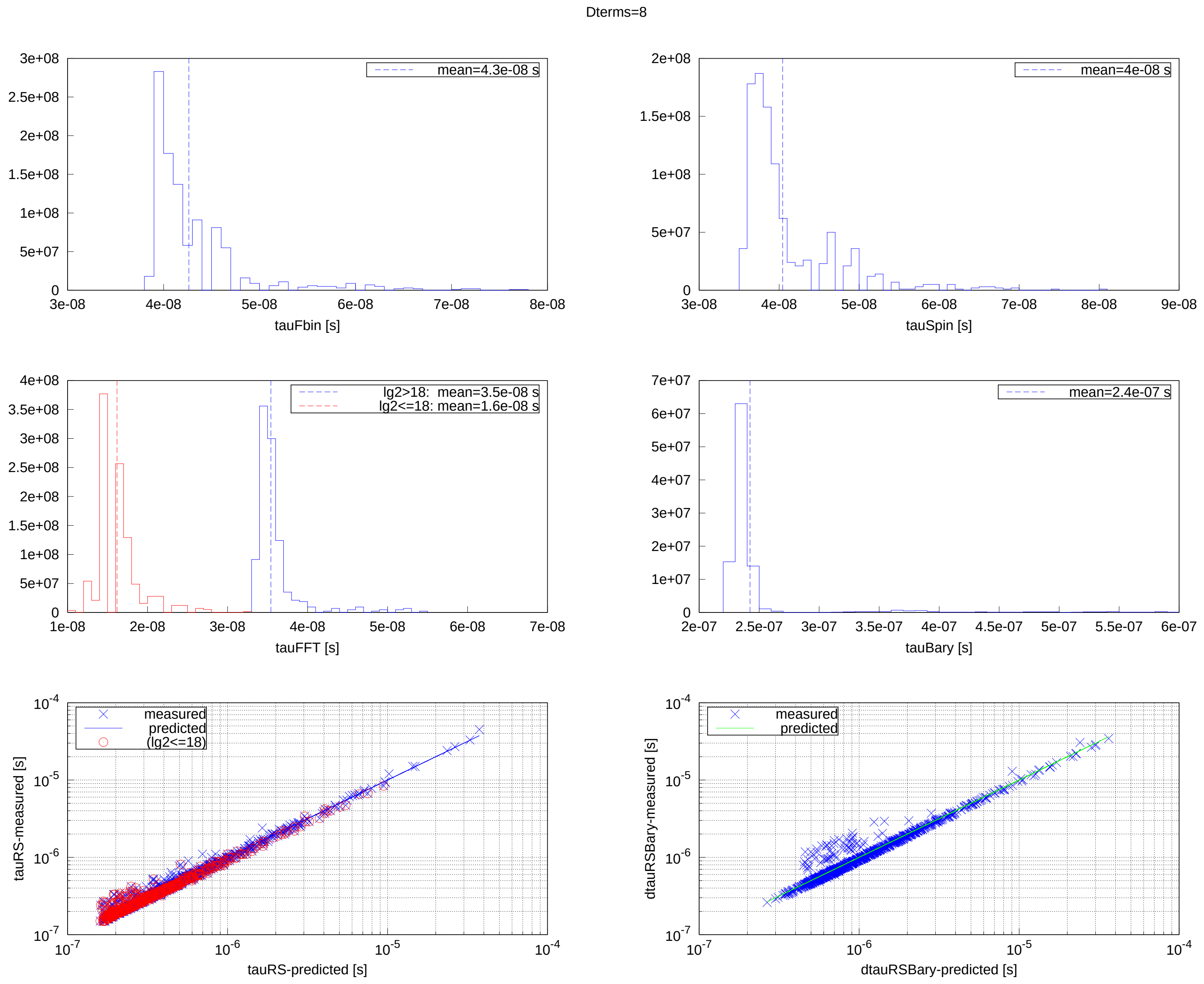




2. Testing new timing model against new (windowed-sinc) Resampling code

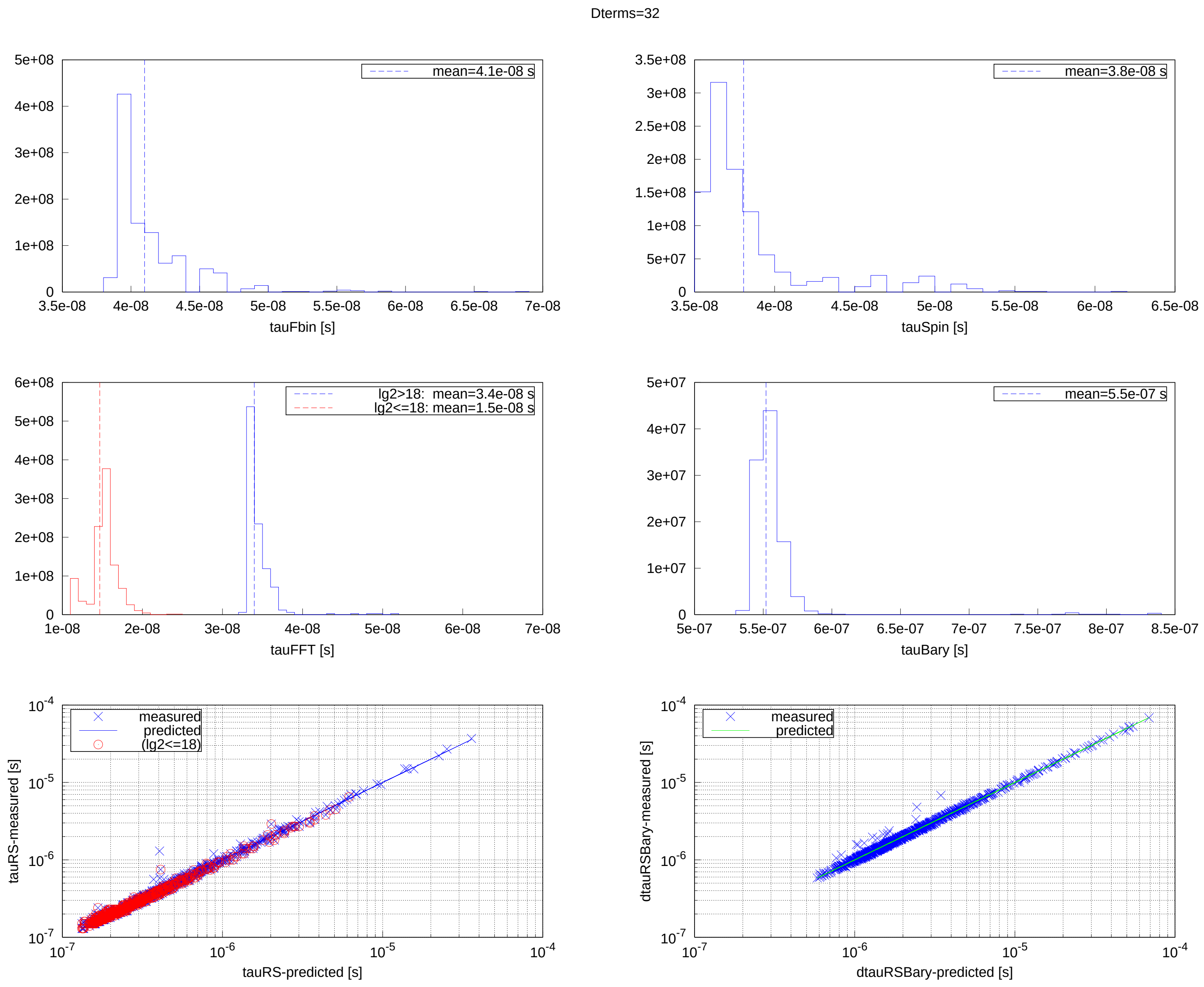
Running 1000 trials of `ComputeFstatBenchmark` using commandline options (`Dterms=8`)

```
$ ./ComputeFstatBenchmark --numSegments=1 --Tseg=216000 --Freq=1499.95 --f1dot=-1e-07 --numTrials=1000
--Dterms=8
```



Running 1000 trials of `ComputeFstatBenchmark` using commandline options (`Dterms=32`)

```
$ ./ComputeFstatBenchmark --numSegments=1 --Tseg=216000 --Freq=1499.95 --f1dot=-1e-07 --numTrials=1000
--Dterms=32
```



3. *Christians GCT timing results (24 runs) [outdated]*

Using Christians 24 timing runs on a fast atlas node, we can compare this model to the obtained timing data. As can be seen in Fig. **??**, for $N_{\mathrm{samp}}^{\mathrm{FFT}} \le 2^{18}$ the FFT seems to be about 30% faster (suspected due to fitting into faster cache memory). However, overall this regime will be less efficient because it's wasting more cycles on $\Delta f_{\mathrm{drift}}$ in relation to Δf) (see Fig. 3). We therefore restrict the trials for τ_{FFT} to those cases where $N_{\mathrm{samp}}^{\mathrm{FFT}} > 2^{18}$, but use all the trials for the other coefficients. Using Eq. (23) this yields the timing coefficients

$$\tau_{\mathrm{Fbin}} \approx 6.1 \times 10^{-8}\,\mathrm{s}\,, \quad \tau_{\mathrm{FFT}} \approx 3.3 \times 10^{-8}\,\mathrm{s}\,, \quad \tau_{\mathrm{spin}} \approx 7.7 \times 10^{-8}\,\mathrm{s}\,. \tag{27}$$

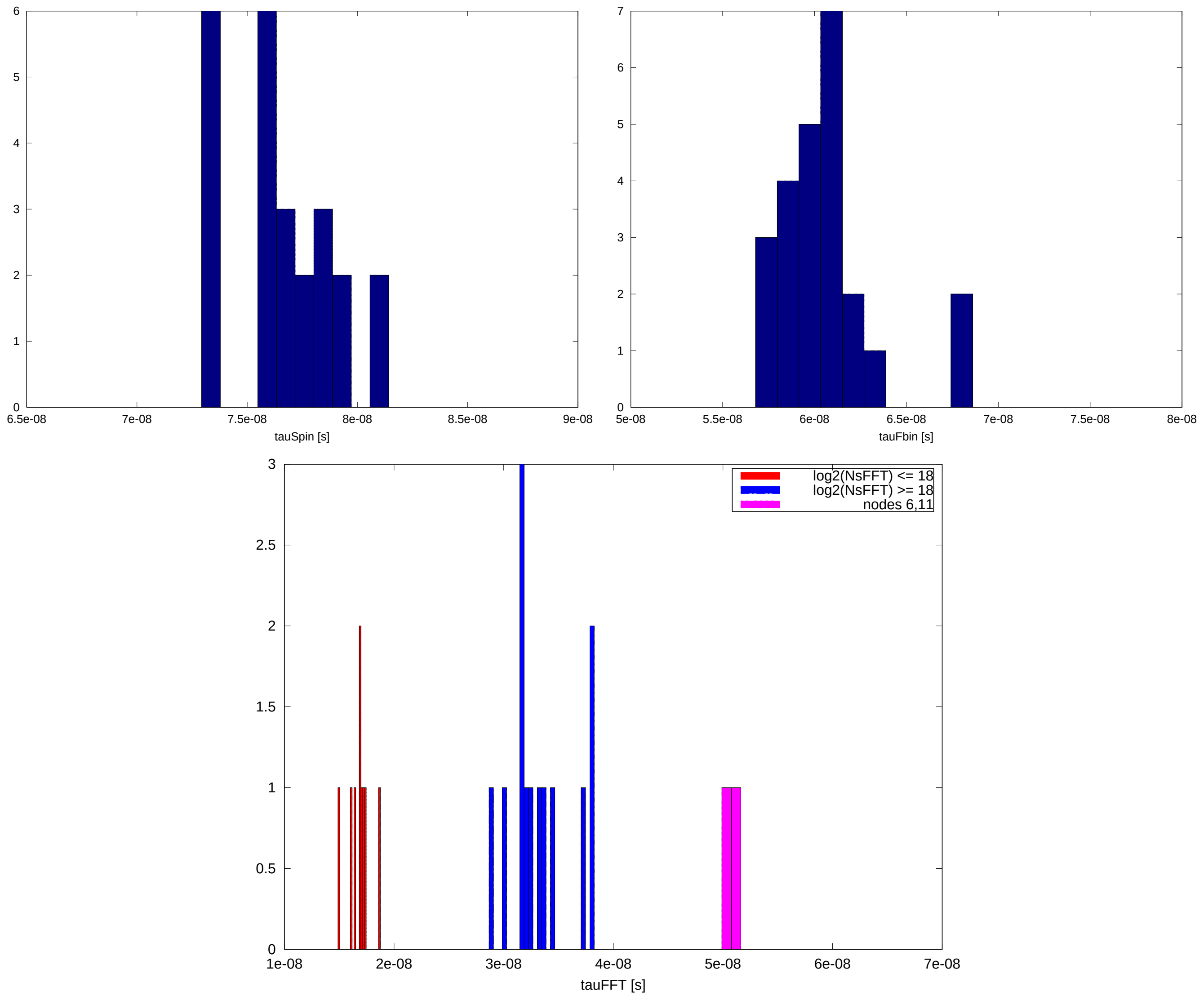


FIG. 1. Histograms over τ_{spin} (left), τ_{Fbin} (right), and τ_{FFT} (bottom).

Finally, let us check if we can accurately predict the code's Resampling parameters N_{Fbin}, $NSampFFT$ etc from the search parameters.

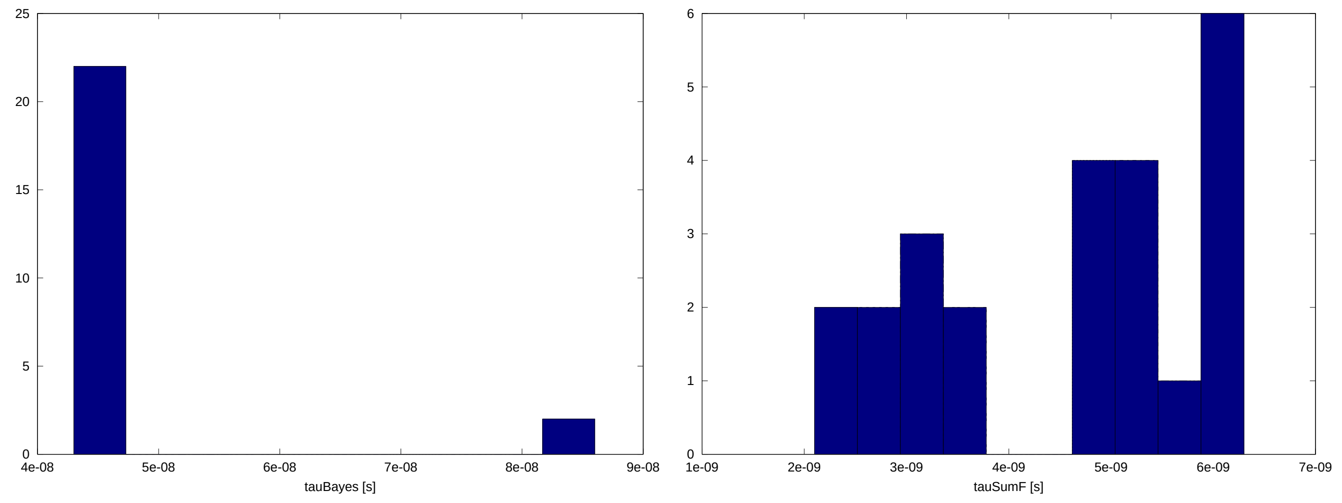


FIG. 2. Histograms over $\tau_{\rm FFT}$ (left), $\tau_{\rm spin}$ (middle) and $\tau_{\rm Fbin}$ (right)

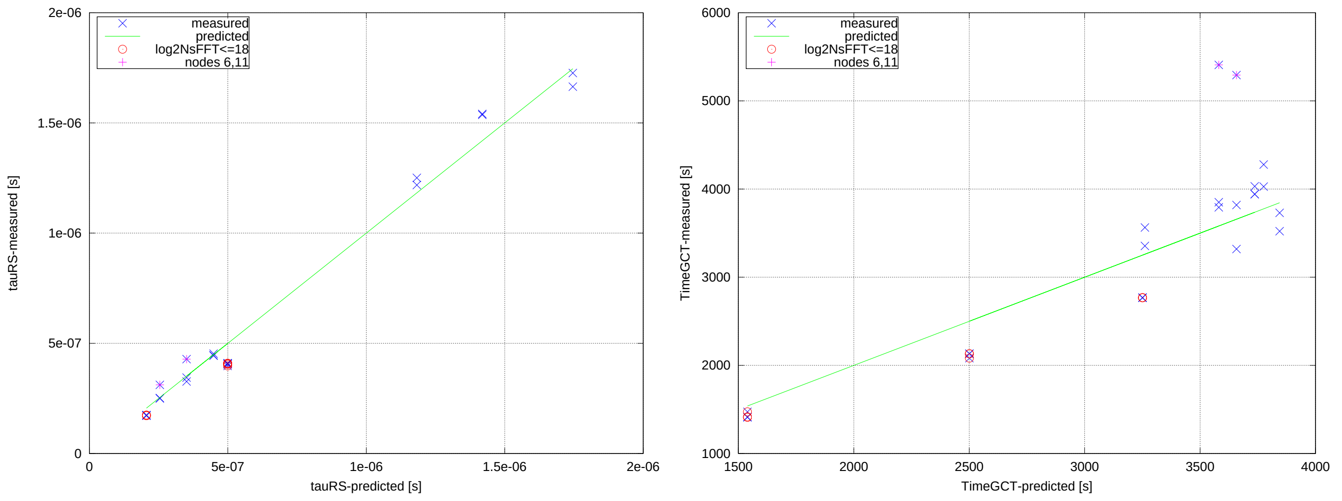


FIG. 3. *Left:* Comparison of measured versus predicted resampling time $\tau_{\rm RS}$ (from Eq. (20)), over all 24 trials. *Right:* Comparsion of measured versus predicted total GCT runtime.

Appendix A: General useful relations for resampling $\mathcal{F}$-statistic

Continuous-time function $x(t)$, Fourier transform is defined as

$$\widetilde{x}(f) \equiv \int_{-\infty}^{\infty} x(t)\, e^{-i2\pi\, f\, t}\, dt\,, \tag{A1}$$

and in the following we will refer to this function as the *spectrum* of $x(t)$. The inverse operation is

$$x(t) = \int_{-\infty}^{\infty} \widetilde{x}(f)\, e^{+i2\pi\, f\, t}\, df\,, \tag{A2}$$

which can be shown using the fact that $\int_{-\infty}^{\infty} e^{i2\pi\, f'(t-t')}\, df = \delta(t-t')$.

1. Sampling theorem for band-limited functions

Sampling theorem: Assuming the continuous function $x(t)$ is *band limited*, which means that the spectrum $\widetilde{x}(f)$ vanishes outside of the support $[-W, W]$, i.e.

$$\widetilde{x}(f) = 0 \quad \text{for} \quad |f| > W\,, \tag{A3}$$

where W is the *bandwidth*, one can show [3] that $x(t)$ is fully determined by a discrete timeseries $\{x_j\}_{j=-\infty}^{\infty}$ given by

$$x_j \equiv x(t_j)\,, \quad \text{where} \quad t_j \equiv j\,\Delta t\,, \tag{A4}$$

with sampling frequency f_s and stepsize Δt given by

$$f_\mathrm{s} = \frac{1}{\Delta t} = 2W\,, \quad \Delta t \equiv \frac{1}{2W}\,. \tag{A5}$$

Proof: In order to see this, let us first consider the *impulse-sampled* function

$$x_\mathrm{p}(t) \equiv \Delta t \sum_{j=-\infty}^{\infty} x_j\, \delta(t - j\Delta t)\,, \tag{A6}$$

with spectrum $\widetilde{x_\mathrm{p}}(f)$ obtain from Eq. (A1) as

$$\widetilde{x_\mathrm{p}}(f) = \Delta t \sum_{j=-\infty}^{\infty} x_j\, e^{-i2\pi\, f\, t_j}\,. \tag{A7}$$

This can be seen as the *Fourier series* representation of the periodic spectrum $\widetilde{x_\mathrm{p}}(f + f_\mathrm{s}) = \widetilde{x_\mathrm{p}}(f)$, with period f_s. Using the inverse transform Eq. (A2) to express x_j, we can further obtain

$$\widetilde{x_\mathrm{p}}(f) = \sum_{n=-\infty}^{\infty} \widetilde{x}(f - n\, f_\mathrm{s})\,, \tag{A8}$$

where we used the identity in Eq. (B3). In other words, the spectrum of the discrete timeseries $\{x_j\}$ sampled from $x(t)$ at sampling rate f_s is periodic with period f_s and can be expressed as an infinite sum of replicas of the original spectrum of $x(t)$ shifted by f_s. If the spectrum $\widetilde{x}(f)$ is band-limited within $[-W,\, W]$ as assumed in Eq. (A3), and $x(t)$ is sampled at a rate $f_\mathrm{s} \geq 2W$, then the shifted spectra do not overlap, and we can recover the original spectrum $\widetilde{x}(f)$ from $\widetilde{x_\mathrm{p}}(f)$, simply via

$$\widetilde{x}(f) = \widetilde{x_\mathrm{p}}(f) \quad \text{for} \quad f \in [-f_\mathrm{s}/2,\, f_\mathrm{s}/2]\,. \tag{A9}$$

Another way to formalize the relation between $\widetilde{x}$ and $\widetilde{x_\mathrm{p}}$ is to write this as

$$\widetilde{x_\mathrm{p}}(f) = \widetilde{x}\left(f'(f)\right)\,, \quad \text{where} \quad f'(f) = f - \mathrm{round}\left(\frac{f}{f_\mathrm{s}}\right) f_\mathrm{s}\,. \tag{A10}$$

We can therefore reconstruct the original continuous function $x(t)$ from Eq. (A2). In other words, the discrete timeseries $\{x_j\}$ fully determines $x(t)$, via the chain

$$\{x_j\} \overset{\text{Eq. (A7)}}{\longrightarrow} \widetilde{x_\mathrm{p}}(f) \overset{\text{Eq. (A9)}}{\longrightarrow} \widetilde{x}(f) \overset{\text{Eq. (A2)}}{\longrightarrow} x(t)\,. \tag{A11}$$

If the shifted spectra $\widetilde{x}(f)$ overlap, i.e. if $x(t)$ is not band-limited within $[-f_\mathrm{s}/2, f_\mathrm{s}/2]$, then we cannot recover the original spectrum $\widetilde{x}(f)$ from $\widetilde{x_\mathrm{p}}(f)$, and we speak of *aliasing.*

The explicit relation Eq. (A11) can be obtained by substituting Eq. (A7) into Eq. (A2) restricted to $\int_{-f_\mathrm{s}/2}^{f_\mathrm{s}/2}$, which yields the well-known sinc-interpolation formula

$$x(t) = \sum_{j=-\infty}^{\infty} x_j\, \frac{\sin \pi\, \delta_j(t)}{\pi\, \delta_j(t)}\,, \quad \text{with} \quad \delta_j(t) \equiv f_\mathrm{s}\, t - j = \frac{t - t_j}{\Delta t}\,. \tag{A12}$$

2. Time-limited functions

Consider the additional constraint that $x(t)$ vanishes outside the interval $t \in [0, T]$, such that only $\{x_j\}_{j=0}^{N-1} \neq 0$, and $T = N\,\Delta t$. Therefore by the sampling theorem, the spectrum of the sampled timeseries Eq. (A7) can now also be sampled without loss of information on N frequency samples spaced by $\Delta f = 1/T$, denoted as $f_k = k\,\Delta f$ for $k = 0, \dots N-1$, resulting in the discrete Fourier transform (DFT)

$$\widetilde{x}_k \equiv \widetilde{x_{\rm p}}(f_k) = \Delta t \sum_{j=0}^{N-1} x_j\, e^{-i2\pi\, jk/N}\,, \tag{A13}$$

where we used the fact that $\Delta t\,\Delta f = 1/N$. The inverse operation is given by

$$x_j = \Delta f \sum_{k=0}^{N-1} \widetilde{x}_k\, e^{i2\pi\, jk/N}\,, \tag{A14}$$

which can be obtained using the identity in Eq. (B4). Note that due to the discrete sampling in the frequency domain, formally the corresponding time-domain series is now also periodic with period T, as from Eq. (A14) we see that $x_{j+N} = x_j$.

The set of $\{\widetilde{x}_k\}_{k=0}^{N-1}$ therefore fully determine the $\{x_j\}_{j=0}^{N-1}$, which fully determine $\widetilde{x_{\rm p}}(f)$ at *any* frequency f via Eq. (A7), namely

$$\widetilde{x_{\rm p}}(f) = \sum_{k=0}^{N-1} \widetilde{x}_k\, D_k(f)\,, \tag{A15}$$

where the "Dirichlet" interpolation kernel is found as

$$D_k(f) \equiv e^{-i\pi(1-1/N)\delta_k(f)} \frac{\sin \pi\delta_k(f)}{N \sin(\pi\delta_k(f)/N)} \tag{A16}$$

$$\overset{N \gg \delta_k(f)}{\approx} \; e^{-i\pi\delta_k(f)} \frac{\sin \pi\delta_k(f)}{\pi\delta_k(f)} \tag{A17}$$

$$= \frac{\sin 2\pi\delta_k(f)}{2\pi\delta_k(f)} - i\,\frac{1 - \cos 2\pi\delta_k(f)}{2\pi\delta_k(f)}\,, \tag{A18}$$

with

$$\delta_k(f) \equiv T\,f - k = \frac{f - f_k}{\Delta f}\,. \tag{A19}$$

This is very analogous to the time-domain sinc-interpolation of Eq. (A12), but also differs in two respects: the 'sinc' form is only obtained approximately in the large-N limit, and there is an additional complex phase factor $e^{-i\pi\delta_k(f)}$ that stems from the fact that the time DFT is defined with respect to the start time $t_j = 0$, while the start-frequency $f_k = 0$ is really the mid-frequency of the original spectrum.

3. Time-shifting DFTs

A question of practical interest: given a DFT $\{\widetilde{x}_k\}_{k=0}^{N-1}$ representing a function $x(t)$, what is the DFT $\{\widetilde{y}_k\}_{k=0}^{N-1}$ of the time-shifted function $y(t) \equiv x(t + \tau)$? It is straightforward to see from Eq. (A1) that $\widetilde{y}(f) = e^{i2\pi f\tau}\,\widetilde{x}(f)$, but we need to be careful when translating this back into the DFT, because $\widetilde{y}_k \equiv \widetilde{y_{\rm p}}(f_k) = \widetilde{y}\,(f'(f_k))$, i.e. we need to use the physical frequencies $f'(f_k)$ to compute the phase shifts, not f_k directly. In other words we obtain

$$\widetilde{y}_k = \begin{cases} e^{i2\pi f_k\,\tau}\,\widetilde{x}_k\,, & \text{for} \quad f_k \leq f_{\rm s}/2\,, \\ e^{i2\pi(f_k - f_{\rm s})\tau}\,\widetilde{x}_k\,, & \text{for} \quad f_k > f_{\rm s}/2\,, \end{cases} \tag{A20}$$

so "negative" frequency bins are multiplied by an extra phase factor of $e^{-i2\pi f_{\rm s}\tau}$. This also ensures that a time-shifted real-valued function remains real-valued.

Appendix B: Useful Fourier identities

From the well-known delta-family expression

$$\sum_{k=-\infty}^{\infty} e^{i2\pi\, kx} = \delta(x)\,, \quad \text{for} \quad x \in [-\frac{1}{2}, \frac{1}{2}]\,, \tag{B1}$$

combined with the fact that the lhs is periodic with period $x \to x+1$, we can write

$$\sum_{k=-\infty}^{\infty} e^{i2\pi\, kx} = \sum_{n=-\infty}^{\infty} \delta(x-n)\,, \tag{B2}$$

and writing $x = t/P$, with the fact that $\delta(ax) = \delta(x)/|a|$, we obtain

$$\frac{1}{P}\sum_{k=-\infty}^{\infty} e^{i2\pi\, \frac{k}{P} t} = \sum_{n=-\infty}^{\infty} \delta(t-nP)\,. \tag{B3}$$

The finite DFT analalogue of this is the identity

$$\frac{1}{N}\sum_{k=0}^{N-1} e^{i2\pi\, jk/N} = \sum_{n=-\infty}^{\infty} \delta_{j,nN}\,, \tag{B4}$$

which is easy to see by using the expression for the sum of a geometric series.

[1] P. Patel, X. Siemens, R. Dupuis, and J. Betzwieser, Phys. Rev. D. **81**, 084032 (2010), 0912.4255.
[2] P. Jaranowski, A. Królak, and B. F. Schutz, Phys. Rev. D. **58**, 063001 (1998).
[3] C. E. Shannon, Proc. of the IRE **37**, 10 (1949).