The Astrophysical Journal Supplement Series, 265:13 (12pp), 2023 March https://doi.org/10.3847/1538-4365/acb351

Bits Missing: Finding Exotic Pulsars Using bfloat16 on NVIDIA GPUs

Jack White[1], Karel Adámek[1,2], Jayanta Roy[3], Sofia Dimoudi[4], Scott M. Ransom[5], and Wesley Armour[1]
[1] Oxford e-Research Centre, Department of Engineering Science, University of Oxford, 7 Keble Road, Oxford OX1 3QG, UK; wes.armour@oerc.ox.ac.uk
[2] Department of Physics, Silesian University in Opava, Bezručovo nám., 746 01, Opava, Czech Republic
[3] National Centre for Radio Astrophysics (NCRA-TIFR), Pune 411 007, India
[4] Centre for Advanced Instrumentation, Durham University, UK
[5] National Radio Astronomy Observatory, Charlottesville, VA 22903, USA
Received 2022 July 12; revised 2023 January 13; accepted 2023 January 14; published 2023 February 23

Abstract

The Fourier domain acceleration search (FDAS) is an effective technique for detecting faint binary pulsars in large radio astronomy data sets. This paper quantifies the sensitivity impact of reducing numerical precision in the graphics processing unit (GPU)-accelerated FDAS pipeline of the AstroAccelerate (AA) software package. The prior implementation used IEEE-754 single-precision in the entire binary pulsar detection pipeline, spending a large fraction of the runtime computing GPU-accelerated fast Fourier transforms. AA has been modified to use bfloat16 (and IEEE-754 double-precision to provide a "gold standard" comparison) within the Fourier domain convolution section of the FDAS routine. Approximately 20,000 synthetic pulsar filterbank files representing binary pulsars were generated using SIGPROC with a range of physical parameters. They have been processed using bfloat16, single-precision, and double-precision convolutions. All bfloat16 peaks are within 3% of the predicted signal-to-noise ratio of their corresponding single-precision peaks. Of 14,971 "bright" single-precision fundamental peaks above a power of 44.982 (our experimentally measured highest noise value), 14,602 (97.53%) have a peak in the same acceleration and frequency bin in the bfloat16 output plane, while in the remaining 369 the nearest peak is located in the adjacent acceleration bin. There is no bin drift measured between the single- and double-precision results. The bfloat16 version of FDAS achieves a speedup of approximately 1.6× compared to single-precision. A comparison between AA and the PRESTO software package is presented using observations collected with the GMRT of PSR J1544+4937, a 2.16 ms black widow pulsar in a 2.8 hr compact orbit.

Unified Astronomy Thesaurus concepts: Computational astronomy (293); Astronomical methods (1043); Astronomy data analysis (1858); Astronomy software (1855); Computational methods (1965); GPU computing (1969); Binary pulsars (153)

1. Introduction

Binary pulsar systems provide the means to test general relativity via observations of their emissions in the radio spectrum. Described by Taylor & Weisberg (1982), these systems include at least one pulsar in orbit with another compact object. As the pulsar accelerates toward and away from the observer, the Doppler shift modulates the otherwise periodic pulsar signal (in its own reference frame) into a pulse train with varying period in the frame of the observer, the nature of which depends on the shape of the orbit. In the case of a simple circular orbit, this acceleration would be sinusoidal. In practice, the eccentricity and orientation of the orbital plane will cause the acceleration to take on more complicated functional forms. In Section 2, we will explore the implications of this phenomenon in more detail.

This effect reduces the sensitivity of the Fourier transform to the pulse trains emitted by these objects, as it spreads the power of the signal across many frequencies, rendering the signals indistinguishable from the background noise.

Due to the accurate clocklike property of the pulsar embedded in the binary system, binary pulsars (particularly millisecond pulsars) are of interest to astronomers globally. They are good references to trace the evolution of the binary system (Bhattacharya & van den Heuvel 1991) and they are useful for testing the theory of gravity (Kramer et al. 2021). Thus, significant efforts are being made in searching for such accelerated binary systems using time-domain radio surveys. Some recent examples of relevant research include Ridolfi et al. (2021), Swihart et al. (2021), Kansabanik et al. (2021), Morello et al. (2022), and Rajwade et al. (2020).

When looking to next-generation radio telescopes, time-domain data rates will be of the order of terabytes per second. For example, SKA is expected to produce 800 GB s^{-1} (Levin et al. 2017). Due to the volume of data, it is not feasible to store, and so analysis must be performed in real time as data are captured. This motivates the investigation of using reduced-precision computations in computationally demanding parts of the pulsar detection pipeline.

Within such detection pipelines, search techniques for binary pulsar systems that aim to capture all of the power emitted by the pulsar and collect it into a single detection are typically the most computationally demanding.

The search techniques of interest in this paper operate on time-series data sets. To generate this one-dimensional signal from a filterbank, the effects of dispersion must be removed (dedispersion) at a range of trial dispersion measures (DMs). This has the impact of increasing the number of time series that each search technique must operate on by many orders of magnitude. For the forthcoming SKA radio telescope, this can be a factor as high as 6000 times (Levin et al. 2017). The

details of the dedispersion algorithm are described in Armour et al. (2012).

Two different methods are favored for searching for binary systems. The first is via time-domain methods (Middleditch & Kristian 1984), where the observed time series from a radio telescope is repeatedly Doppler shifted and Fourier transformed in an attempt to move the observed data into a frame whereby the emitter appears to the observer to be stationary.

The second method relies on convolving the frequency spectrum of the observed time series with templates that represent the frequency domain signatures of pulsars with given accelerations. The computation of this approach can be accelerated via the convolution theorem and GPU-accelerated fast Fourier transforms (FFTs).

Fourier domain acceleration search (FDAS) is an implementation of the latter approach, a search technique for detecting accelerated binary pulsar signals in time-series data sets, generated by radio telescopes. It was proposed in Ransom et al. (2002) and then extended by Andersen & Ransom (2018) to account for nonlinear acceleration, known as jerk searching. AstroAccelerate (AA) implements GPU-accelerated FDAS and jerk searching (Dimoudi & Armour 2017; Dimoudi et al. 2018; Adámek & Armour 2019; Adámek et al. 2020a, 2020b). In AA, the majority of the numerical computation in the binary pulsar detection pipeline is performed during FDAS.

The goal of this work is to establish whether switching the bulk of the numerical processing in the binary pulsar detection pipeline of AA to reduced (bfloat16) precision leads to acceptable sensitivity in the output, compared to the existing single-precision version. The motivation for this is to speed up the pipeline to allow the widest possible parameter search space in real time on next-generation radio telescopes, or to reduce the hardware requirements needed to perform such searches.

When performing FDAS, AA spends a significant fraction of the runtime performing GPU-accelerated FFTs. The performance of a GPU-accelerated FFT is limited by memory bandwidth on current-generation hardware (Adámek 2021), and so by reducing the numerical precision of the calculations from 32 to 16 bits, this bottleneck can be alleviated simply by halving the amount of data that need to be transported to and from the GPU processing cores.

In this paper, we present the results of testing a bfloat16 implementation of FDAS in AA with synthetic pulsar files using SIGPROC's (Lorimer 2011) `fake` filterbank generator, spanning a wide range of parameters, and finally a confirmation of our findings using observations from the GMRT.

2. Searching for Binary Pulsars

2.1. Detailed Explanation

FDAS uses convolution to compare the Fourier transformed time-domain signal with templates representing a range of acceleration values. A peak in the response will be seen when the signal and template overlap, indicating the presence of a pulsar with similar parameters to the template. The number of templates used depends on the search parameters desired by the radio astronomer. During pulsar searches, the radio astronomer is primarily interested in pulsars that have not already been discovered. Therefore, the range of search parameters must be wide enough to also include "exotic" pulsars. This might mean, for example, pulsars with an extremely fast pulse period, combined with extreme acceleration—the latter suggesting a very tight orbit around a compact object. In such cases, more pulsar templates are required to span the extreme values of acceleration. The computing and memory requirements of FDAS increase linearly with the number of templates.

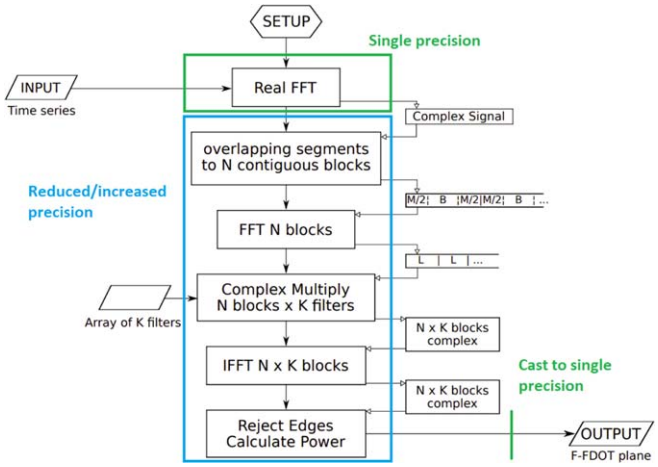

Figure 1. Flowchart depicting the process of Fourier domain matched filtering using the overlap–save method on a GPU. Courtesy of Dimoudi et al. (2018).

These templates will represent varying values of Doppler shift (which is a function of the particular orbital dynamics of the system). The span of the Doppler shift values defines the breadth of a binary pulsar search, along the axis known as frequency derivative, or $\dot{f}$, which is related to the acceleration of the pulsar and the Fourier bin drift, z, as

$$\text{Acceleration} = \frac{\dot{f}\,c}{f} = \frac{zc}{fT^2}. \tag{1}$$

Where f is the frequency of the pulsar and T is the observation time. Equation (4) in Freire et al. (2001) demonstrates how to predict the acceleration of a pulsar given a set of orbital parameters.

The signal is convolved with all templates resulting in the f–$\dot{f}$ plane. Frequency f and frequency derivative $\dot{f}$ therefore make up the primary axes of the pulsar search; a pulsar with a given pulse frequency and acceleration value will appear as a peak on this plane, with diminishing harmonics at integer multiples of both f and $\dot{f}$. The summed height (power) of the peak and harmonics can be used to calculate the signal-to-noise ratio (S/N) value of the detected pulsar against the statistics of the background noise.

The convolution of a signal and a set of templates could be completed entirely in the time domain, but a time-domain convolution involves $O(N^2)$ multiplications and $O(N^2)$ additions, leading to overall $O(N^2)$ complexity. Using the FFT and convolution theorem, this can be reduced to an overall complexity $O(N \log N)$ in the Fourier domain. The overlap–save method is used to perform the convolutions on the GPU, as described in Dimoudi et al. (2018), Adámek et al. (2020c) and Figure 1. Briefly, this involves splitting the long input signal into windows that are separately convolved with the templates, and then the central region of each windowed convolution is saved, discarding the contaminated regions at the edge of the windows.

2.2. FDAS versus Jerk Search

As mentioned in Section 1, two related methods are available when searching for binary systems in the Fourier domain; these are the FDAS and jerk search. FDAS makes the assumption that the acceleration of the emitting object over the observation time is approximately constant. The observation time should be no longer than approximately 10% of the orbital period for this assumption to hold.

Jerk searching takes into account a change in acceleration, which relaxes the observation time constraint, allowing a wider range of pulsars to be detected. Therefore, the jerk search extends the f–$\dot{f}$ plane into an f–$\dot{f}$–$\ddot{f}$ volume. More candidate templates are required to populate the third dimension, $\ddot{f}$, for each $\dot{f}$ value that is evaluated. Because jerk searching is based on the same principle of convolving templates, for this paper we have chosen to focus on the less computationally extensive FDAS, to allow us to profile the numerical sensitivity of AA across a wider range of $\dot{f}$ than we would be able to in a given time with jerk searching.

3. Updating the Pipeline to bfloat16

The existing version of AA used single precision in all stages of FDAS (see Figure 1). For this work, we aim to profile the numerical sensitivity of the output of AA to changing precision in FDAS, with a focus on reducing precision to bfloat16 and comparison with IEEE-754 double precision.

Refactoring the code to mixed-precision is a straightforward process, involving replacement of the existing single-precision functions with their bfloat16 equivalents.

The "Real FFT" in Figure 1 was intentionally left in single precision, as it does not significantly affect the overall execution time (it does not scale with K, the number of filters). Reducing the precision of this FFT disproportionately affects the quality of the output for two reasons. First, at this stage of the pipeline, the data may not span a large dynamic range, and they may not be centered on zero, which highlights the weakness of bfloat16. Compared to single precision, at large scales the spacings between bfloat16 numbers are extremely coarse, so small perturbations on the input data may not be representable without the use of single precision. Second, this FFT (with a length equal to the entire input data) is sufficiently long for the numerical error, which accumulates with the larger number of calculations, to negatively impact sensitivity. Section 4.1 of Murillo et al. (2022) compares the numerical sensitivity of bfloat16 and single precision in the calculation of FFT spectra. Mishra et al. (2022) compares the throughput of bfloat16 logic units with single-precision equivalents.

In standalone testing of cuFFT, we found that bfloat16 FFTs execute 1.7× faster than the single-precision implementation, once the data were transferred to the GPU. The data transfer speeds up by 1.9× when using bfloat16 versus single precision. As we shall present in Section 5.4, we saw this lead to an overall speedup of 1.6× in AA. An input length of 1024 was used because this is the window size employed in the overlap–save implementation described in Section 2.

3.1. Casting between Single Precision and bfloat16

As can be seen in Figure 2, in binary form a bfloat16 number can be represented in single precision simply by appending 16× zero bits to the end. Therefore, single precision and bfloat16 have approximately the same valid range

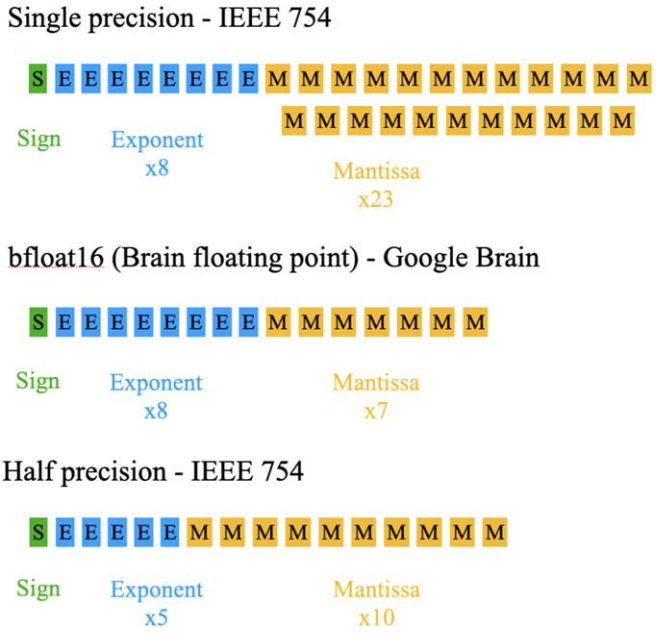

Figure 2. Comparison of bit layouts of IEEE-754 single- and half precision with bfloat16.

($\approx \pm 3 \times 10^{38}$), but bfloat16 numbers are far coarser within this range.

To minimize time spent converting data once the output f–$\dot{f}$ plane had been produced, in the bfloat16 version of the code the final cast to single precision (see Figure 1) is completed by copying the two bytes of data for each bfloat16 number into the corresponding positions of an all-zero single-precision output array.

This quick casting is only possible due to the perfect bit alignment between bfloat16 and single precision, it is not possible with half precision and single precision. Quick casting all but eliminates the performance impact of implicitly casting the data using a standalone GPU kernel.

3.2. IEEE-754 Half versus bfloat16 Precision

The valid range of any data format type is a consideration when comparing floating point formats. Half precision (IEEE-754) has a valid range of ±65504. This introduces some considerations for hypothetically using half precision in this application.

When calculating the Fourier spectrum of a time series, the zeroth bin is the sum of all elements of the time series, often referred to as the "DC component" of the signal. Unless the signal is perfectly centered on zero, this may mean that, in long FFT calculations, the zeroth bin accumulates power that takes its value outside of the ±65,504 bounds imposed by IEEE-754 half-precision. Depending on how the hardware handles numerical overflows, in the worst case this could introduce an `Inf` or `NaN`, while in the best case, any information outside this range would be lost. If an `Inf` or `NaN` is present in a spectrum that is subsequently inverse Fourier transformed, all values of the output will be polluted and subsequently unusable.

Given the risk of introducing overflow errors to the pipeline, we selected bfloat16 as our candidate data format for reducing the precision, which also included the benefit of quick casting.

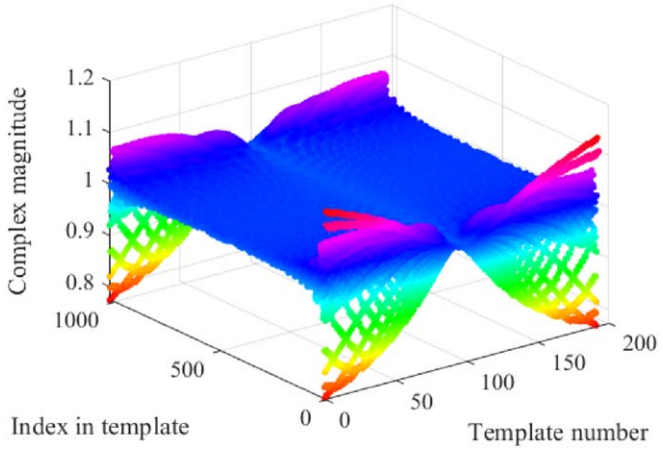

Figure 3. Single-precision acceleration templates—complex magnitude plotted.

3.3. Template Conversion

To convert the convolution step of FDAS to a lower precision, it is first necessary to convert the convolution templates to lower precision. These can be thought of as the candidate pulsar signatures that the code will look for in the data sets. Each template is a 1D array of complex numbers, and the single-precision complex magnitudes are plotted in Figure 3. In all experiments, the convolution templates are first generated in single precision and then explicitly/implicitly converted to bfloat16/double precision by a tuned GPU kernel before being transformed into the frequency domain.

For the real component and the imaginary component, we calculate the relative difference between the single-precision and bfloat16-precision value of the template as

$$\% = \frac{2 \times (b - s)}{b + s} \times 100, \tag{2}$$

where b is the bfloat16 value and s is the single-precision value. The result of applying this equation to the real and imaginary part of all templates is plotted in Figures 4 and 5.

This metric will be reused in multiple experiments, as it is important to have a comparison of bfloat16's ability to recreate higher-precision results that is invariant to the size of the numbers being dealt with.

The aggregated statistics of the relative differences are shown in Table 1, it is clear that at this early stage in the pipeline, only a very small amount of numerical error has been introduced by reducing the precision of the generation of the acceleration templates from single precision to bfloat16.

4. Methodology

4.1. Generating Synthetic Data

SIGPROC (Lorimer 2011) was used to generate synthetic pulsar data of binary pulsars in circular orbits. The input parameters to SIGPROC's `fake` determine the physical properties of the binary system being simulated.

To test and compare the varying precisions of AA across a representative range of potential pulsars, the `fake` input parameters are sampled from a log-uniform distribution with upper and lower bounds (Table 2), so as to get an even representation of potential numerical scales, without introducing bias toward any particular range. The bounds are picked to represent a wide range of potentially physically viable pulsars, although some combinations of parameters are at the extrema of our ranges and are included to investigate the limits of the numerical processing.

Sampling a single point in this space is computationally costly; synthetic `fake` file generation followed by running the generated data through AA takes roughly 2 minutes. Therefore, it is desirable to reduce the dimensionality of the input parameter space wherever it is physically justifiable, to reduce the "curse of dimensionality."

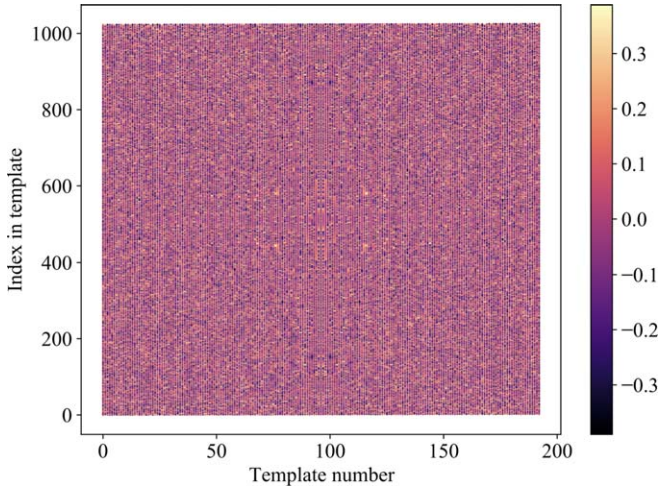

Figure 4. Relative difference between real components of acceleration templates; color scale represents relative difference as in Equation (2).

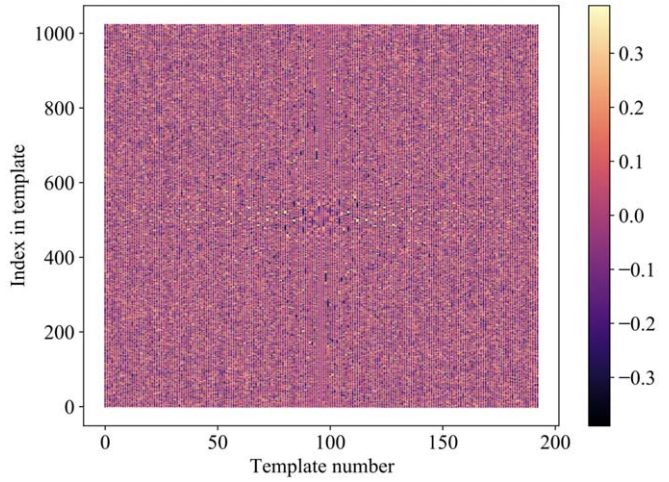

Figure 5. Relative difference between imaginary components of acceleration templates; color scale represents relative difference as in Equation (2).

Table 1
Statistics of the Difference between Single Precision and bfloat16 after Template Conversion

Component	Mean	St. Dev	Max	Min
Imaginary	−0.001569%	0.1537%	0.3883%	−0.3894%
Real	−0.001093%	0.1532%	0.3878%	−0.3898%

Note. Aggregated from data used to generate Figures 4 and 5.

Table 2
Configuration Arguments for SIGPROC `fake` Command to Generate a Binary Pulsar Signal

Parameter	Range	Units
Spin period	[1.25, 1000]	ms
Pulse width	[4, 50]	%
Input signal S/N (`snrpeak`)	[0.0125, 0.125]	
Dispersion measure	[5.. 500]	cm^{-3} pc
Bits per sample	8	
Number of channels	1024	
Sampling time	128	μs
Observation time	600	s
First channel frequency	1550	MHz
Channel offset	0.292968752	MHz
Additional flags	-binary	
Orbital period	[1.5, 336]	hr
Starting orbital phase	0.2	
Pulsar mass	[1.0, 1.5]	$M_\odot$
Companion mass	[0.1, 5.0]	$M_\odot$

Note. [a, b] represents the closed interval inclusive of the bounds, while [a.. b] represents the integer interval inclusive of the bounds.

4.2. Selecting an Optimal Bphase Value

The `bphase` parameter to `fake` represents the starting orbital phase of the binary system, and it can take values between 0 and 1. To reduce the dimensionality of the parameter space, it is necessary to determine roughly which value of `bphase` leads to the largest acceleration value of the detected peaks in the output space, to ensure each test is of a pulsar signature with at least some acceleration. Conceptually, the largest observed acceleration is when the emitter is traveling directly toward or directly away from the observer.

The simple experiment chosen is to hold all parameters fixed at typical values, and allow `bphase` to sweep over its entire range. Then, by plotting the output frequency–acceleration plane, the acceleration value of the highest peak can be captured and plotted (Figure 6).

Through the results of this sweep, the static value selected for `bphase` is 0.2, this approximately centers the observation window over the period of orbit with maximum linear acceleration toward the observer, as shown in Figure 6, although it is worth noting that, at this value, simulations will be affected by significant levels of jerk. This is an approximation, as the `bper` value (which sets the orbital period in hours) is a variable input parameter to `fake`, and it is the combination of `bphase` and `bper` that sets the midpoint of the observation window.

4.3. Peakfinding Methodology

An accelerated pulsar signal can be observed in the f–$\dot{f}$ plane as a series of peaks at integer multiples of f and $\dot{f}$. Quantization error in both the frequency and acceleration axes of the search lead to the observed peaks drifting into adjacent bins. To account for this, a harmonic sum algorithm is used to gather the power from all harmonics into a single detection. However, in this work we are interested in the location of each harmonic peak individually, so a harmonic sum is not used.

A more general and naive peakfinding and harmonic separation approach is taken to enable us to look at each harmonic separately. Given that the pulse period of the synthetic pulsar is known a priori, being an input parameter to `fake`, it is possible to pre-emptively split the f–$\dot{f}$ output plane with boundaries (Table 3) derived from the fundamental frequency f, which is calculated as $f = \frac{1}{\text{spin period}}$ Hz.

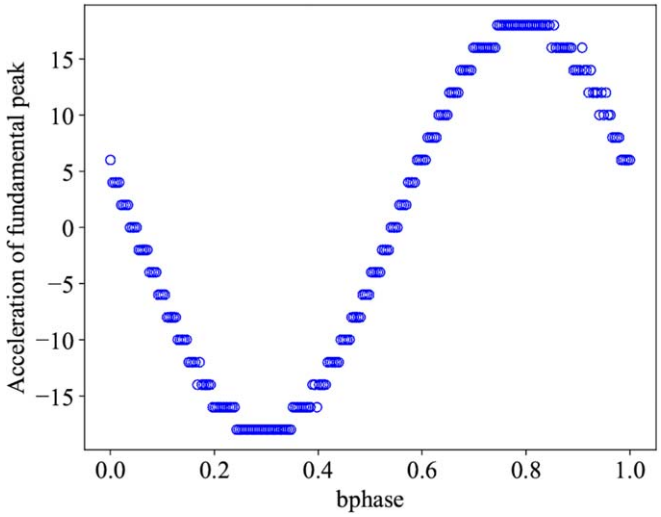

Figure 6. Fundamental peak acceleration ("z" bin) against `bphase`.

Table 3
Harmonic Band Search Bounds

Band	Range (Hz)
Fundamental	$[0, 1.5 \times f]$
First harmonic	$[1.5 \times f, 2.5 \times f]$
Second harmonic	$[2.5 \times f, 3.5 \times f]$
Nth harmonic	$[(N + 0.5) \times f, (N + 1.5) \times f]$

Note. [a, b] represents the closed interval inclusive of the upper bound, and lower in the case of the fundamental.

Once the f–$\dot{f}$ plane has been split into bands based on the calculated fundamental frequency, the nth harmonic is taken to be the highest point in the nth band. In the case of low-S/N pulsar signals, peaks and their harmonics are likely to be at similar powers to the background noise. This leads to a failure mode where the peak selection method picks a random point of noise as the harmonic in that band. As the noise is randomly distributed, the highest point in the bfloat16 plane and the single-precision plane are not guaranteed to be in the same position, leading to some extreme observed values of bin drift. This failure mode is discussed and quantified in Section 5.3.

In the majority of cases where the peak height is not at the noise floor, measuring the "bin drift" of a peak in a given band between precisions allows us to quantify the combined extent of the spectral leakage, accumulated numerical error, and scalloping loss, and its effect on our ability to resolve a given pulsar from a certain noisy background. Chapter 2 of Ransom et al. (2002) contains an in-depth discussion of sources of signal loss in FDAS, such as scalloping loss, which can cause a loss of up to 60% in amplitude.

The following section compares the ability of our FDAS implementation (in various precisions) to detect the presence of accelerated pulsars in synthetic data sets.

5. Results

In this section, we present the results from analyzing output f–$\dot{f}$ planes with randomly generated inputs, to understand the

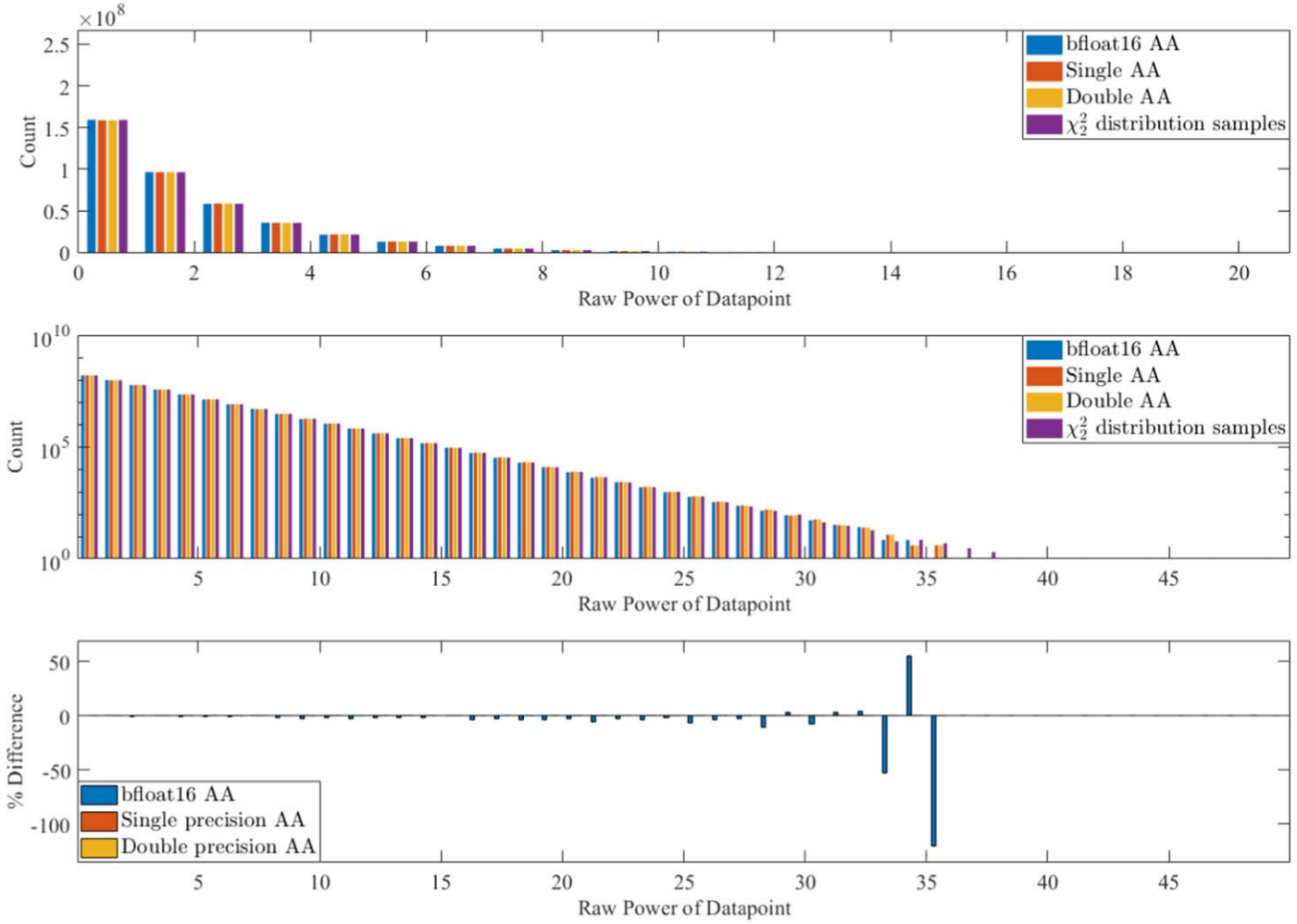

Figure 7. Raw noise power histogram with χ^2_2 (=χ^2 with 2 d.o.f) comparison, linear scale (top). Raw noise power histogram with χ^2_2 comparison, log scale (middle). Percent difference between bfloat16 vs. single-precision bin count and double-precision vs. single-precision bin count (bottom). In all data series, $N = 404{,}750{,}143$ samples. Raw power (complex magnitude squared) is the internal measure of detection strength in AA (or PRESTO) before being converted to a significance level.

conditions under which bfloat16 is a good (or bad) approximation to the higher-precision implementations.

5.1. Noise Properties in Varying Precisions

To understand if a reduction in precision has any statistically significant effect on the background noise distribution, synthetic data without a pulsar signal are analyzed in this section.

The histograms in Figure 7 (top) and (middle) represent the distribution of output power across an entire f–$\dot{f}$ plane for a filterbank file generated using a SIGPROC `snrpeak` (Table 2) parameter of 0. These can be thought of as comparisons that plot the probability density functions (PDFs) that the noise distribution has been drawn from. The `snrpeak` parameter can be thought of as proportional to the brightness of the signal. When `snrpeak` is 0, there is no signal in the file being generated. Between precisions, there is no significant skewing, and there are only very slight perturbations on the count in each bin (compared to the value in single precision), which can be observed upon careful examination of Figure 7 (bottom).

It is interesting to note that, however slight the perturbations are on the count in each bin, the overall trend is that there are marginally fewer large values in the bfloat16 noise than the single-precision noise.

We conclude that reducing or increasing precision has no significant effect on the PDF of the noise.

Figure 7 shows that the noise distribution follows a χ^2_2 distribution; this is expected, as the input data without signal are well-approximated by white noise. We expect the output to follow a χ^2_2 distribution because, in this case, without any signal present, the input is complex Gaussian noise, and the resulting distribution of power is the sum of the squares of each component.

5.2. Synthetic Data Set Results—Power

In this section, we discuss the effect of changing precision on the ability of AA's implementation of FDAS to resolve the power of a particular peak, where a pulsar signal is known to be present in the data set.

Figure 8 shows the relative difference in detection power between candidate peaks in bfloat16 and single precision. To calculate this, we have taken the power of the highest peak in a given harmonic band in bfloat16, as well as the highest peak in the same harmonic band in the single-precision output plane, and compared their power via Equation (2). This process is repeated across many samples spanning the input parameter space (Table 2), to quantify the limitations of the reduced-precision mode.

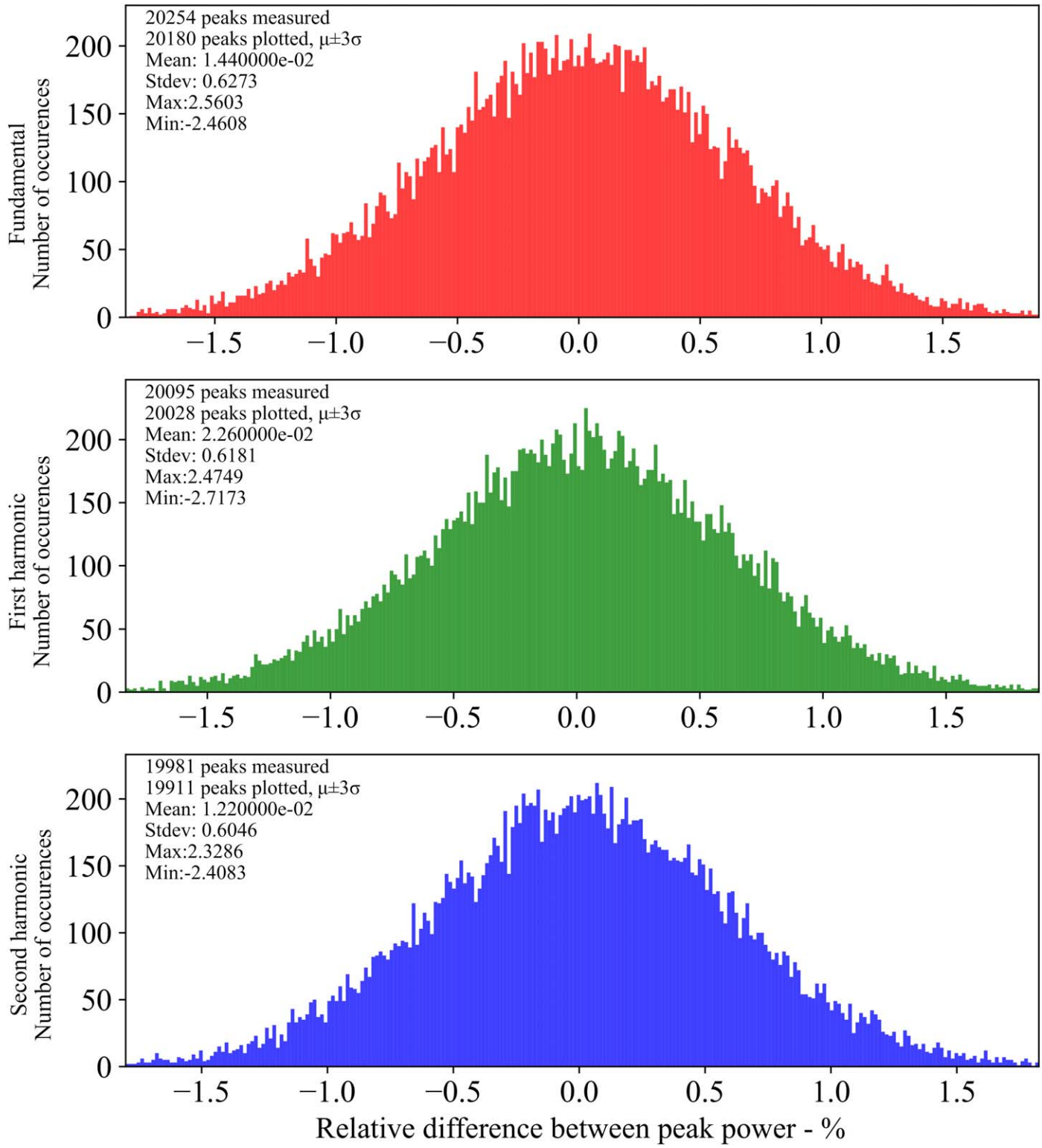

Figure 8. Graphs depicting the spread in peak power recreation between bfloat16 vs. single precision.

The first three harmonics are analyzed separately in each row of Figure 8. There are three different precisions under investigation (bfloat16, single precision, and double precision). The comparison in Figure 8 is made between bfloat16 and single precision. The analysis of the equivalent single-precision versus double-precision data showed that, for 98.8% of the peaks, there was no difference in height, and the maximum observed difference was 0.0098%. This is as we would expect, because rounding a result to single precision eliminates the majority of the benefit of a preceding double-precision calculation, unless extreme amounts of error were accumulating in single precision—for example, if the algorithm were numerically unstable. For comparison, all the bfloat16 peaks were of different heights than their single-precision

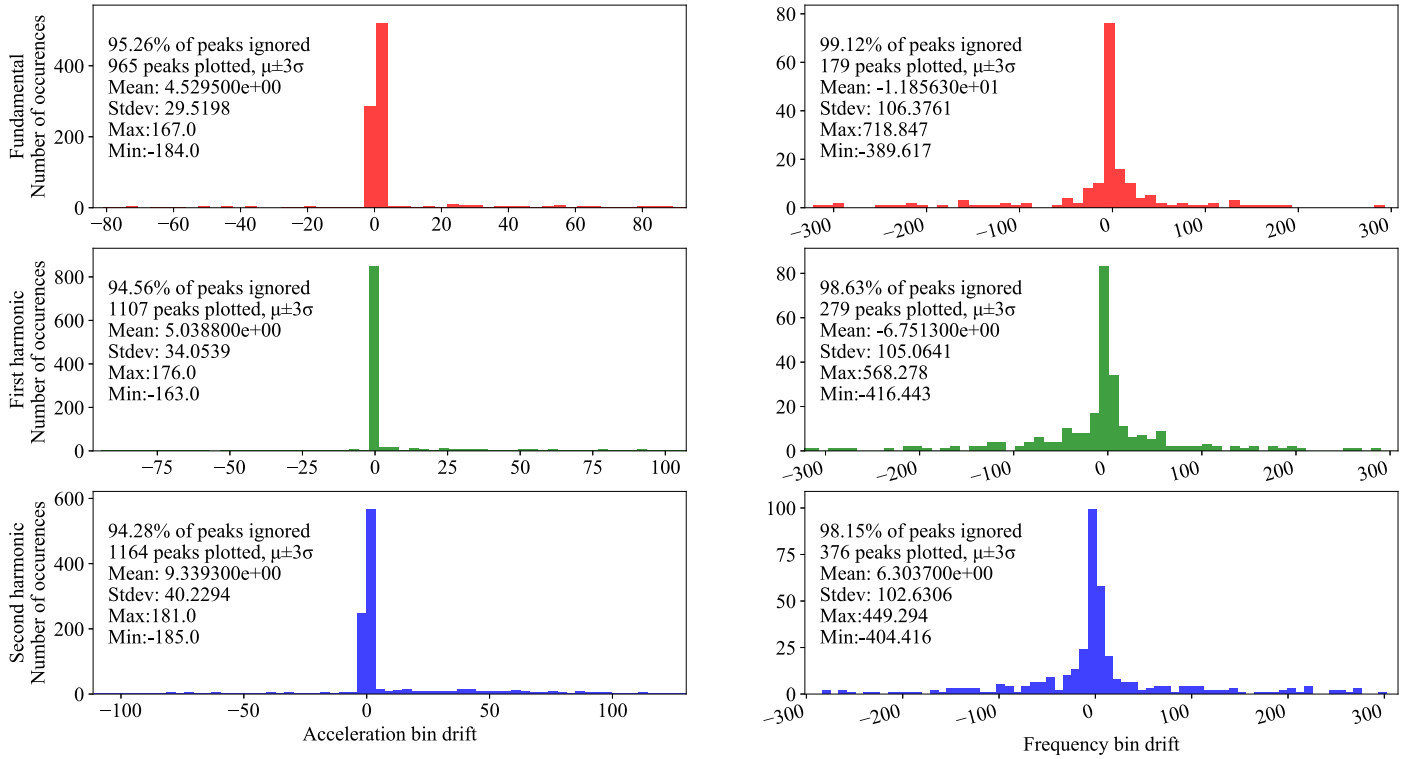

Figure 9. Graphs depicting acceleration (left) and frequency (right) bin drift between bfloat16 and single-precision results.

counterparts, and the maximum difference observed between any corresponding pair of bfloat16 and single-precision peaks was 2.7173%.

With a sufficiently large data set, the distribution of peak power differences when comparing bfloat16 and single-precision output follows a normal distribution, centered on zero. In the first three harmonics, it is seen that the standard deviation is consistently ≈0.6%.

5.3. Synthetic Data Set Results—Bin Drift

A reduced-precision detection pipeline must not only be able to reproduce the detection power to within an acceptable error but also localize the detection in the f–$\dot{f}$ plane correctly.

As such, it is important to understand whether reducing precision to bfloat16 also affects the location of a pulsar signature peak and its harmonics on the acceleration ($\dot{f}$) or frequency axis.

The results of this section aim to quantify the impact of reducing precision in FDAS from single precision to bfloat16 when measuring the frequency and acceleration ($\dot{f}$) properties of binary pulsar signatures.

The bin drifts on the frequency and acceleration axes between bfloat16 and single precision are shown in Figure 9. Across all tests performed, there is no measurable bin drift between the single- and double-precision versions of FDAS.

When comparing single precision and bfloat16, the majority (>94%) of peaks do not exhibit any bin drift; this is shown by the large fraction of peaks that are ignored in Figure 9. Although in cases where there is bin drift, it is usually limited to adjacent bins, i.e., the peak in the bfloat16 band is ±1 bins from the peak in single precision. This is shown by the tall peak at the center of the histograms.

5.3.1. Extreme Bin Drifts

As previously stated, there are some cases where the bin drift is extreme and looks to be far more than could be attributed to just accumulated numerical error. This is due to the limitations of the chosen peakfinding method misclassifying noise as a peak in the response in files where the signal of the pulsar is extremely weak, below the noise floor.

Figure 10 is important in understanding this particular limitation of our chosen analysis technique. Ignoring the red lines, this is a 2D histogram plot of the same data as in Figure 9. The y-axis represents the peak powers, so larger peaks are at the top of the histogram, and now the color intensity represents the number of occurrences in a given bin drift/peak power bin. Again, points with zero bin drift are ignored, leaving a vertical strip of data at a bin drift of ±1, which is more noticeable in the acceleration plots. However in both the acceleration and frequency plots, there is a band of very large bin drifts at lower values of peak power, around the noise floor. The explanation of this effect follows, assisted by the red lines.

In cases where a peak lies in or below the canopy of the noise, the large shifts occur under the following conditions:

1. In each harmonic band under investigation, the two tallest peaks have very similar heights (in both bfloat16 and single precision), within the range shown in Figure 8.
2. AND, when changing between precisions, the peak that was the higher of the two in bfloat16 becomes the lower of the two peaks in single precision, i.e., they "flip."

This is why it is only a small subset of many cases within the "noise canopy" where the effect is seen.

If both of the two highest peaks are due to noise, they could be located anywhere randomly in the harmonic band. This means that when the peakfinding methodology selects the "harmonic" from the harmonic band, it is effectively just

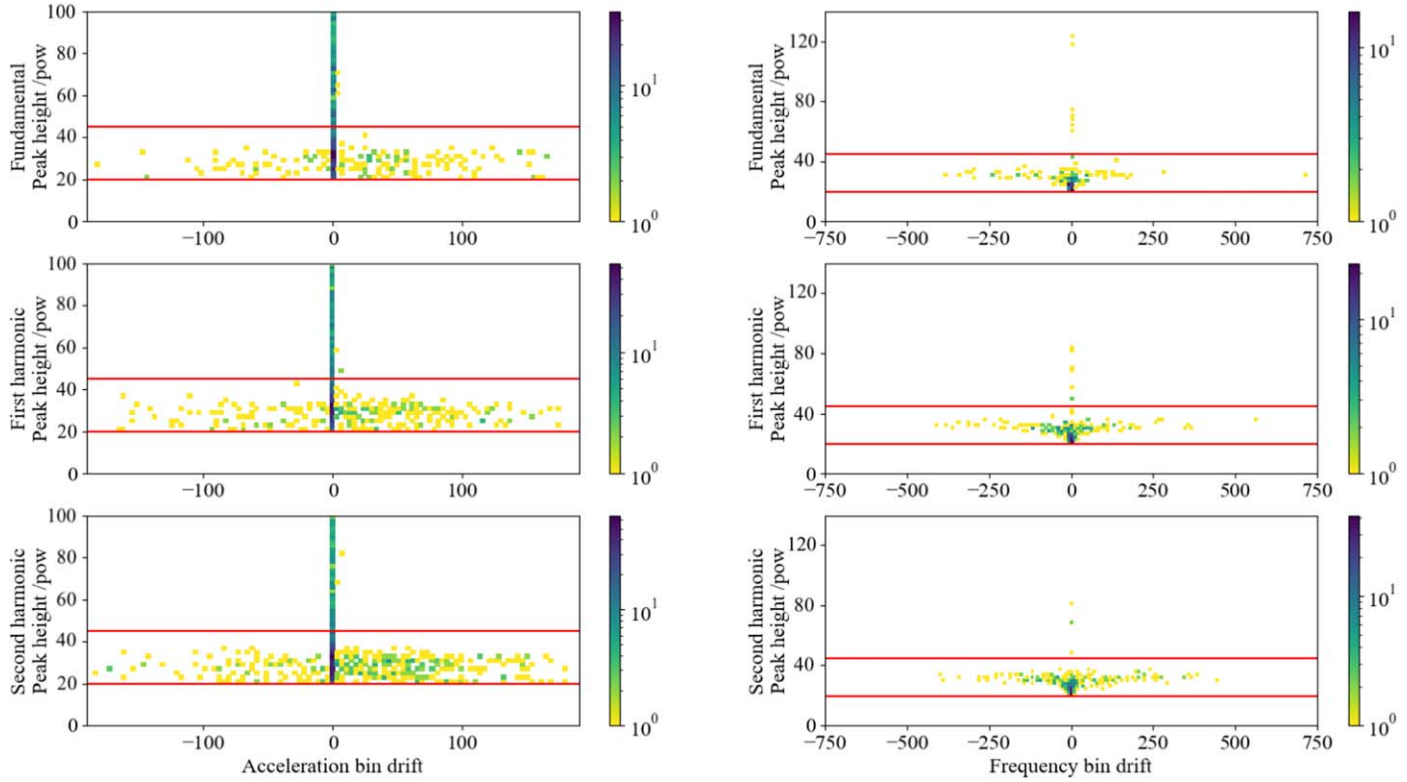

Figure 10. Two-dimensional histograms depicting the spread of acceleration (left) and frequency (right) bin drift with peak power; color scale represents absolute count. Upper and lower red bars represent measured maximum and minimum values of the largest noise value in noise-only data samples.

picking two random locations, which can lead to a very large measured bin drift.

The proportion of cases that we attribute to this explanation account for much less than 1% of the data set, and are extremely low-power peaks.

We have designed and carried out an experiment to measure the location of the relevant noise canopy. The experimental method is as follows:

1. Generate a selection of noise-only output f–$\dot{f}$ planes (without any pulsar signature present) by setting SIGPROC's `snrpeak` (Table 2) to 0, and process each one through the AA FDAS pipeline in single-precision mode.
2. For each extreme bin drift event (magnitude > 1) seen in Figure 10, note the frequency (calculated as $f = \frac{1}{\text{spin period}}$ Hz) of the pulsar that was in the synthetic data file that led to the extreme bin drift event. This frequency will have determined the width of the noise band the peak selection algorithm was picking from, as in Table 3.
3. For each extreme bin drift event, take an unused band of noise from the noise-only f–$\dot{f}$ planes with a frequency width of $1.5 \times f$, and record the highest point.
4. Record the maximum and minimum of all of the "highest points." These are the y-coordinates of the bounds (horizontal red lines) in Figure 10.

The bounds on Figure 10 demonstrate that all the observed extreme bin drift events occur in the region (the "noise canopy") where we would expect to see the randomly located noise peaks. In this region, a pulsar detection is already ambiguous, so we do not consider the performance of FDAS to be degraded by these edge cases.

5.3.2. Acceleration Skewing

A subtle skewing within the wide horizontal band in Figure 10 can also be observed, with values on the right-hand side of the x-axis occurring more frequently. The opposite trend is visible in Figure 11, where we have reversed the phenomenon that leads to the skewing.

The effect arises as an artifact of how the peak in each band is selected. In the bfloat16 arrays, multiple peaks are rounded to the same value (due to the coarse spacing of bfloat16 numbers), and therefore the direction in which the array is searched (for the max value) affects on which side the clashing value is first found. We have confirmed this by reversing the direction of the search and observing that the skew shows up on the opposite side, which can be seen in Figure 11.

5.4. Performance

In Figure 12, we present the speedup of the GPU-accelerated section of AA's implementation of FDAS with varying precision; this includes all steps listed within the blue box in Figure 1. This section represents a variable fraction of the overall execution time of a binary pulsar search, depending on the number of templates and number of DMs analyzed.

Measurements were taken with varying input data length (measured in number of samples). Each data point represents the average speedup of 256 runs, calculated by comparing the execution time of the modified precision version with the average execution time of the corresponding single-precision

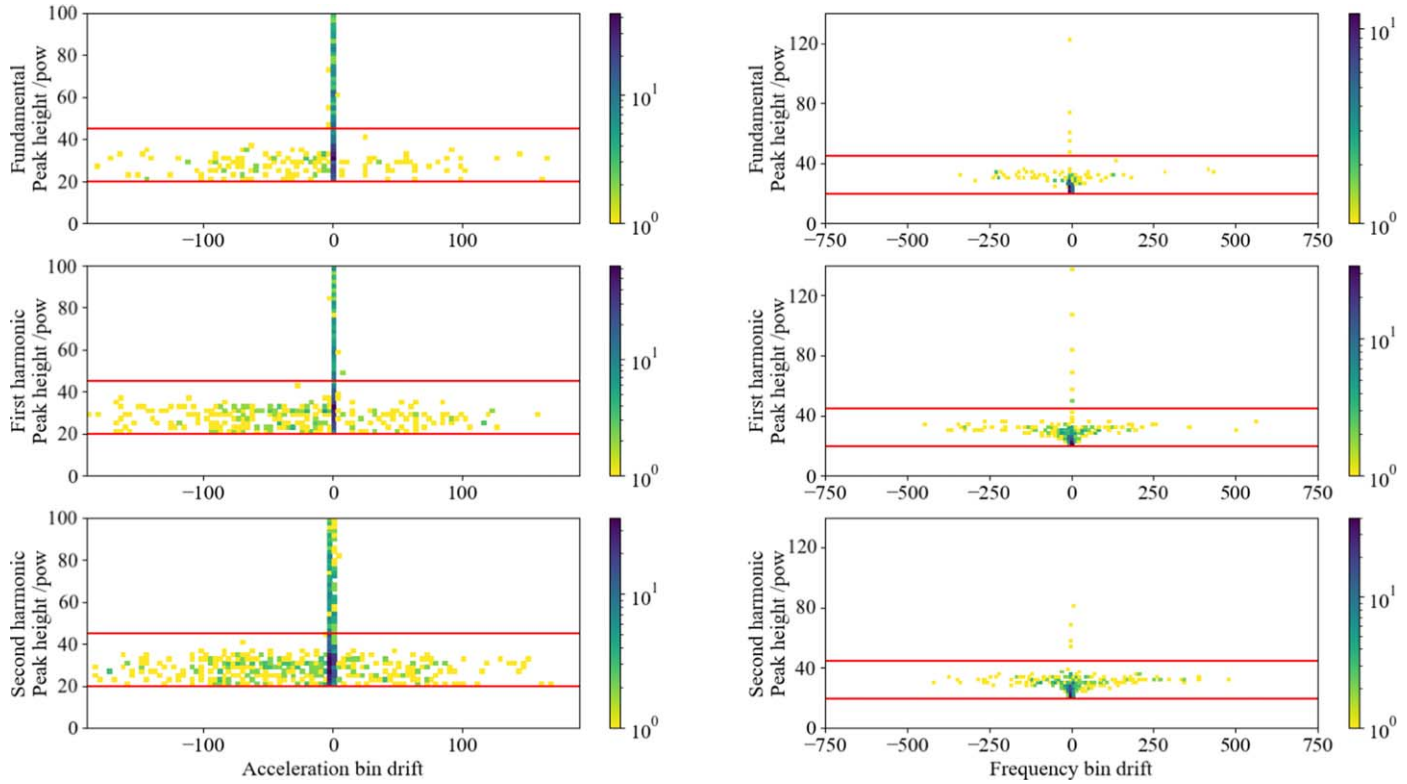

Figure 11. Two-dimensional histograms depicting the spread of acceleration (left) and frequency (right) bin drift with peak power; color scale represents absolute count. Upper and lower red bars represent measured maximum and minimum values of the largest noise value in noise-only data samples. The arrays were searched for the peak in the opposite direction to Figure 10, leading to the skewing appearing on the other side of the acceleration axis. The extreme values cause the automatically generated bin edges to differ slightly to Figure 10.

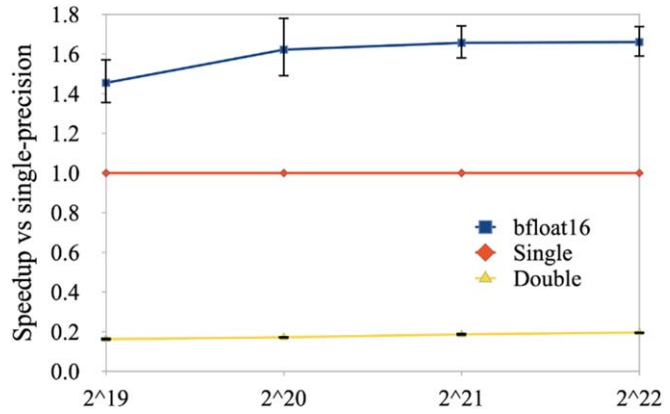

Figure 12. GPU performance comparison with varying precision and input length (number of samples) in AstroAccelerate.

version. The error bars are the measured standard deviation of the speedup.

Despite halving the internal memory traffic of FDAS by reducing the footprint of each number to 16 bits from 32 bits, the speedup achieved by bfloat16 is less than double, due to the overhead costs of implementing a mixed-precision code rather than a fully end-to-end reduced-precision pipeline.

5.5. Real Data

In Table 4, we present the ability of our bfloat16 version of AA to detect a pulsar in data from a real telescope, the GMRT. We used an observation of a 2.16 ms pulsar (J1544+4937) in a 2.8 hr compact orbit. The 550–750 MHz filterbank data were dedispersed at the nominal DM of 23.23 pc cm^{-3} (Bhattacharyya et al. 2013), generating a time series with a resolution of 81.92 μs. An observation time of 345 s was used to compare the bloat16 pipeline to the corresponding single- and double-precision versions; these results are also compared to those from PRESTO.

Due to slight variations in the exact number of samples processed by PRESTO and AA (owing to limitations on input length to the varying FFT implementations used), the frequency bin centers will be offset between the results from either software package. This is demonstrated by the fact that the peaks (fundamental and first harmonic) that have the greatest differences in detection frequencies also see the greatest differences in power. The peaks that have the most similar detection frequency (second and third harmonic) have extremely close detection powers. Overall, the discrepancy in the power values between PRESTO and AA (7.6% of summed power) is expected.

Importantly, the reported acceleration and frequency values match for all three choices of numerical precision of AA.

Linear regression on the first three peaks from the AA and PRESTO outputs leads to respective measured accelerations of the pulsar of 2.839 m s^{-2} (AA) and 2.931 m s^{-2} (PRESTO).

6. Conclusions

This work has demonstrated that it is possible to reduce the precision of the convolution section of the FDAS binary pulsar detection pipeline.

We have compared three different precisions, bfloat16, single-precision floating point (IEEE-754), and double-precision

Table 4
Comparison of J1544+4937 GMRT Data, Processed with PRESTO and AstroAccelerate in Varying Numerical Precisions

Harmonic	PRESTO Power	AA Power—bfloat16	AA Power—Single	AA Power—Double
Summed Power	326.1886	301.5	302.035	302.036
Fundamental	188.07	132	132.227	132.227
1	71.96	103	103.258	103.259
2	47.4545	48.5	48.777	48.777
3	18.7041	18	17.773	17.773
Harmonic	PRESTO "z" Bin	AA "z" Bin—bfloat16	AA "z" Bin—Single	AA "z" Bin—Double
Fundamental	0.75	0.5	0.5	0.5
1	1.5	1.25	1.25	1.25
2	1.5	1.5	1.5	1.5
3	2	2	2	2
Acceleration	2.931 m s^{-2}	2.839 m s^{-2}	2.839 m s^{-2}	2.839 m s^{-2}
Harmonic	PRESTO Frequency	AA Frequency—bfloat16	AA Frequency—Single	AA Frequency—Double
Fundamental	463.098 Hz	463.097 Hz	463.097 Hz	463.097 Hz
1	926.196 Hz	926.192 Hz	926.192 Hz	926.192 Hz
2	1389.29 Hz	1389.29 Hz	1389.29 Hz	1389.29 Hz
3	1852.39 Hz	1852.39 Hz	1852.39 Hz	1852.39 Hz

Note. Power values taken from the f–$\dot{f}$ plane in each program. As can be seen, slightly different frequency and "z" bin measurements were made in either program, leading to an approximately 7.6% loss in total summed power across four harmonics.

floating point (IEEE-754), across a wide range of binary system parameters, including real data of a millisecond pulsar in a compact orbit.

Comparing 64-bit double precision with 32-bit single precision, we have found that the benefit of using double precision is very limited, and it usually does not result in a measurable difference when the results are rounded to single precision (as may be required for later processing). Given the performance impact of using 64-bit double precision (a five-fold reduction in throughput), these results do not justify increasing precision in the FDAS pipeline of AA from 32-bit single precision to 64-bit double precision.

Comparing 32-bit single precision with the 16-bit bfloat16 implementation, there is a small but measurable difference between the output of the bfloat16 implementation of FDAS and the single-precision implementation. However, in all cases the peak power is within a few percent of the single-precision result, with no strong bias to be either higher or lower. Lowering the memory traffic requirement of FDAS has led to an approximately 1.6× speedup. The improved compute performance of bfloat16 is not a factor in this case, as the pipeline is memory bandwidth bound, with a low computational overhead (Adámek 2021). The speedup enables users of GPU-accelerated FDAS to increase their pulsar parameter search space on a given set of hardware when performing a real-time search, or to reduce their upfront hardware requirement for a given search.

This work has received support from STFC Grant (ST/T000570/1). The authors would like to acknowledge the use of the University of Oxford Advanced Research Computing (ARC) facility in carrying out this work (Richards 2015). The authors would also like to express their gratitude to the Research Centre for Theoretical Physics and Astrophysics, Institute of Physics, Silesian University in Opava for institutional support.

The National Radio Astronomy Observatory is a facility of the National Science Foundation operated under cooperative agreement by Associated Universities, Inc. S.M.R. is a CIFAR Fellow and is supported by the NSF Physics Frontiers Center awards 1430284 and 2020265.

ORCID iDs

Jack White https://orcid.org/0000-0003-2690-6858
Karel Adámek https://orcid.org/0000-0003-2797-0595
Jayanta Roy https://orcid.org/0000-0002-2892-8025
Sofia Dimoudi https://orcid.org/0000-0002-0967-1332
Scott M. Ransom https://orcid.org/0000-0001-5799-9714
Wesley Armour https://orcid.org/0000-0003-1756-3064

References

Adámek, K., & Armour, W. 2019, in ASP Conf. Ser. 523, Astronomical Data Analysis Software and Systems XXVIII, ed. P. J. Teuben (San Francisco, CA: ASP), 489
Adámek, K., Dimoudi, S., Giles, M., & Armour, W. 2020a, in ASP Conf. Ser. 522, Astronomical Data Analysis Software and Systems XXVII, ed. P. Ballester (San Francisco, CA: ASP), 477
Adámek, K., Dimoudi, S., Giles, M., & Armour, W. 2020c, ACM Trans. Archit. Code Optim., 17, 18
Adámek, K., Novotný, J., Dimoudi, S., & Armour, W. 2020b, in ASP Conf. Ser. 527, Astronomical Data Analysis Software and Systems XXIX, ed. R. Pizzo et al. (San Francisco, CA: ASP), 671
Adámek, K., Novotný, J., Thiyagalingam, J., & Armour, W. 2021, IEEE Access, 9, 18167
Andersen, B. C., & Ransom, S. M. 2018, ApJL, 863, L13
Armour, W., Karastergiou, A., Giles, M., et al. 2012, in ASP Conf. Ser. 461, Astronomical Data Analysis Software and Systems XXI, ed. P. Ballester et al. (San Francisco, CA: ASP), 33
Bhattacharya, D., & van den Heuvel, E. P. J. 1991, PhR, 203, 1
Bhattacharyya, B., Roy, J., Gupta, Y., et al. 2013, ApJL, 773, L12
Dimoudi, S., Adámek, K., Thiagaraj, P., et al. 2018, ApJS, 239, 28
Dimoudi, S., & Armour, W. 2017, in ASP Conf. Ser. 512, Astronomical Data Analysis Software and Systems XXV, ed. N. P. F. Lorente et al. (San Francisco, CA: ASP), 599
Freire, P. C., Kramer, M., & Lyne, A. G. 2001, MNRAS, 322, 885
Kansabanik, D., Bhattacharyya, B., Roy, J., & Stappers, B. 2021, ApJ, 920, 58

Kramer, M., Stairs, I. H., Manchester, R. N., et al. 2021, PhRvX, 11, 041050
Levin, L., Armour, W., Baffa, C., et al. 2017, in IAU Symp. 337, Pulsar Astrophysics the Next Fifty Years (Cambridge: Cambridge Univ. Press), 171
Lorimer, D. R. 2011, SIGPROC: Pulsar Signal Processing Programs, Astrophysics Source Code Library, ascl:1107.016
Middleditch, J., & Kristian, J. 1984, ApJ, 279, 157
Mishra, S. M., Tiwari, A., Shekhawat, H. S., et al. 2022, in 2022 32nd Int. Conf. Radioelektronika (RADIOELEKTRONIKA) (Piscataway, NJ: IEEE), 1
Morello, V., Rajwade, K. M., & Stappers, B. W. 2022, MNRAS, 510, 1393
Murillo, R., Barrio, A. A. D., & Botella, G. 2022, in 2022 Annual Modeling and Simulation Conf. (ANNSIM) (Piscataway, NJ: IEEE), 152
Rajwade, K., Stappers, B., Williams, C., et al. 2020, Proc. SPIE, 11447, 114470J
Ransom, S. M., Eikenberry, S. S., & Middleditch, J. 2002, AJ, 124, 1788
Richards, A. 2015, University of Oxford Advanced Research Computing, Zenodo, doi:10.5281/zenodo.22558
Ridolfi, A., Gautam, T., Freire, P. C. C., et al. 2021, MNRAS, 504, 1407
Swihart, S. J., Strader, J., Aydi, E., et al. 2021, ApJ, 909, 185
Taylor, J. H., & Weisberg, J. M. 1982, ApJ, 253, 908