



PINT: Maximum-likelihood Estimation of Pulsar Timing Noise Parameters

Abhimanyu Susobhanan^{1,2} , David L. Kaplan² , Anne M. Archibald³ , Jing Luo^{4,19} , Paul S. Ray⁵ , Timothy T. Pennucci⁶ , Scott M. Ransom⁷ , Gabriella Agazie² , William Fiore^{8,9} , Bjorn Larsen¹⁰ , Patrick O'Neill³ , Rutger van Haasteren¹ , Akash Anumarlapudi² , Matteo Bachetti¹¹ , Deven Bhakta¹² , Chloe A. Champagne¹³ , H. Thankful Cromartie^{14,21} , Paul B. Demorest¹⁵ , Ross J. Jennings^{8,9,20} , Matthew Kerr⁵ , Sasha Levina¹⁶ , Alexander McEwen² , Brent J. Shapiro-Albert^{8,9,17} , and Joseph K. Swiggum^{18,20}

¹ Max-Planck-Institut für Gravitationsphysik (Albert-Einstein-Institut), Leibniz Universität Hannover, Callinstraße 38, D-30167, Hannover, Germany; abhimanyu.susobhanan@nanograv.org

² Center for Gravitation, Cosmology and Astrophysics, Department of Physics, University of Wisconsin-Milwaukee, P.O. Box 413, Milwaukee, WI 53201, USA
³ Newcastle University, Newcastle upon Tyne, Tyne and Wear, NE1 7RU, UK

⁴ Department of Astronomy & Astrophysics, University of Toronto, 50 Saint George Street, Toronto, ON M5S 3H4, Canada

⁵ Space Science Division, Naval Research Laboratory, Washington, DC 20375-5352, USA

⁶ Institute of Physics and Astronomy, Eötvös Loránd University, Pázmány P. s. 1/A, 1117 Budapest, Hungary

⁷ National Radio Astronomy Observatory, 520 Edgemont Road, Charlottesville, VA 22903, USA

⁸ Department of Physics and Astronomy, West Virginia University, P.O. Box 6315, Morgantown, WV 26506, USA

⁹ Center for Gravitational Waves and Cosmology, West Virginia University, Chestnut Ridge Research Building, Morgantown, WV 26505, USA

¹⁰ Department of Physics, Yale University, New Haven, CT 06520, USA

¹¹ Istituto Nazionale di Astrofisica-Osservatorio Astronomico di Cagliari, via della Scienza 5, I-09047 Selargius (CA), Italy

¹² University of Virginia, Department of Astronomy, P.O. Box 400325, Charlottesville, VA 22904, USA

¹³ Electrical and Computer Engineering Department, Vanderbilt University, Nashville, TN 37235-1824, USA

¹⁴ National Research Council Research Associate, National Academy of Sciences, Washington, DC 20001, USA

¹⁵ National Radio Astronomy Observatory, 1003 Lopezville Road, Socorro, NM 87801, USA

¹⁶ William H. Miller III Department of Physics and Astronomy, Johns Hopkins University, 3400 N. Charles Street, Baltimore, MD 21218, USA

¹⁷ Giant Army, 915A 17th Avenue, Seattle, WA 98122, USA

¹⁸ Department of Physics, Lafayette College, Easton, PA 18042, USA

Received 2024 May 3; revised 2024 June 11; accepted 2024 June 18; published 2024 August 14

Abstract

PINT is a pure-Python framework for high-precision pulsar timing developed on top of widely used and well-tested Python libraries, supporting both interactive and programmatic data analysis workflows. We present a new frequentist framework within PINT to characterize the single-pulsar noise processes present in pulsar timing data sets. This framework enables parameter estimation for both uncorrelated and correlated noise processes, as well as model comparison between different timing and noise models in a computationally inexpensive way. We demonstrate the efficacy of the new framework by applying it to simulated data sets as well as a real data set of PSR B1855+09. We also describe the new features implemented in PINT since it was first described in the literature.

Unified Astronomy Thesaurus concepts: [Pulsars \(1306\)](#); [Astronomy software \(1855\)](#); [Astronomy data analysis \(1858\)](#)

1. Introduction

Since their discovery, pulsars have been used as celestial laboratories to probe a wide range of time-domain phenomena, owing to their remarkable rotational stability. This is especially true for millisecond pulsars (MSPs), which are pulsars with millisecond-scale rotational periods spun up by accretion from their companion stars (Manchester 2017). Such applications include constraints on the neutron star equation of state (e.g., Cromartie et al. 2020), the discovery of exoplanets (Wolszczan & Frail 1992), tests of theories of gravity (e.g., Kramer et al. 2021), probing the interstellar medium (e.g., Donner et al. 2020) and solar wind (e.g., Tiburzi et al. 2021), the creation of an international time standard (Hobbs et al. 2019),

characterizing the uncertainties present in the solar system ephemerides (e.g., Caballero et al. 2018), and more. These exciting results were produced with the help of pulsar timing, the technique of tracking a pulsar's rotational phase using the measured times of arrival (TOAs) of its pulses, allowing it to be used as a celestial clock (Lorimer & Kramer 2012). Pulsar timing was instrumental in the recent evidence for a nanohertz gravitational-wave background (Agazie et al. 2023b, 2024a; Antoniadis et al. 2023; Reardon et al. 2023; Xu et al. 2023) by Pulsar Timing Array (PTA) experiments (Sazhin 1978; Foster & Backer 1990), inaugurating the era of nanohertz gravitational-wave astronomy.

High-precision pulsar timing experiments such as PTAs and tests of gravity require modeling the TOAs down to nanosecond-level precision. A pulsar timing model or pulsar ephemeris is a generative mathematical description of the deterministic astrophysical processes influencing the measured TOAs. These processes include pulsar rotation, pulsar binary dynamics, interstellar dispersion, solar system dynamics, proper motion, and solar wind, etc., and they must be accurately incorporated into the timing model to achieve the required precision (Edwards et al. 2006). The timing model is

¹⁹ Deceased.

²⁰ NANOGrav Physics Frontiers Center Postdoctoral Fellow.

²¹ Resident at Naval Research Laboratory, Washington, DC 20375, USA.



Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](#). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

often accompanied by a noise model, which incorporates stochastic processes affecting the TOAs, such as radiometer noise, pulse jitter, rotational irregularities, interstellar medium variability, and radio frequency interference (RFI), etc. (Agazie et al. 2023a).

In practice, pulsar timing involves the creation and incremental refinement of a pulsar timing model that matches the observed TOAs, typically using frequentist methods. This is usually performed using one of the three standard software packages: `tempo` (Nice et al. 2015), `tempo2` (Edwards et al. 2006; Hobbs et al. 2006), or `PINT` (Luo et al. 2021), often in an interactive manner. Noise characterization is usually performed separately in the Bayesian paradigm, using software packages such as `ENTERPRISE` (Johnson et al. 2024) and `TEMPOEST` (Lentati et al. 2014), starting from a post-fit timing model. `ENTERPRISE` can also be used to characterize deterministic and stochastic signals common across multiple pulsars, such as the stochastic gravitational-wave background and solar system ephemeris errors (e.g., Vallisneri et al. 2020; Agazie et al. 2023b). Since accurate noise characterization is necessary for accurate timing and vice versa (e.g., Coles et al. 2011), the timing and noise characterization steps must be repeated alternately to ensure that the results remain accurate and stable, and this turns out to be expensive both in time and computation.

`PINT`²² is a flexible pure-Python framework for pulsar timing that is written on top of widely used scientific computing libraries, such as `numpy` (Harris et al. 2020), `scipy` (Virtanen et al. 2020), `astropy` (Price-Whelan et al. 2022), and `matplotlib` (Hunter 2007), developed under the aegis of the North American Nanohertz Observatory for Gravitational Waves (NANOGrav: Demorest et al. 2012). The reliability of this package is ensured via its reliance on these well-tested libraries, strict version control, and an extensive continuous integration and testing suite. `PINT` is primarily designed to be used as a Python library to ensure that (a) *all* of its functionality remains easily accessible to the user; (b) it is easily extensible; and (c) it can be easily composed with other Python packages. It also provides a graphical user interface (named `pintk`) and command-line tools for specific tasks. In comparison, `tempo` and `tempo2`, written in FORTRAN and C-style C++, respectively, are primarily designed to be used as command-line applications (`tempo2` also has a graphical user interface named `plk` and a Python wrapper named `libstempo`; Vallisneri 2020).

In this paper, we present a new frequentist framework in `PINT` for characterizing the noise processes affecting pulsar timing, allowing the noise parameters to be fit together with the timing model parameters in a maximum-likelihood way for a single pulsar. Our framework also enables model comparisons within `PINT` using the Akaike Information Criterion (AIC: Burnham & Anderson 2004). The new framework should allow us to easily incorporate noise characterization into pulsar timing pipelines and interactive workflows, obtaining relatively quick noise estimates and enabling swift refinement of noise models. This is in contrast to conventional Bayesian noise characterization, which is performed as a separate step from

pulsar timing and is relatively more computationally expensive, but can also include common noise terms between pulsars (e.g., Agazie et al. 2023a). Our framework can complement the Bayesian methods for noise characterization in the following ways. During the initial data preparation and combination stages, our method can hasten the iterative refinement of noise models. The frequentist estimates can be used to cross-check the Bayesian results. They can also act as starting points for Markov Chain Monte Carlo (MCMC) samplers (e.g., Jones & Qin 2022), allowing them to burn in faster. Finally, in cases where Bayesian analysis is not considered worth the cost, our method provides an inexpensive alternative for noise characterization.

This paper is arranged as follows. Section 2 provides a quick overview of pulsar timing. Section 3 briefly describes the fitting methods used in `PINT`. The newly implemented methods for estimating noise parameters are described in Section 4. Model comparison using the AIC is described in Section 5. We demonstrate the new framework using simulations in Section 6 and using a real data set in Section 7. In Section 8, we discuss some of the new developments in `PINT`, implemented since the publication of Luo et al. (2021), which initially described the package. Finally, we summarize our work in Section 9.

2. A Brief Overview of Pulsar Timing

2.1. TOAs

The primary measurable quantity in pulsar timing is the TOA. In conventional pulsar timing, the pulsar time-series data are coherently averaged (“folded”) into an integrated pulse profile to improve its signal-to-noise ratio and to mitigate the effects of pulse-to-pulse variations (Lorimer & Kramer 2012). A TOA can be measured from an integrated pulse profile by matching it against a noise-free template (Taylor 1992). In the traditional narrowband paradigm, the observation is split into multiple frequency sub-bands, and the TOAs are estimated in each sub-band independently. On the other hand, in the more recently developed wideband paradigm, a frequency-resolved integrated pulse profile is cross-correlated against a two-dimensional template in frequency and pulse phase to simultaneously measure a TOA and a dispersion measure (DM)²³ for the whole observation (Pennucci et al. 2014; Pennucci 2019). Algorithms for folding and manipulating integrated pulse profiles and for measuring TOAs are available in packages like `DSPSR` (van Straten & Bailes 2011), `PRESTO` (Ransom 2001), `PSRCHIVE` (Hotan et al. 2004), and `PulsePortraiture` (Pennucci et al. 2014; Pennucci 2019). We restrict ourselves to the narrowband paradigm in this work for the sake of simplicity, unless explicitly stated otherwise.

The TOAs are generally recorded against local observatory clocks. `PINT` applies a series of clock corrections to the measured TOAs, bringing them to the Barycentric Dynamical Time (TDB), a relativistic timescale defined at the solar system barycenter (SSB). Detailed descriptions of clock corrections may be found in Hobbs et al. (2006) and Luo et al. (2021). Note that `tempo2` uses the Barycentric Coordinate Time (TCB) by default, which differs from TDB by a constant factor, and timing models using TCB need to be converted to TDB to be `PINT`-compatible (see Section 8.1 for the newly available TCB-to-TDB conversion feature).

²² Available as `pint-pulsar` via `pip` and `conda` package managers. The source code is available at <https://github.com/nanograv/PINT>. The documentation is available at <https://nanograv-pint.readthedocs.io/>. This paper corresponds to `PINT` v1.0.0 (Agazie et al. 2024b). “`PINT`” is an acronym for “`PINT` Is Not `TEMPO3`.”

²³ DM quantifies the interstellar dispersion of the radio waves and is proportional to the electron column density along the line of sight to the pulsar.

In PINT, a set of observed TOAs is represented by the TOAs class in the `pint.toa` module. See Luo et al. (2021) for details on the internal representation of TOAs. They are usually stored in human-readable text files known as “tim” files, which can be read using the `pint.toa.get_TOAs()` function.

2.2. The Timing and Noise Model

Pulsar timing involves connecting the pulse number N , related to the rotational phase of the pulsar as $\Phi(N) = 2\pi N$, to the time of emission t_{em} , as

$$N = N_0 + f(t_{\text{em}} - t_0) + \frac{1}{2}\dot{f}(t_{\text{em}} - t_0)^2 + \dots, \quad (1)$$

where f is the rotational frequency, \dot{f} is the rotational frequency derivative, and t_0 is a fiducial time. The right-hand side of the above equation may also include higher-order frequency derivative terms and rotational irregularity effects such as glitches (Hobbs et al. 2006). The measured TOA t_{arr} is related to t_{em} by

$$t_{\text{arr}} = t_{\text{em}} + \Delta_B + \Delta_{\text{DM}} + \Delta_{\odot} + \dots + \mathcal{N}. \quad (2)$$

Here, the delay Δ_B originates from the motion of the pulsar in a binary system and includes the Rømer delay, Shapiro delay, and Einstein delay (Damour & Deruelle 1986). Δ_{DM} denotes the delay caused by interstellar dispersion and is given by $\Delta_{\text{DM}} = \mathcal{K}D/\nu^2$, where ν is the observing frequency, D is the DM, and \mathcal{K} is known as the DM constant (Lorimer & Kramer 2012). Δ_{\odot} denotes the delays caused by the solar system motion, including the Rømer delay and the Shapiro delay,²⁴ and they are computed using the solar system ephemerides published by space agencies (e.g., Park et al. 2021). Detailed descriptions of the various timing model components can be found in Edwards et al. (2006) and Luo et al. (2021). Finally, \mathcal{N} denotes the noise present in the TOA, including correlated and uncorrelated noise components (see Section 4.1). Note that in Equation (2), we have ignored an unmeasurable constant term corresponding to the light travel time in vacuum from the pulsar (or the binary barycenter in the case of binary pulsars) to the SSB at some fiducial epoch.

The various timing and noise model components available in PINT are listed in Table 1, and the new/updated components are highlighted therein. These are available in the `pint.models` module of PINT. The timing and noise model as a whole, comprising these components, is represented by the `TimingModel` class in the `pint.models` module. Pulsar ephemerides are usually stored in human-readable text files known as “par” files and can be read using the `pint.models.get_model()` function. A pair of “par” and “tim” files belonging to the same pulsar can be read together using the `pint.models.get_model_and_toas()` function.²⁵

2.3. Timing Residuals

The timing model can be used to predict the phase Φ_i associated with each TOA, and this allows us to compute the

²⁴ Solar system Einstein delay is corrected for while converting the TOAs into TDB, since it is part of the definition of the TDB.

²⁵ The latter is the preferred way, since some parts of the timing model, such as clock and solar system ephemeris information, can affect how the TOAs object is constructed.

timing residuals:

$$r_i = \frac{\Phi_i - 2\pi N[\Phi_i]}{F}, \quad (3)$$

where $N[\Phi_i]$ is the integer closest to $\Phi_i/2\pi$ and F is the pulse frequency.²⁶ The procedure for computing a timing residual can be summarized as follows:

1. Apply the various clock corrections to convert the measured TOA from the observatory timescale to a timescale defined at the SSB.
2. Successively correct for the various delays influencing the TOA, bringing it to the pulsar frame (Equation (2)).
3. Compute the pulsar rotational phase at the corrected TOA using the rotational frequency and its derivatives (Equation (1)).
4. Successively correct for the other effects that influence the rotational phase of the pulsar.
5. Compute the phase residual by subtracting the closest integer from the estimated rotational phase. The timing residual is the phase residual divided by the instantaneous rotational frequency (Equation (3)).

In the `pint.residuals` module, narrowband timing residuals are represented by the `Residuals` class, and wideband residuals are represented by the `WidebandTOAResiduals` class.

How we fit a timing model to the observed TOAs using timing residuals is described in the next section.

3. Fitting for Timing Model Parameters

3.1. Fitting in the White-noise-only Case

If the initial (pre-fit) timing model is sufficiently close to its best-fit counterpart, the pre-fit timing residuals \mathbf{r} and the post-fit timing residuals \mathbf{s} are related by the linear relation

$$\mathbf{r} - \mathbf{s} = \mathbf{M}\boldsymbol{\beta}, \quad (4)$$

where \mathbf{r} and \mathbf{s} are n -dimensional vectors containing elements r_i and s_i , respectively, \mathbf{M} is the $n \times p$ -dimensional pulsar timing design matrix containing partial derivatives $\frac{\partial s_i}{\partial b_\alpha}$ with respect to the p timing model parameters b_α , and $\boldsymbol{\beta}$ is a p -dimensional vector containing timing model parameter deviations β_α from their best-fit values \hat{b}_α (i.e., $\beta_\alpha = \hat{b}_\alpha - b_\alpha$), where n is the number of TOAs, and p is the number of timing model parameters. In the absence of correlated noise, the log-likelihood function can be written, up to an additive constant, as

$$\ln L = -\frac{1}{2}\mathbf{s}^T \mathbf{N}^{-1} \mathbf{s} - \frac{1}{2}\ln \det \mathbf{N}, \quad (5)$$

²⁶ In PINT, F can be the rotational frequency F_0 at some fiducial epoch, the instantaneous rotational frequency $F(t_i)$ computed incorporating the rotational frequency derivatives, or the topocentric pulse frequency $\tilde{F}(t_i) = d\Phi/dt_{\text{arr}}$ incorporating all delay and phase corrections computed using finite differences. On the other hand, `tempo2` uses $F = F_0$, and `tempo` uses the instantaneous rotational frequency that also incorporates the first rotational frequency derivative. The topocentric frequency is the most accurate option in principle, but it is more expensive to compute and the differences between different options are negligible for MSPs in most cases. However, not using the topocentric frequency can have a significant impact on the timing accuracy in the case of young or accreting pulsars with large frequency derivatives (e.g., Ray et al. 2019).

Table 1
Updated List of Timing Model Components Available in PINT (in the `pint.models` Module)

Component	Description	References
<i>Pulsar rotation, rotational phase, and rotational irregularities</i>		
Spindown	Taylor-series representation of the pulsar rotation	Backer & Hellings (1986)
PiecewiseSpindown §	Piecewise-constant corrections to pulsar rotation	
Glitch	Pulsar glitches	Hobbs et al. (2006)
IFunc	Piecewise-constant or spline representation of rotational period	Deng et al. (2012)
WaveX § ‡	Fourier series representation of ARN (supersedes the deprecated <code>Wave</code> model)	Hobbs et al. (2006) Sections 4.1.3, 4.3
PLRedNoise	Fourier GP representation of ARN	Lentati et al. (2014)
AbsPhase	Reference TOA with respect to which the rotational phase is measured	
PhaseOffset § ‡	Overall phase offset between reference TOA and the physical TOAs	Section 4.2
PhaseJump ¶	Phase offsets between TOAs measured using different systems	Hobbs et al. (2006)
<i>Binary system</i>		
BinaryBT	Simple parameterized post-Keplerian binary model (only Rømer delay)	Blandford & Teukolsky (1976)
BinaryBTPiecewise § ‡	Similar to <code>BinaryBT</code> , but with piecewise-constant orbital parameters	
BinaryDD	Parameterized post-Keplerian model (with Shapiro delay and Einstein delay)	Damour & Deruelle (1986)
BinaryDDGR §	Similar to <code>BinaryDD</code> , but assumes General Relativity	Taylor & Weisberg (1989)
BinaryDDH §	Similar to <code>BinaryDD</code> , but uses a harmonic representation of Shapiro delay (for low-inclination systems)	Freire & Wex (2010) Weisberg & Huang (2016)
BinaryDDK	Similar to <code>BinaryDD</code> , but includes Kopeikin delay	Damour & Taylor (1992) Kopeikin (1995, 1996)
BinaryDDS §	Similar to <code>BinaryDD</code> , but uses an alternative representation of Shapiro delay (for almost edge-on orbits)	Kramer et al. (2006) Rafikov & Lai (2006)
BinaryELL1 †	Binary model specialized for nearly circular orbits using Laplace–Lagrange parameters (includes up to third-order terms in eccentricity)	Lange et al. (2001) Zhu et al. (2018) Fiore et al. (2023)
BinaryELL1H	Similar to <code>BinaryELL1</code> , but uses a harmonic representation of Shapiro delay (for low-inclination systems)	Freire & Wex (2010)
BinaryELL1k §	Similar to <code>BinaryELL1</code> , but includes an exact treatment of advance of periastron (for highly relativistic or tidally interacting binaries)	Susobhanan et al. (2018)
<i>Interstellar dispersion and DM variations</i>		
DispersionDM ¶	Taylor-series representation of DM	Backer & Hellings (1986)
DispersionDMX ¶	Piecewise-constant representation of DM variations	Arzoumanian et al. (2015)
DMWaveX § ‡	Fourier series representation of DM variations	Sections 4.1.3, 4.3
PLDMNoise §	Fourier GP representation of DM variations	Lentati et al. (2014)
DispersionJump §	Offsets between wideband DMs measured using different systems (no delay)	Alam et al. (2021)
FDJumpDM §	DM offsets between narrowband TOAs measured using different systems	
<i>Astrometry and solar system delays</i>		

Table 1
(Continued)

Component	Description	References
AstrometryEcliptic	Astrometry in ecliptic coordinates	Edwards et al. (2006)
AstrometryEquatorial	Astrometry in equatorial coordinates	
SolarSystemShapiro	Solar system Shapiro delay	Shapiro (1964)
<i>Solar wind</i>		
SolarWindDispersion †	Solar wind model assuming a radial power-law relation for the electron density	Edwards et al. (2006) You et al. (2007, 2012)
SolarWindDispersionX § ‡	Similar to SolarWindDispersion, but with a piecewise-constant representation of the electron density.	Madison et al. (2019) Hazboun et al. (2022)
<i>Troposphere</i>		
TroposphereDelay	Tropospheric zenith hydrostatic delay	Davis et al. (1985) Niell (1996)
<i>Time-uncorrelated noise</i>		
ScaleToaError	Modifications to the measured TOA uncertainties	Lentati et al. (2014)
ScaleDMError §	Modifications to the measured wideband DM uncertainties	Alam et al. (2021)
EcorrNoise †	Correlation between TOAs measured from the same observation.	Arzoumanian et al. (2015) Johnson et al. (2024)
<i>Frequency-dependent profile evolution</i>		
FD ¶	Frequency-dependent profile evolution	Arzoumanian et al. (2015)
FDJump §	System and frequency-dependent profile evolution	Appendix

Notes. “§” denotes newly implemented components and “†” denotes components that have had significant changes since the publication of Luo et al. (2021). The `BinaryELL1` component now includes up to $\mathcal{O}(e^3)$ terms in orbital eccentricity. The `SolarWindDispersion` component now allows electron density radial power-law indices other than 2. The `EcorrNoise` component now incorporates a faster algorithm for inverting the TOA covariance matrix. “¶” denotes components that were renamed after the publication of Luo et al. (2021). “‡” denotes components only available in `PINT`. The choice of the binary model is specified in the “par” files using the “BINARY” keyword, e.g., “BINARY ELL1.” Additionally, there is a generic timing model in `tempo2` called the “T2” model, which chooses the underlying binary model based on the available binary parameters. When “BINARY T2” is encountered, `PINT` emits an informative error message indicating the best guess for the underlying binary model. It can also construct the binary timing model based on this guess optionally.

where $\mathbf{N} = \text{diag}[\varsigma_i^2]$ is the diagonal uncorrelated (white-) noise TOA covariance matrix, and ς_i represents the scaled TOA uncertainties (see Section 4 for a detailed explanation). The quantity $\mathbf{s}^T \mathbf{N}^{-1} \mathbf{s}$ appearing in the first term is usually referred to as the chi-squared (χ^2).

The parameter deviations $\boldsymbol{\beta}$ can be estimated by maximizing the above likelihood function. If \mathbf{N} is fixed, the maximum-likelihood estimate involves minimizing the χ^2 , and this turns out to be

$$\hat{\boldsymbol{\beta}} = (\mathbf{M}^T \mathbf{N}^{-1} \mathbf{M})^{-1} \mathbf{M}^T \mathbf{N}^{-1} \mathbf{r}, \quad (6)$$

with a parameter covariance matrix

$$\mathbf{K}_\beta = \frac{1}{n - p} (\mathbf{s}^T \mathbf{N}^{-1} \mathbf{s}) (\mathbf{M}^T \mathbf{N}^{-1} \mathbf{M})^{-1}. \quad (7)$$

The conventional fitting algorithm for the pulsar timing model involves updating the parameter values $\mathbf{b} \rightarrow \mathbf{b} - \hat{\boldsymbol{\beta}}$. In practice, the right-hand side of Equation (6) is evaluated with the help of a singular value decomposition (SVD) in `PINT` (see also Section 3.3.1; Press et al. 1992). In contrast, `tempo2` uses a QR decomposition for this purpose.

This method is usually referred to as weighted least-squares (WLS). See, e.g., Coles et al. (2011) for a detailed discussion on this method.

3.2. Fitting in the Presence of Correlated Noise

The more general case of the above fitting algorithm that accounts for the presence of correlated noise involves maximizing the likelihood function

$$\ln L = -\frac{1}{2} \mathbf{s}^T \mathbf{C}^{-1} \mathbf{s} - \frac{1}{2} \text{Indet} \mathbf{C}, \quad (8)$$

where \mathbf{C} is the nondiagonal covariance matrix that incorporates both white and correlated noise. Given fixed noise parameters (i.e., fixed \mathbf{C}), the maximum-likelihood values for the parameter deviations can be written formally as (Coles et al. 2011)

$$\hat{\boldsymbol{\beta}} = (\mathbf{M}^T \mathbf{C}^{-1} \mathbf{M})^{-1} \mathbf{M}^T \mathbf{C}^{-1} \mathbf{r}, \quad (9)$$

along with the parameter covariance matrix

$$\mathbf{K}_\beta = \frac{1}{n - p} (\mathbf{s}^T \mathbf{C}^{-1} \mathbf{s}) (\mathbf{M}^T \mathbf{C}^{-1} \mathbf{M})^{-1}. \quad (10)$$

Once $\hat{\boldsymbol{\beta}}$ are computed, the parameter values can be updated as $\mathbf{b} \rightarrow \mathbf{b} - \hat{\boldsymbol{\beta}}$, similar to the WLS case, and this method is

Table 2
Fitting Methods Available in PINT (in the `pint.fitter` Module)

Fitter	Description	References
<code>PowellFitter</code>	Fitting using the modified Powell algorithm (uses <code>scipy</code>)	Powell (1964)
<code>WLSFitter</code>	WLS fitting	Hobbs et al. (2006)
<code>DownhillWLSFitter</code> §	Downhill WLS fitting Allows fitting for noise parameters	Section 3.3.2
<code>GLSFitter</code>	GLS fitting	Coles et al. (2011)
<code>DownhillGLSFitter</code> §	Downhill GLS fitting Allows fitting for noise parameters	Section 3.3.2
<code>WidebandTOAFitter</code>	GLS fitting for wideband TOAs	Alam et al. 2021
<code>WidebandDownhillFitter</code> §	Downhill GLS fitting for wideband TOAs	Section 3.3.2
<code>MCMCFitter</code> †	MCMC optimization (uses <code>emcee</code>)	Foreman-Mackey et al. (2013)

Note. “†” denotes fitters that have had significant changes since the publication of Luo et al. (2021) and “§” denotes newly implemented fitters. `MCMCFitter` was updated to use the `emcee` package, providing parallel processing capabilities

usually referred to as generalized least-squares (GLS). Note that Equation (9) is evaluated in PINT using a Cholesky decomposition for nondegenerate systems and using an SVD for degenerate systems (see Section 3.3.1 for details).

Since \mathbf{C} is in general not diagonal, computing \mathbf{C}^{-1} is much more expensive than computing \mathbf{N}^{-1} , and scales as $\mathcal{O}(n^3)$ as opposed to $\mathcal{O}(n)$ in the worst-case scenario. Fortunately, \mathbf{C} can be represented in most cases as a rank- m update to the diagonal matrix \mathbf{N} , as (van Haasteren & Vallisneri 2014a)

$$\mathbf{C} = \mathbf{N} + \mathbf{U}\Phi\mathbf{U}^T, \quad (11)$$

where \mathbf{U} is an $n \times m$ -dimensional correlated noise basis matrix and Φ is an $m \times m$ -dimensional diagonal matrix containing the correlated noise weights. In this case, \mathbf{C}^{-1} can be written using the Woodbury identity as

$$\mathbf{C}^{-1} = \mathbf{N}^{-1} - \mathbf{N}^{-1}\mathbf{U}\Sigma^{-1}\mathbf{U}^T\mathbf{N}^{-1}, \quad (12)$$

where

$$\Sigma^{-1} = \Phi^{-1} + \mathbf{U}^T\mathbf{N}^{-1}\mathbf{U}. \quad (13)$$

Additionally, $\det \mathbf{C}$ appearing in the $\ln L$ expression can be computed using the identity

$$\det \mathbf{C} = \det \mathbf{N} \times \det \Sigma \times \det \Phi. \quad (14)$$

This allows us to evaluate $\ln L$, $\hat{\beta}$, and \mathbf{K}_β with $\mathcal{O}(nm^2)$ time complexity, assuming $m \ll n$.

3.3. Failure Modes of Linear Fitting and the Downhill Fitter Algorithm

3.3.1. Handling Parameter Degeneracies

The above-described fitting algorithm, while successful in the vast majority of cases, can nevertheless fail to correctly estimate the maximum-likelihood parameters under certain conditions. The most obvious such scenario is parameter degeneracy, which leads to $\mathbf{M}^T\mathbf{C}^{-1}\mathbf{M}$ being singular. Ideally, this should be addressed by reparameterizing the timing model to avoid the degeneracy. Alternatively, it can be addressed in an ad hoc manner by restricting the fitting algorithm to operate only in a subspace of the parameter space where the fitting problem is nonsingular. This is done by replacing the inverse

$(\mathbf{M}^T\mathbf{C}^{-1}\mathbf{M})^{-1}$ by the pseudoinverse $\mathbf{V}\bar{\mathbf{S}}^{-1}\mathbf{U}^T$, where $\mathbf{M}^T\mathbf{C}^{-1}\mathbf{M} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ is an SVD such that \mathbf{U} and \mathbf{V}^T are orthogonal matrices, \mathbf{S} is a diagonal matrix containing the singular values, and $\bar{\mathbf{S}}^{-1}$ is obtained by replacing the diagonal elements of \mathbf{S}^{-1} that are greater than a certain threshold by 0.

3.3.2. The Downhill Fitter Algorithm

Unfortunately, the fitting algorithm can fail even in the absence of parameter degeneracies under the following conditions: (a) the linear approximation underlying Equation (4) breaks down and the nonlinear terms become important; and (b) the $\mathbf{b} \rightarrow \mathbf{b} - \hat{\beta}$ update brings a parameter outside its physically meaningful range (e.g., orbital eccentricity $e \in [0, 1]$, Shapiro delay shape $\sin \iota \in [0, 1]$), such that the likelihood function becomes ill-defined. PINT implements a robust fitting algorithm, named the Downhill fitter, to deal with such cases, and it is briefly described below.

In the Downhill fitter, the usual $\mathbf{b} \rightarrow \mathbf{b} - \hat{\beta}$ update is replaced by a more general update $\mathbf{b} \rightarrow \mathbf{b} - \lambda\hat{\beta}$, with $\lambda \in (0, 1]$. Even in the problematic cases mentioned above, it may be possible to find a value of λ that leaves the likelihood function both well-defined and improved, even if $\lambda = 1$ does not provide an acceptable solution. In practice, this is done by reducing λ iteratively by some factor, starting from 1, until an acceptable solution is found. If an improved solution cannot be found even after a certain maximum number of iterations of reducing λ , the original solution is retained.

The various fitting methods available in PINT (in the `pint.fitter` module) are listed in Table 2 and the new/updated features are highlighted therein. We have also added a method `pint.fitter.Fitter.auto()` that selects the appropriate type of fitter, depending on the input data (narrowband versus wideband, white noise versus correlated noise), with a preference for the downhill fitter variants.

4. Fitting for Noise Parameters

The fitting methods discussed in Section 3 assume that the TOA covariance matrix \mathbf{C} (or \mathbf{N} in the absence of correlated noise) is fixed. However, the noise characteristics of a given set of TOAs are not usually known a priori and must be

determined from the data themselves. In practice, this is handled by ignoring the noise parameters during the initial data preparation stages, then performing a separate Bayesian noise analysis step on the data once a reasonable (but not necessarily optimal) timing solution is found. The timing model is then refined by applying the estimated noise parameters, and this process is iterated.

In this section, we develop a framework within `PINT` to estimate noise parameters together with timing model parameters, obviating the need for computationally expensive Bayesian noise analysis iterations during pulsar timing. Note that the noise estimates obtained using these methods may still need to be refined *after* the data preparation stage using Bayesian methods. Still, the convergence of such analyses can be accelerated by providing the initial noise estimates as starting points for MCMC samplers.

4.1. Types of Noise

We begin by briefly discussing the different types of noise seen in pulsar timing. These processes can be divided into three broad categories, as described below.

4.1.1. Uncorrelated (White) Noise

Uncorrelated noise or white noise refers to the component of the noise that is independent for each TOA. It may arise from radiometer noise, pulse jitter, RFI, or polarization miscalibration, etc. It is characterized by the diagonal matrix \mathbf{N} , which is populated by the scaled TOA variances ς_i^2 , given by

$$\varsigma_i^2 = F_i^2(\sigma_i^2 + Q_i^2), \quad (15)$$

where

$$F_i = \prod_a f_a^{\mathcal{F}_{ia}}, \quad (16)$$

$$Q_i^2 = \sum_a q_a^2 Q_{ia}. \quad (17)$$

The quantities f_a and q_a are known as EFACs (“error factors”) and EQUADs (“errors added in quadrature”), respectively. \mathcal{F}_{ia} and Q_{ia} represent TOA selection masks, which can be 0 or 1, based on some criterion, which may depend on the observing epoch, observing frequency, or observing system, etc. EFACs and EQUADs are implemented in the `ScaleToaError` component (see Table 1).

4.1.2. Time-uncorrelated Correlated Noise

Time-uncorrelated correlated noise refers to the component of the noise that is correlated among the TOAs derived from the same observation, but that are uncorrelated otherwise. This is usually referred to as ECORR and may arise from pulse jitter, RFI, polarization miscalibration, or interstellar scattering, etc., which are correlated across different frequency sub-bands of the same observation. Since ECORR is uncorrelated across different observations, it is possible to express its contribution to the TOA covariance matrix as a sparse block-diagonal matrix. This allows us to evaluate \mathbf{C}^{-1} using the Sherman–Morrison identity (the rank-1 special case of the Woodbury identity), which turns out to be less expensive than the general case. The TOA covariance matrix contribution arising from

ECORR can be expressed as (Johnson et al. 2024)

$$\mathbf{C}_{\text{ECORR}} = \sum_{ab} c_a^2 \mathbf{v}_{ab} \mathbf{v}_{ab}^T, \quad (18)$$

where c_a represents the ECORR parameter for each selection group, the index b represents the various TOA epochs, and the basis vector \mathbf{v}_{ab} contains “1”s for each TOA belonging to the epoch b and selection group a , and “0”s otherwise. ECORRs are implemented in the `EcorrNoise` component (see Table 1).

4.1.3. Time-correlated Noise

Time-correlated noise in pulsar timing may arise due to the rotational irregularities of the pulsar (spin noise/achromatic red noise or ARN), time-variable interstellar dispersion (DM noise or DMN), time-variable interstellar scattering (scattering noise), or long-timescale RFI (system/band noise), etc. This type of noise can be represented either as a deterministic signal (such as a piecewise-constant function or Fourier expansion, etc.) or as a Gaussian process (GP) that contributes a dense component to the TOA covariance matrix, usually represented using a reduced rank approximation (see Equation (11); van Haasteren & Vallisneri 2014b). The deterministic signal representation provides a delay of the form $\mathbf{U}\varphi$, where \mathbf{U} is a basis matrix and φ is an amplitude vector. For example, the Fourier basis matrix corresponding to a time-correlated noise component is given by

$$U_{ij} = \left(\frac{\nu_{\text{ref}}}{\nu_i} \right)^\alpha \begin{cases} \sin\left(\frac{\pi(j+1)(t_i - t_{\text{ref}})}{T_{\text{span}}} \right) & \text{odd } j \\ \cos\left(\frac{\pi j(t_i - t_{\text{ref}})}{T_{\text{span}}} \right) & \text{even } j, \end{cases} \quad (19)$$

where ν_i is the observing frequency, ν_{ref} is an arbitrary reference frequency (conventionally taken as 1400 MHz), α is known as the chromatic index ($\alpha = 0$ for the ARN and $\alpha = 2$ for the DMN), t_i is the TOA, t_{ref} is an arbitrary reference time, and T_{span} is the total time span of the TOAs. The deterministic representation can be converted into a GP representation by analytically marginalizing the amplitudes φ assuming Gaussian priors (van Haasteren & Vallisneri 2014b). The deterministic Fourier representation of the ARN is implemented as the `WaveX` component and that of the DMN is implemented as the `DMWaveX` component. Their GP counterparts assuming a power-law spectral density are `PLRedNoise` and `PLDMNoise`, respectively (see Table 1).

4.2. Fitting for White-noise Parameters and ECORRs

If the time-correlated noise components are treated using a deterministic representation, as discussed above, the covariance matrix can be written as

$$\mathbf{C} = \mathbf{N} + \sum_{ab} c_a^2 \mathbf{v}_{ab} \mathbf{v}_{ab}^T, \quad (20)$$

which turns out to be block-diagonal. Each block \mathbf{C}_{ab} can be inverted using the Sherman–Morrison identity

(Johnson et al. 2024):

$$\begin{aligned} C_{ab}^{-1} &= (N_{ab} + c_a^2 \mathbf{v}_{ab} \mathbf{v}_{ab}^T)^{-1} \\ &= N_{ab}^{-1} - \frac{c_a^2 N_{ab}^{-1} \mathbf{v}_{ab} \mathbf{v}_{ab}^T N_{ab}^{-1}}{1 + \mathbf{v}_{ab}^T N_{ab}^{-1} \mathbf{v}_{ab}}, \end{aligned} \quad (21)$$

where N_{ab} is the portion of N corresponding to the selection group a and observing epoch b . Similarly, the determinant of each block can be computed using the identity

$$\det \mathbf{C}_{ab} = \det N_{ab} \times (1 + \mathbf{v}_{ab}^T N_{ab}^{-1} \mathbf{v}_{ab}). \quad (22)$$

Defining the inner product $(\mathbf{x}|\mathbf{y}) = \mathbf{x}^T N^{-1} \mathbf{y}$, the log-likelihood can be expressed as $\ln L = \sum_{ab} \ln L_{ab}$, where

$$\begin{aligned} \ln L_{ab} &= -\frac{1}{2} \left\{ (s_{ab}|\mathbf{s}_{ab}) + \ln \det N_{ab} \right. \\ &\quad \left. - \frac{c_a^2 (\mathbf{s}_{ab}|\mathbf{v}_{ab})^2}{1 + (\mathbf{v}_{ab}|\mathbf{v}_{ab})} + \ln [1 + (\mathbf{v}_{ab}|\mathbf{v}_{ab})] \right\}. \end{aligned} \quad (23)$$

We estimate the white-noise and ECORR parameters by numerically maximizing the above expression over the parameters f_a , q_a , and c_a , while fixing the timing model parameters. In practice, we alternate the above maximization procedure with the timing model parameter fitting described in Section 3 several times. The covariance matrix of the noise parameters $\beta_i \in \{f_a, q_a, c_a\}$ can be computed by inverting the Hessian matrix \mathbf{H} of the log-likelihood function, whose elements are $H_{ij} = \frac{\partial^2 \ln L}{\partial \beta_i \partial \beta_j}$ (the Hessian is computed in practice by numerically differentiating $\ln L$). This method is implemented in `DownhillWLSFitter` (without ECORR) and `DownhillGLSFitter` (with ECORR), as mentioned in Table 2.

The pulse phases corresponding to the observed TOAs are computed with respect to a fiducial TOA. Since the choice of this fiducial TOA is arbitrary, there can be an overall phase offset between the measured TOAs and the fiducial TOA. Traditionally, this offset has been taken care of by subtracting a weighted mean from the timing residuals. Unfortunately, this procedure is inadequate when correlated noise (including ECORR) is present, since the weighted mean does not account for the correlated noise. Hence, we treat this offset as a free parameter while fitting for correlated noise parameters (this is implemented in the `PhaseOffset` component; see Table 1).

While the iterative method described in this section effectively maximizes the likelihood function over both timing and noise parameters, the estimated parameter uncertainties may be underestimated, because the timing parameters are kept fixed while estimating noise parameters and vice versa. This effect is usually small, since most timing parameters are determined by the entire data set, while the white noise and ECORR are time-uncorrelated. Nevertheless, this caveat can become important while employing timing parameters whose effect is time-localized, such as epoch-wise DM measurements using `DispersionDMX` (e.g., Tarafdar et al. 2022).

We note here in passing that a white-noise characterization utility is available in `tempo2` via the `fixData` plugin (Hobbs 2014). However, the `PINT` functionality described herein differs from the `fixData` plugin in the following ways: (a) the `fixData` plugin cannot estimate ECORR parameters; (b) it does not provide uncertainties associated with noise parameter estimates; and (c) noise parameter estimation in

`PINT` is integrated into the usual interactive pulsar timing procedures, whereas `fixData` is to be run as a separate step.

4.3. Fitting for Time-correlated Noise Parameters

Recalling the basis given in Equation (19), we can express the deterministic Fourier representation of a time-correlated noise in the following way:

$$\begin{aligned} R(t_i) &= \sum_{j=1}^{N_{\text{harm}}} \left(\frac{\nu_{\text{ref}}}{\nu_i} \right)^\alpha \left\{ a_j \cos \left[\frac{2\pi j(t_i - t_{\text{ref}})}{T_{\text{span}}} \right] \right. \\ &\quad \left. + b_j \sin \left[\frac{2\pi j(t_i - t_{\text{ref}})}{T_{\text{span}}} \right] \right\}. \end{aligned} \quad (24)$$

We estimate the coefficients a_j and b_j by treating them as free parameters and fitting them simultaneously with the timing model parameters. Note that the noise components with frequencies less than T_{span}^{-1} will be absorbed while fitting for the pulsar rotational frequency and its first derivative in the case of the ARN when the spectral index is less than 7. Similarly, such low-frequency components will be absorbed into the first two DM derivatives in the case of the DMN. In the case of steeper spectra ($\gamma > 7$), higher derivatives of rotational frequency and/or DM may be required (Blandford et al. 1984; van Haasteren & Levin 2012; van Haasteren & Vallisneri 2014a; Keith & Nitu 2023).

The power spectrum of a time-correlated noise component is often modeled using a power-law function of the form (Lentati et al. 2014)

$$P(f_j) = \langle a_j^2 \rangle = \langle b_j^2 \rangle = \frac{A^2}{12\pi^2 f_{\text{yr}}^3 T_{\text{span}}} \left(\frac{f_{\text{yr}}}{f_j} \right)^\gamma, \quad (25)$$

where A is the power-law amplitude, γ is the spectral index (not to be confused with the chromatic index α), and $f_{\text{yr}} = 1 \text{ yr}^{-1}$. In our framework, the power-law parameters A and γ can be estimated by maximizing the log-likelihood function:

$$\begin{aligned} \ln \Lambda &= -\frac{1}{2} \sum_{j=1}^{N_{\text{harm}}} \left\{ \frac{\hat{a}_j^2}{P(f_j) + \epsilon_{aj}^2} + \frac{\hat{b}_j^2}{P(f_j) + \epsilon_{bj}^2} \right. \\ &\quad \left. + \ln [P(f_j) + \epsilon_{aj}^2] + \ln [P(f_j) + \epsilon_{bj}^2] \right\}, \end{aligned} \quad (26)$$

where \hat{a}_j and \hat{b}_j are the maximum-likelihood estimates of the Fourier coefficients obtained by fitting Equation (24) to the TOAs as part of the timing model, and ϵ_{aj} and ϵ_{bj} are the measurement uncertainties thereof. Here, we are modeling the measured Fourier amplitudes \hat{a}_j and \hat{b}_j as Gaussian random variables with zero mean whose variance is the sum of the intrinsic variance $P(f_j)$ and the measurement variance ϵ_{aj}^2 or ϵ_{bj}^2 . The uncertainties in the A and γ measurements can be estimated using the Hessian of $\ln \Lambda$, similar to how the EFAC, EQUAD, and ECORR uncertainties are estimated in Section 4.2. We note in passing that the likelihood function given in Equation (26) is analogous to the frequency-domain likelihood given in Laal et al. (2023).

The likelihood function of Equation (26) will, in general, provide different parameter estimation results than the likelihood function of Equation (8) (including the GP

representation of the time-correlated noise), since the latter acts directly on the timing residuals, whereas the former acts on the estimated Fourier coefficients. Additionally, the interpretation of $P(f_j)$ as the variance of the Fourier coefficients a_j and b_j is imposed in Equation (26) after estimating a_j and b_j , whereas it is imposed as a *prior* distribution in the Bayesian analysis involving the GP representation. Nevertheless, we expect their results to be broadly consistent.

These computations are implemented in the `pint.utils.plrednoise_from_wavex()` function for the ARN and in the `pint.utils.pldmnoise_from_dmwavex()` function for the DMN.

Certain caveats regarding the issues that may arise while applying these techniques to real data sets are in order. The astrometric parameters of the pulsar, which determine the solar system Rømer delay with a 1 yr period, are covariant with the Fourier coefficients of the ARN corresponding to the frequency closest to 1 yr^{-1} (and possibly its higher harmonics). Similarly, some of the binary orbital parameters will be covariant with the ARN Fourier coefficients corresponding to the frequency closest to the orbital frequency (and possibly its higher harmonics), if the orbital period is long enough to be between $T_{\text{span}}/N_{\text{harm}}$ and T_{span} . If the solar wind electron density is included in the timing model as a free parameter, it can be covariant with the Fourier coefficients of the DMN corresponding to the frequency closest to 1 yr^{-1} . We recommend excluding such Fourier coefficients while estimating the spectral parameters using Equation (26) to avoid bias. Further, our technique assumes that the Fourier basis given in Equation (19) is orthogonal. Although this assumption is exactly true only in the case of uniformly sampled data sets, it will remain approximately true as long as there are no large data gaps or discrepancies in data quality over time. How to handle cases where this assumption breaks down will be explored in a future work. Finally, there can be large covariances between WaveX and DMWaveX coefficients of the same frequency when the observing bandwidth is low or when only high-frequency observations are available.

A similar utility for estimating the red-noise spectrum is available in `tempo2` via the `spectralModel` plugin (Coles et al. 2011). In terms of usage, it differs from our method in the following ways: (a) `spectralModel` only performs the spectral characterization of the ARN, whereas our method also handles the DMN; (b) in `spectralModel`, the optimal spectral index is estimated by manual iteration, whereas our method estimates it automatically; (c) `spectralModel` uses a model that is a power law at high frequencies but saturates to a constant at low frequencies, whereas our method uses a simple power-law model; and (d) `spectralModel` does not provide uncertainties on the estimated spectral parameters.

5. Model Comparison

An important problem in creating and refining a timing model is determining what configuration of model components produces the optimal fit to the data. In a Bayesian setting, this comparison is accomplished by computing Bayes factors, through techniques such as nested sampling (Ashton et al. 2022) and product-space sampling (Hee et al. 2015). Since `PINT` follows a frequentist maximum-likelihood approach to parameter estimation, these techniques are not suitable to be used within `PINT`, in addition to being too computationally expensive to be used during interactive pulsar timing. Hence,

Table 3
Prior Distributions used for the Bayesian Noise Analysis using ENTERPRISE in Sections 6 and 7

Noise Parameter	Units	Prior Distribution
EFAC		Uniform[0.5, 2.0]
EQUAD	μs	Uniform[0.01, 100]
ECORR	μs	Uniform[0.01, 100]
TNREDAMP		Uniform[−20, −11]
TNREDGAM		Uniform[0, 7]
TNDMAMP		Uniform[−20, −11]
TNDMGAM		Uniform[0, 7]

Note. Note that EFAC, EQUAD, and ECORR are expressed in linear scale, whereas TNREDAMP and TNDMAMP are expressed in log scale in this table. The parameter names used here correspond to how they are represented in “par” files.

we implemented the AIC in `PINT` to perform model comparison. The AIC is defined as

$$\text{AIC} = 2q - 2\ln\hat{L}, \quad (27)$$

where q is the total number of free parameters, including the timing model parameters and the noise model parameters, and \hat{L} is the maximum value of the likelihood for the given model. Given multiple models applied to the same data, the preferred model is the one that minimizes the AIC (with a minimum value AIC_{min}). The i th model can be said to be $\exp[\text{AIC}_{\text{min}} - \text{AIC}_i]$ times as probable as the favored model in minimizing information loss (the quantity $\text{AIC}_{\text{min}} - \text{AIC}_i$ is usually referred to as the AIC difference). See, e.g., Burnham & Anderson (2004) for a detailed description and interpretation of the AIC. The AIC can be computed in `PINT` using the `akaike_information_criterion()` function available in the `pint.utils` module.

The application of the AIC in selecting the applicable noise components and the estimation of noise parameters is demonstrated in the next section. Although these examples only involve noise components, the AIC can be used for other types of comparisons, e.g., between different binary models.

6. Simulation Studies

In this section, we demonstrate the efficacy of our maximum-likelihood noise estimation methods by applying them to simulated data sets and comparing the results with Bayesian estimates obtained using the ENTERPRISE package (see Table 3 for the prior distributions used in the Bayesian analyses). In what follows, we investigate five cases: (a) white-noise only; (b) white noise + ECORR; (c) ARN only; (d) DMN only; and (e) ARN + DMN.

The simulations are performed using the `pint.simulation` module of `PINT` (see Section 8.3). Each simulation corresponds to a fictitious isolated pulsar with a spin frequency of 100 Hz and a spin frequency derivative of -10^{-15} Hz^2 , located at R.A. $05^{\text{h}}00^{\text{m}}00^{\text{s}}$ and decl. $15^{\circ}00'00''$, with a DM of 15 pc cm^{-3} . The solar system delays are estimated using the DE440 solar system ephemeris (Park et al. 2021). Each simulation contains 2000 narrowband TOAs taken at 250 equally spaced epochs spanning an MJD range of 53000–57000 taken at the Green Bank Telescope (GBT). Each epoch contains TOAs from eight equally spaced frequency subbands in the 500–1500 MHz range. The unscaled TOA

Table 4

Noise Parameters Injected into Various Simulations

Simulation #	Noise Parameter	Units	Injected Value
(a)	EFAC[tel gbt]		1.3
	EQUAD[tel gbt]	μs	0.9
(b)	EFAC[tel gbt]		1.3
	ECORR[tel gbt]	μs	0.9
(c)	TNREDAMP		-13
	TNREDGAM		3.5
	TNREDC		30
(d)	TNDMAMP		-13.5
	TNDMGAM		4.0
	TNDMC		30
(e)	TNREDAMP		-13
	TNREDGAM		3.5
	TNREDC		30
	TNDMAMP		-13.5
	TNDMGAM		4.0
	TNDMC		30

Note. The parameter names used here correspond to how they are represented in “par” files. TNREDAMP, TNREDGAM, and TNREDC represent the GP log-amplitude, spectral index, and number of harmonics of the ARN. TNDMAMP, TNDMGAM, and TNDMC represent the GP log-amplitude, spectral index, and number of harmonics of the DMN.

uncertainties (σ_i) are drawn uniformly from the interval 0.5–2.0 μs . Table 4 lists the noise parameters injected into each simulation.

These simulations are intentionally kept simplistic, so that the demonstrations focus on one aspect at a time and remain clear. Please see Section 7 for an application of our method on a real data set.

6.1. Simulation (a): White-noise Only, EFAC and EQUAD

In this simulation, we modify the TOA uncertainties by an EFAC and an EQUAD (see Table 4). We fit the generated TOAs for both timing and noise parameters, starting from an initial model with EFAC = 1 and EQUAD = $10^{-3} \mu\text{s}$ (a small positive value close to 0). To determine which noise parameters are necessary, we explore four versions of this fit, where: (a) the parameters are frozen at EFAC = 1 and EQUAD = 0; (b) EFAC is allowed to vary but EQUAD is frozen at 0; (c) EQUAD is allowed to vary but EFAC is frozen at 1; and (d) both EFAC and EQUAD are allowed to vary. The AIC differences for these configurations are listed in Table 5.

Clearly, both the EFAC and the EQUAD are required to model the white noise in this data set. The comparison of the measured EFAC and EQUAD values with the injected values is shown in Figure 1(b). Figure 1(b) also shows the posterior distribution obtained from a Bayesian analysis performed using ENTERPRISE and PTMCMCSampler (Johnson et al. 2024); see Table 3 for the prior distributions used. We see that the maximum-likelihood EFAC and EQUAD estimates are consistent with the injected values and the Bayesian estimates within error bars.

To ensure that the results are robust, we generated 1000 instances of this simulation and estimated the maximum-likelihood noise parameters for each instance. These measurements, along with the injected values, are shown in Figure 1(c).

Table 5

AIC Differences for Different Noise Model Configurations for Simulation (a) with EFAC and EQUAD (White-noise Only)

Configuration	AIC Difference
Free EFAC, Free EQUAD	0
Free EFAC, EQUAD = 0	44
EFAC = 1, Free EQUAD	198
EFAC = 1, EQUAD = 0	5191

Note. See Section 6.1 for more details.

Table 6

AIC Differences for Different Noise Model Configurations for Simulation (b) with EFAC and ECORR

Configuration	AIC Difference
Free EFAC, Free ECORR	0
Free EFAC, ECORR = 0	228
EFAC = 1, Free ECORR	331
EFAC = 1, ECORR = 0	1238

Note. See Section 6.2 for more details.

Comparing the measured EFAC and EQUAD values with their injected values using Student’s t-test, we find no evidence that the estimates are biased (the p -values are around 0.8 for EFAC and 0.9 for EQUAD). The covariance between EFAC and EQUAD seen in the Bayesian posterior distribution in Figure 1(b) is also seen in Figure 1(c).

6.2. Simulation (b): EFAC and ECORR

In this simulation, we modify the TOA uncertainties by an EFAC and include an ECORR noise component. Similar to the white-noise-only case, we fit the generated TOAs for both timing and noise parameters, starting from an initial model with EFAC = 1 and ECORR = 10^{-3} . We explore four versions of this fit, where each noise parameter is allowed to be free or frozen. The AIC differences for these configurations are listed in Table 6.

Clearly, both the EFAC and the ECORR are required to model the noise in this data set. The comparison of the measured EFAC and ECORR values with the injected values is shown in Figure 2(b), along with the posterior distribution obtained from a Bayesian ENTERPRISE analysis; see Table 3 for the prior distributions used. We see that the maximum-likelihood estimates are consistent with the Bayesian estimates within error bars, whereas both estimates are offset from the injected values.

To check the robustness of our method, we ran 1000 instances of this simulation and estimated the parameters for each. The results of this exercise are shown in Figure 2(c). Comparing the measured values of the EFAC and ECORR with their injected values using Student’s t-test, we find that there is no evidence for bias in the EFAC estimations, whereas the ECORRs are underestimated by around 0.2σ (the p -values are around 0.8 and 3×10^{-11} , respectively). The parameter estimates in Figure 2(b), while more than 1σ away from the injected value, are still within the distribution seen in Figure 2(c), implying that the apparent offset can be attributed to random chance.

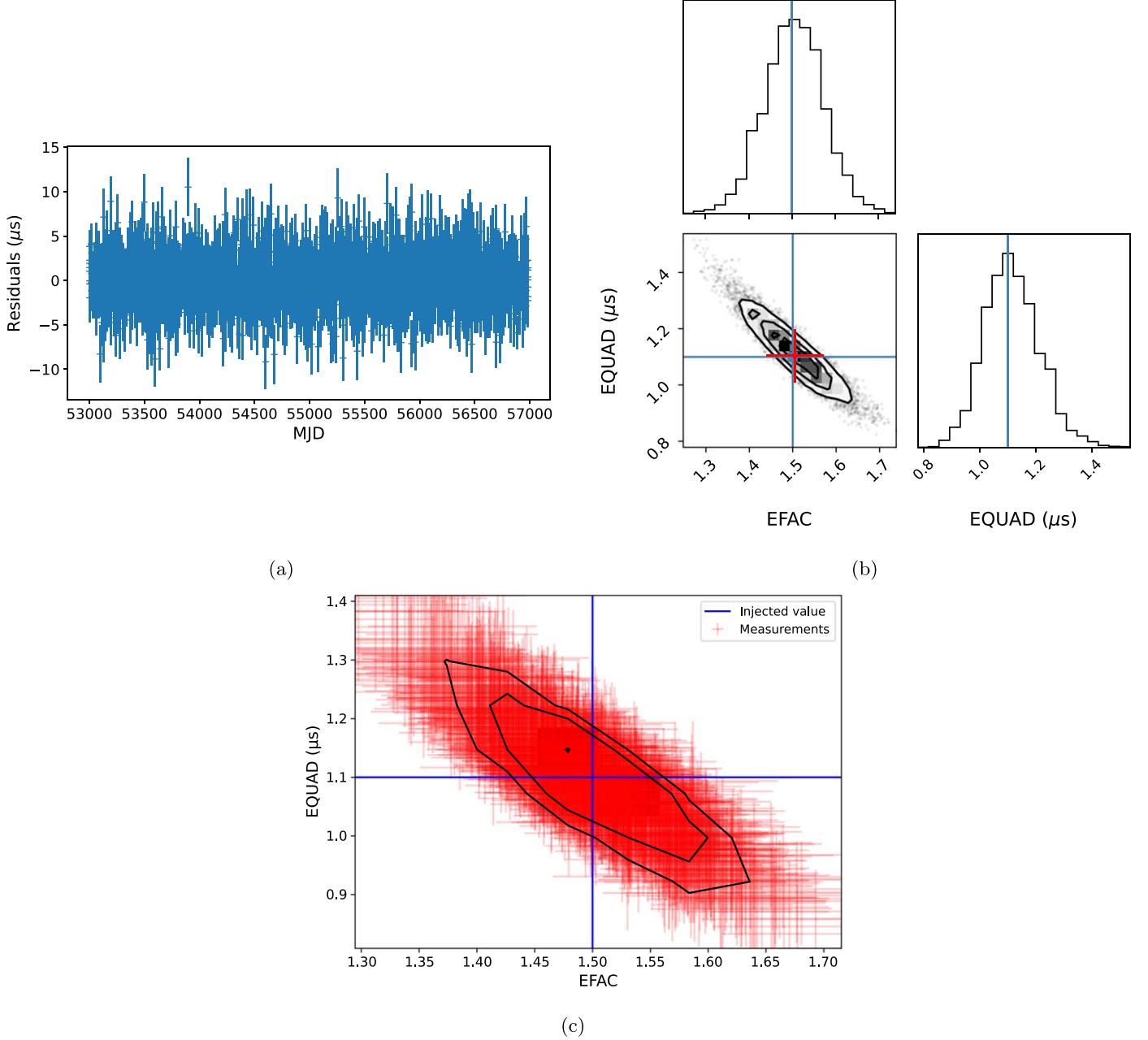


Figure 1. Timing residuals (panel (a)) and parameter estimation results (panel (b)) for the simulation (a) with EFAC + EQUAD (see Section 6.1). In panel (b), the blue lines show the injected values, the red point with error bars shows the maximum-likelihood estimate obtained using PINT, and the corner plot (black) shows the posterior distribution obtained from the ENTERPRISE analysis. Panel (c) shows the results of 500 repetitions of this simulation; the red points with error bars show the maximum-likelihood estimates for each realization, the blue lines indicate the injected values, and the black contours indicate the distribution of the maximum-likelihood values.

6.3. Simulation (c): ARN Only

In this simulation, we inject a realization of the ARN component with a power-law spectrum, modeled as a Fourier GP (implemented in PINT as the PLRedNoise model), into the TOAs. We model this noise using the WaveX model, where the component frequencies are taken to be the harmonics of a fundamental frequency T_{span}^{-1} , where T_{span} is the total observation span. To determine the optimal number of harmonics needed to model the noise, we fit the simulated TOAs using different numbers of harmonics and compute the AIC value corresponding to each case. These AIC values are plotted in Figure 3(a), and the optimum number of harmonics turns out to

be 17 (the injected value is 30; see Table 4). The maximum-likelihood estimates for the Fourier coefficients \hat{a}_i and \hat{b}_i are plotted in the top panel of Figure 3(c), along with the power spectrum in the bottom panel. In Figure 3(c), we see an outlier in the frequency bin containing 1 yr^{-1} . This can be attributed to the covariance of these Fourier coefficients with the astrometric parameters, as discussed in Section 4.3. We fit a power law to the estimated Fourier coefficients, as described in Section 4.3, estimating the power-law amplitude and the spectral index while ignoring the 1 yr^{-1} bin. The best-fit power-law model is also shown in the bottom panel of Figure 3(c), along with its Bayesian counterpart obtained using ENTERPRISE, as well as

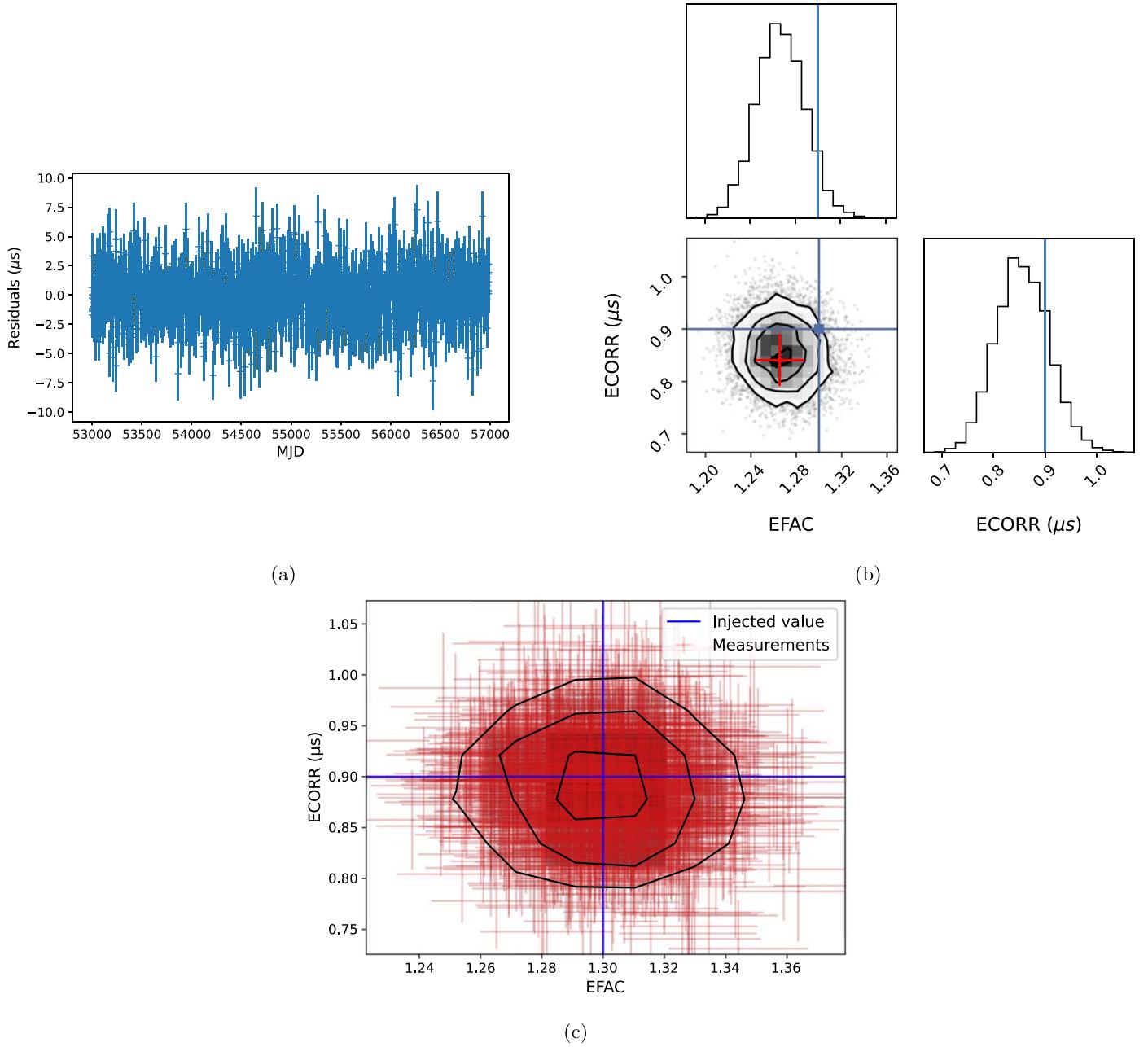


Figure 2. The timing residuals (panel (a)) and the parameter estimation results (panel (b)) for simulation (b) with EFAC and ECORR (see Section 6.2). Panel (c) shows the results of 500 repetitions of this simulation. The plotting conventions are identical to Figure 1.

the injected power-law spectrum for comparison. Further comparisons of these spectral parameter estimates are shown in Figure 3(d), where the posterior distribution obtained from the ENTERPRISE analysis and the injected values are also plotted.

To assess the robustness of our method, we created 1000 instances of this simulation and estimated the parameters for each. The distribution of the measured values is plotted in Figure 3(e). We see that the mean of this distribution is consistent with the injected value (the Student t-test p -values are around 0.6 and 0.7 for the log-amplitude and the spectral index, respectively). However, the distribution has a long tail toward lower amplitudes and steeper spectral indices. Based on Figure 3(e), we see that the offset between the injected and measured values seen in Figure 3(d) is consistent with it being due to random chance.

6.4. Simulation (d): DMN Only

In this simulation, we inject a realization of the DMN with a power-law spectrum, modeled as a Fourier GP (implemented in PINT as the PLDMNoise model), into the TOAs. We model this noise using the DMWaveX model, where the component frequencies are taken to be harmonics of T_{span}^{-1} . We begin by determining the optimal number of harmonics using the AIC, and the AIC differences for different numbers of harmonics are plotted in Figure 4(a). The optimal number of harmonics turns out to be 22 (the injected value is 30). Figure 4(c) shows the estimated Fourier coefficients in the top panel and the estimated spectral powers, the injected power-law spectrum, and the estimated power-law spectra (both using PINT and ENTERPRISE) in the bottom panel. Figure 4(d) shows the corresponding power-law amplitude and spectral index estimates. We

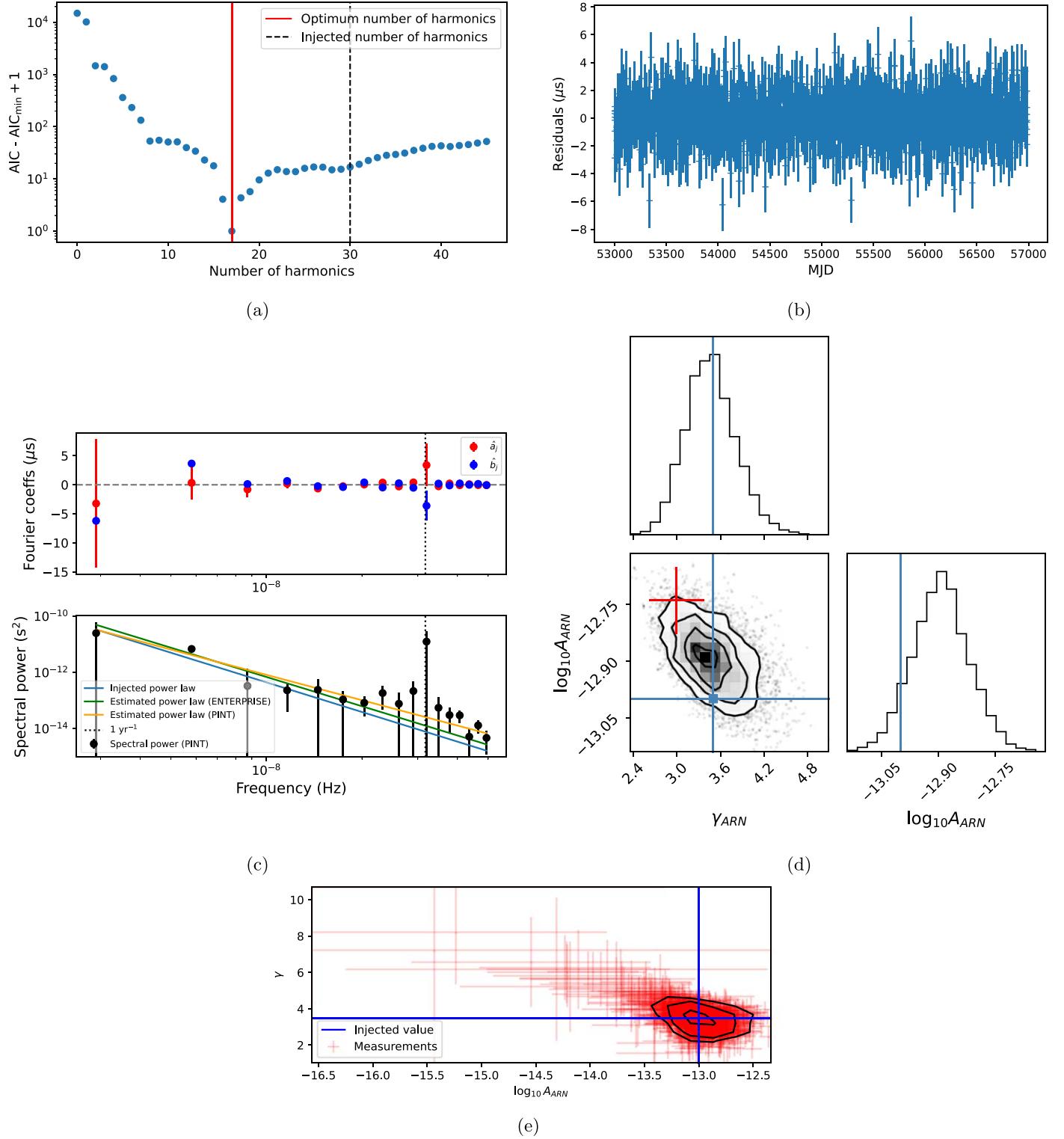


Figure 3. Panel (a) shows the $AIC - AIC_{\min} + 1$ values for the different numbers of harmonics for simulation (c) with only the ARN (see Section 6.3). The red vertical line indicates the optimum number of harmonics. The injected value is indicated by the black dashed vertical line. The Y -axis is plotted in a log scale for better visibility (the $+1$ term is added to $AIC - AIC_{\min}$ to enable this). Panel (b) shows the post-fit timing residuals. Panel (c) shows the parameter estimation results. The top sub-panel shows the maximum-likelihood estimates of the Fourier sine (blue points) and cosine (red points) coefficients \hat{a}_j and \hat{b}_j appearing in Equation (24). The bottom panel shows the corresponding spectral powers (black points) and the power-law fit thereof (orange line) obtained by maximizing the likelihood function given in Equation (26). For comparison, the bottom panel also shows the injected power-law spectrum (blue line) and the power-law spectrum estimated using ENTERPRISE in a Bayesian way (green line). The black dotted line corresponds to 1 yr^{-1} . Panel (d) shows the comparison of the power-law spectral parameters obtained using PINT and ENTERPRISE. The plotting conventions here are identical to Figure 1. Panel (e) shows the results of 1000 repetitions of this simulation, with identical plotting conventions as Figure 1(c).

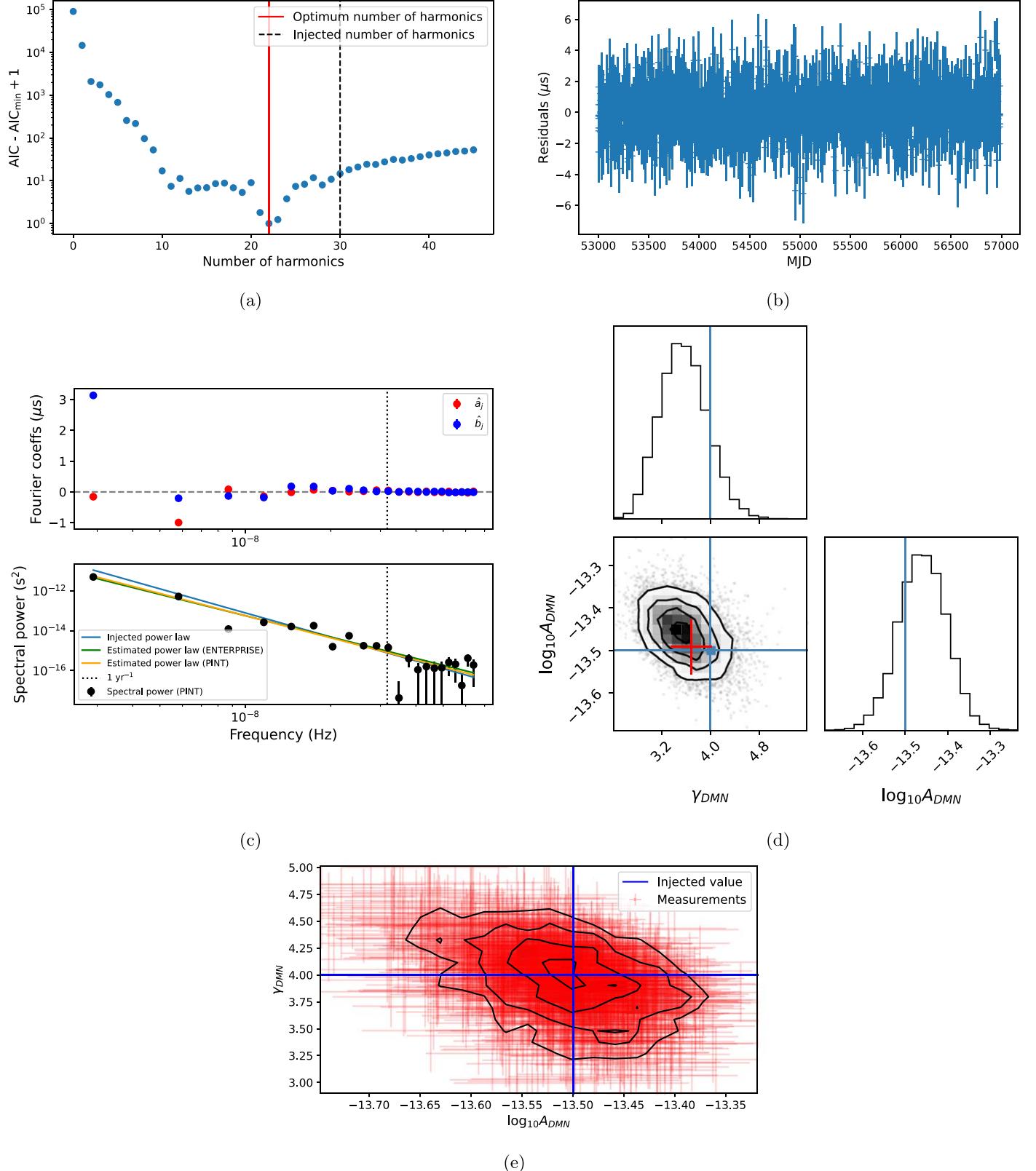


Figure 4. Panel (a) shows the $AIC - AIC_{\min} + 1$ values for the different numbers of harmonics for simulation (d) with only the DMN (see Section 6.4). The plotting conventions are identical to Figure 3(a). Panel (b) shows the post-fit timing residuals. Panel (c) shows the parameter estimation results. The plotting conventions are identical to Figure 3(c). The error bars in the top sub-panel are too small to be visible. Panel (d) shows the comparison of the power-law spectral parameters obtained using PINT and ENTERPRISE. The plotting conventions here are identical to Figure 1. Panel (e) shows the results of 1000 repetitions of this simulation, with identical plotting conventions as Figure 1(c).

see that the power spectra estimated using `PINT` and `enterprise` are consistent with each other as well as with the injected values within the uncertainty levels.

We repeated this simulation 1000 times and measured the noise parameters in each case, and the results of this exercise are plotted in Figure 4(e). The means of the measured log-amplitude and spectral index are both offset from the injected values by around 0.1σ (Student's t-test p -values are approximately 4×10^{-6} and 2×10^{-4} , respectively).

6.5. Simulation (e): ARN and DMN

In this simulation, we inject realizations of the DMN and the ARN with power-law spectra, modeled as Fourier GPs, into the TOAs. We model this noise using the `DMWaveX` and `WaveX` models, where the component frequencies are taken to be harmonics of T_{span}^{-1} . We begin by determining the optimal number of harmonics using the AIC, and the AIC differences for different numbers of harmonics of the DMN and the ARN are plotted in Figure 5(a) using a color map. The optimal numbers of harmonics turn out to be 5 for the DMN and 7 for the ARN (the injected values are 30 and 30, respectively). Figure 4(c) shows the estimated Fourier coefficients in the top panels, and the estimated spectral powers, the injected power-law spectra, and the estimated power-law spectra (both using `PINT` and `ENTERPRISE`) in the bottom panels, for both the DMN (right panels) and the ARN (left panels). Figure 4(d) shows the corresponding estimates for the power-law amplitudes and the spectral indices. We see that the power spectra estimated using `PINT` and `enterprise` are consistent with each other for both the DMN and ARN. However, the ARN spectral parameter estimates are offset from their injected values, although the DMN parameter estimates show good agreement with the injected values.

We repeated this simulation 1000 times and measured the DMN and ARN spectral parameters for each instance. The results of this exercise are plotted in Figure 5(e). We find that the mode of the measured DMN parameter values is close to the injected values (the spectral index measurements are offset by approximately 0.1σ from the injected value, whereas there is no clear evidence that the log-amplitude measurements are offset; the Student t-test p -values are approximately 4×10^{-3} and 0.2, respectively). However, the measured ARN parameter values are more visibly offset from the injected values, namely the spectral index is offset by 0.4σ ($p \sim 10^{-36}$) and the log-amplitude is offset by 0.7σ ($p \sim 10^{-83}$). Further, we find covariance between the measured amplitude and spectral index for the ARN and DMN in Figure 5(e), similar to what is seen in the posterior distribution in Figure 5(d).

These results indicate that the correlated noise spectral parameter estimates can be biased when both the ARN and DMN are present in a data set.

7. Application to PSR B1855+09

We now proceed to demonstrate and test our methods on the NANOGrav 9 yr (NG9) narrowband data set of PSR B1855+09. This data set contains 4005 TOAs taken using the Arecibo telescope during 2004–2013 (Arzoumanian et al. 2015). It was chosen because it is distributed as an example data set together with the `PINT` source code.

We used the “par” file from the NG9 data set as a starting point for our analysis, after removing the DMX parameters

representing piecewise-constant DM variations. We begin our analysis by fitting the timing model parameters, including:

- (a) Astrometric parameters (sky location, proper motion, and parallax);
- (b) DM and its time derivatives (up to the second derivative);
- (c) Pulsar binary parameters for the `BinaryDD` model (binary period and period derivative, projected semimajor axis of the pulsar orbit, eccentricity, argument of periapsis, epoch of periapsis passage, companion mass, and orbital sine inclination);
- (d) Frequency-dependent profile evolution parameters (up to third order);
- (e) Pulsar rotational frequency and its derivative;
- (f) Phase jump between different receivers (*430* and *L-wide*); and
- (g) Overall phase offset.

Thereafter, we included the `DMWaveX` and `WaveX` models with 40 harmonics each to model the DMN and ARN, along with EFAC, EQUAD, and ECORR parameters for each observing system (receiver–backend combinations, denoted as *430_ASP*, *430_PUPPI*, *L-wide_ASP*, and *L-wide_PUPPI*). We have used T_{span}^{-1} as the fundamental frequency of the `DMWaveX` and `WaveX` components.

We then used the AIC to determine which noise parameters are necessary for the given data set, as demonstrated in Sections 6.1 and 6.2, and it turned out that EQUAD parameters were not needed for *430_ASP* and *430_PUPPI*. This was verified using the Savage–Dickey ratio (Dickey 1971) for these parameters, obtained from a Bayesian analysis performed using `ENTERPRISE` and `PTMCMCSampler`. Similarly, the optimal number of `DMWaveX` harmonics was determined to be 10 using the AIC, similar to Section 6.4, and this is shown in Figure 6(a). The optimal number of `WaveX` harmonics turned out to be only 1, with an AIC difference of ~ 16 against a model without `WaveX` but including `DMWaveX` and time-uncorrelated noise parameters, as mentioned above. The sine amplitude a_i associated with this single harmonic turned out to be consistent with zero, and we found high covariances between these parameters and the rotational frequency and frequency derivative. Hence, we excluded `WaveX` from our subsequent analysis.

Finally, we performed a maximum-likelihood fit using the optimal model. The spectral parameters of the DMN were estimated following Section 4.3. The post-fit timing residuals are plotted in Figure 6(b), and the maximum-likelihood DMN parameter estimates are plotted in Figure 6(c). Figure 6(d) shows a comparison between the maximum-likelihood noise parameter estimates and their Bayesian counterparts (see Table 3 for the prior distributions used) using a corner plot. These plots show that the frequentist and Bayesian estimates agree with each other within their respective uncertainties.

8. New Features in `PINT`

In this section, we briefly summarize the new features that have been implemented in `PINT` since the publication of Luo et al. (2021). The new timing model components and new fitting methods were discussed in Sections 2.2 and 3–4, respectively, and are not included here.

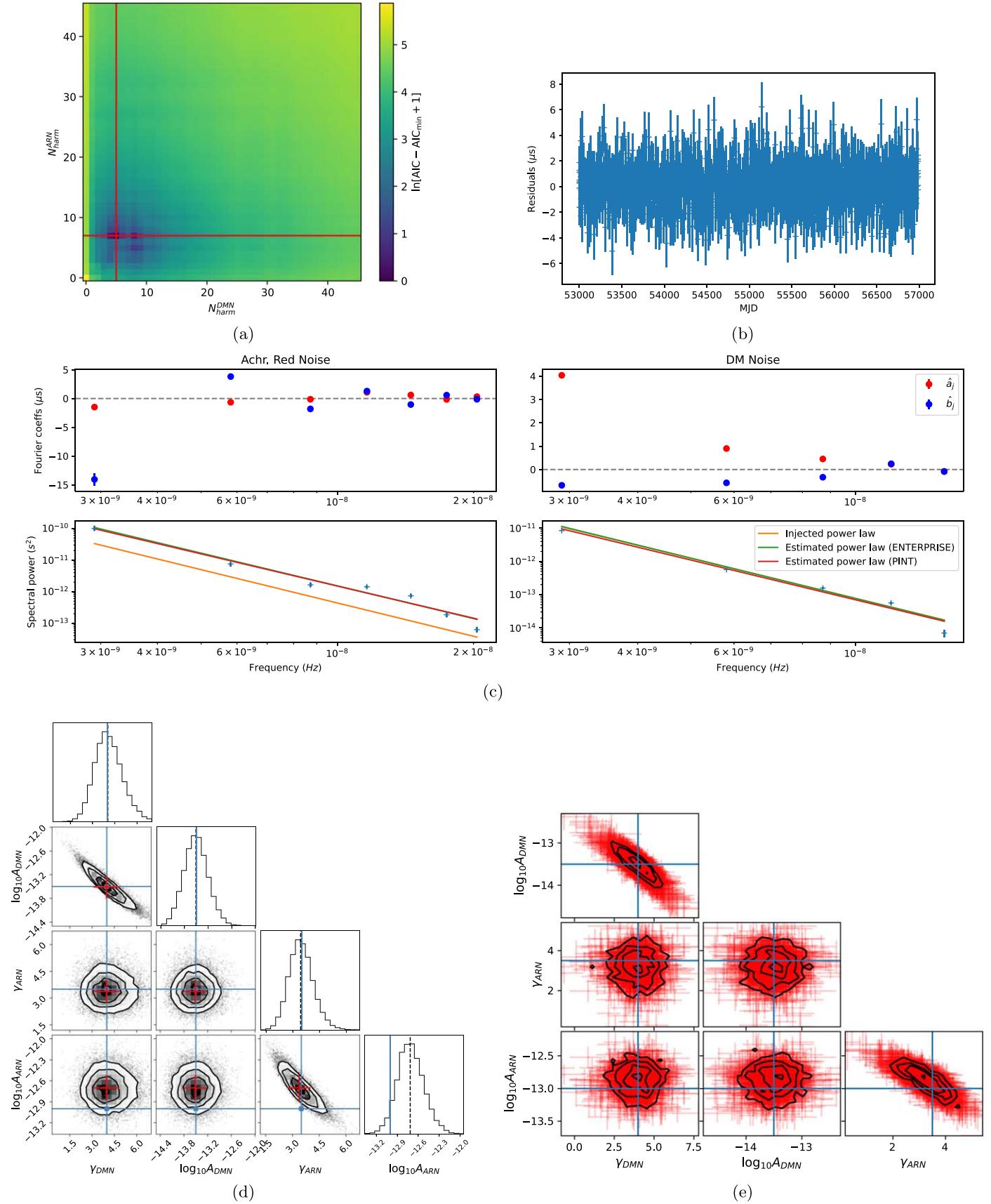


Figure 5. Panel (a) shows the AIC differences in log scale as a function of the number of harmonics of the ARN and DMN. Panel (b) shows the post-fit timing residuals. Panel (c) shows the comparison between the injected and measured power-law spectra. Panel (d) shows the comparison between the injected and measured noise parameters. Panel (e) shows the results of 1000 repetitions of this simulation. The plotting conventions are identical to Figure 3.

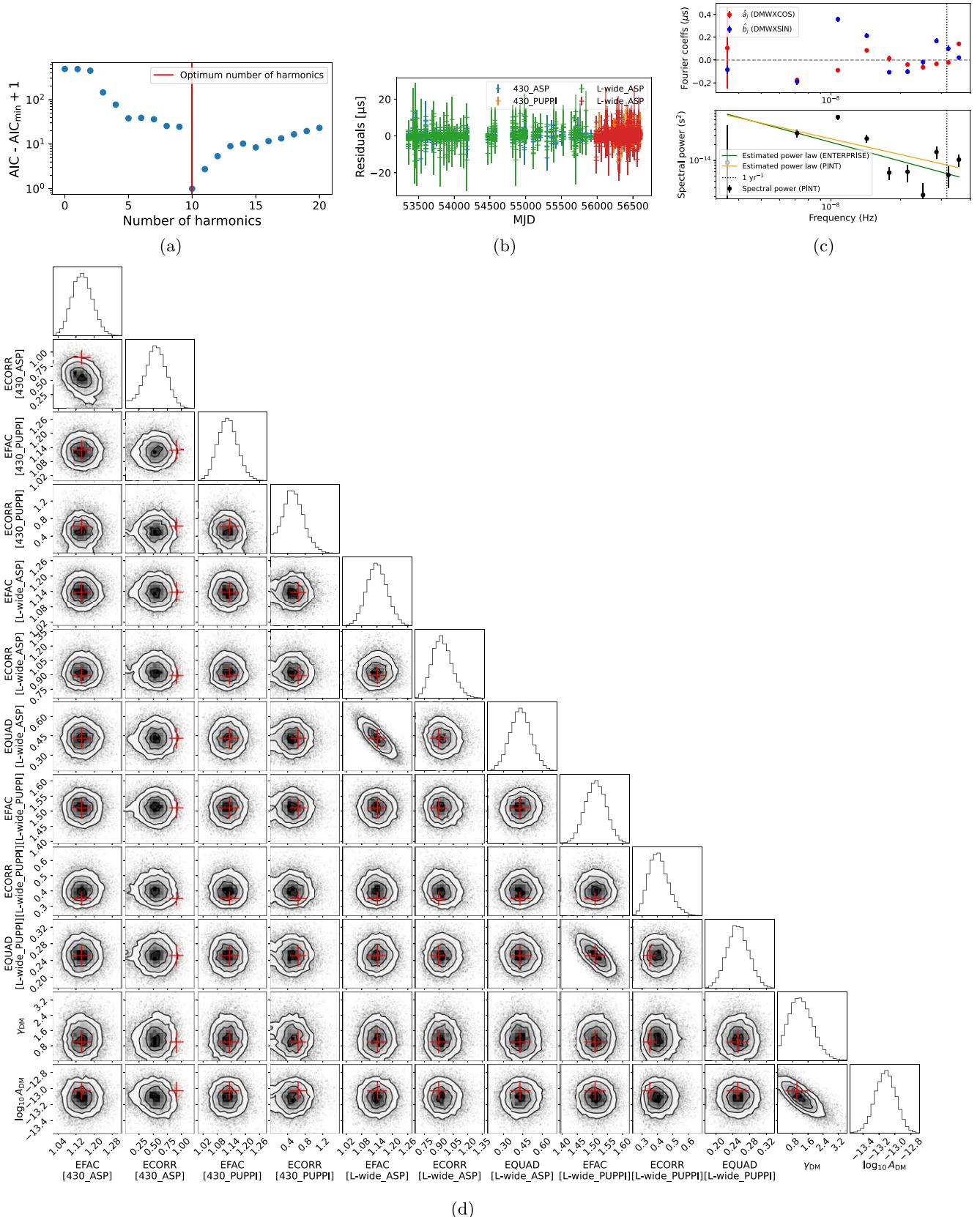


Figure 6. Parameter estimation results for the NG9 data set of PSR B1855+09 (see Section 7). Panel (a) shows the $AIC - AIC_{\min} + 1$ values for the different numbers of DMN harmonics, with identical plotting conventions to Figure 3(a). Panel (b) shows the post-fit timing residuals. The different colors in this panel represent different receiver-backend combinations. Panel (c) shows the parameter estimation results for the DMN, with identical plotting conventions to Figure 3(c). Panel (d) shows the comparison of the noise parameters obtained using PINT and ENTERPRISE, with identical plotting conventions as Figure 1. Note that the EQUAD and ECORR parameters have units of microseconds.

8.1. Timing Model Comparison and Conversion Utilities

Different timing models can be compared with each other using the `TimingModel.compare()` function. This functionality is also available via the `compare_parfiles` command-line utility.

The `convert_binary()` function in the `pint.binaryconvert` module allows the user to convert between different binary models (see Table 1). This functionality can also be accessed via the `convert_parfile` command-line utility. `convert_parfile` also allows the user to convert between different “par” file formats.

PINT only supports the TDB timescale internally, unlike the `tempo2` package, which supports both TDB and TCB timescales (Hobbs et al. 2006). However, PINT can now read “par” files in the TCB timescale and convert them to TDB automatically. This conversion can also be performed using the `tcb2tdb` command-line utility.

8.2. Global Repository for Clock Files

Since the TOAs are usually measured against local observatory clocks, a series of clock corrections must be applied to bring them to a particular realization of the Terrestrial Time timescale and then to the TDB timescale (see Luo et al. 2021 for details). These clock corrections are usually distributed as “clock files,” containing a series of measured differences between the local observatory clock and an international time standard (usually UTC(GPS)) over time. PINT now accesses these files from a central repository²⁷ maintained by the International Pulsar Timing Array consortium (Verbiest et al. 2016). This allows PINT to retain access to the most up-to-date clock corrections without the user having to manually update the clock files.

8.3. TOA Simulations

PINT’s simulation functionality is implemented in the `pint.simulation` module, and can be invoked using the `zima` command-line utility. This module now provides a wide range of functionality on top of simple TOA generation, including the simulation of wideband TOAs, the simulation of white noise incorporating EFACs and EQUADs, and the simulation of different types of single-pulsar correlated noise, including ECORR, ARN, and DMN. Note that this module can only simulate single-pulsar signals, and it cannot simulate signals that are correlated across pulsars, such as the gravitational-wave background.

8.4. Bayesian Interface

The `BayesianTiming` class in the `pint.bayesian` module can be used to perform Bayesian parameter estimation and model selection for pulsar timing data sets. This interface can be used to evaluate the likelihood function, prior distribution, and prior transform function, and is compatible with both MCMC and nested samplers. This interface supports Bayesian inference on both narrowband and wideband data sets and allows the user to estimate the timing model and white-noise parameters. However, estimating correlated noise parameters (in their GP representation) is not yet supported.

²⁷ <https://github.com/ipta/pulsar-clock-corrections>

8.5. χ^2 Grids

In some cases, rather than determine the best-fit values of all parameters, it is desirable to fit for all but a few parameters while stepping over a grid of a subset of the parameters. This is commonly done with post-Keplerian binary parameters (Damour & Deruelle 1986) that can constrain the pulsar and companion masses (e.g., Cromartie et al. 2020; Fonseca et al. 2021), since these parameters are often only marginally significant and give overlapping constraints on the masses.

To facilitate this, we have implemented within the `pint.gridutils` module several methods. These enable the χ^2 to be computed over grids of intrinsic parameters (such as Shapiro delay companion mass and range) or derived parameters created from combinations of intrinsic parameters (such as characteristic age). The individual model fits are done with the help of the `concurrent.futures` module (de Groot 2020), which enables the asynchronous launching and tracking of multiple jobs and has extensions like `clusterfutures`²⁸ that enable deployment on high-performance computing clusters with batch scheduling using `slurm` (Yoo et al. 2003).

8.6. Publication Output

The `publish()` function in the `pint.output.publish` module can be used to generate a LaTEX table summary of a data set comprising a timing model and a set of TOAs. The same functionality is also available through the `pintpublish` command-line utility.

9. Summary and Discussion

In this paper, we describe the new developments in the PINT pulsar timing package, with a focus on frequentist parameter estimation methods. This includes the newly implemented timing model components (Table 1), fitting algorithms (Table 2), and features for performing specific tasks, such as simulation and Bayesian inference, etc. (Section 8). In particular, we described the Downhill fitter algorithm, an improved version of the linear fitting algorithm commonly used in pulsar timing that is robust against significant nonlinearity in the likelihood function as well as regions of the parameter space where the likelihood function is ill-defined.

We presented a new framework within the PINT pulsar timing package to perform frequentist pulsar timing noise characterization, involving the maximum-likelihood estimation of both uncorrelated and correlated noise parameters together with timing model parameters, as well as a model comparison functionality using the AIC. We demonstrated our parameter estimation and model comparison methods using simulated data sets as well as the NG9 data set of PSR B1855+09. The results obtained from these exercises show good agreement between our methods and conventional Bayesian methods, indicating the reliability of our methods. However, we find small parameter estimation biases (within 1σ , but still statistically significant) in our simulations. In future work, we plan to explore these biases and the various covariances that may arise due to data gaps and low observing bandwidth, etc.

Additionally, we also present other new features of PINT, such as new timing model components, timing model comparison and conversion utilities, a global repository for

²⁸ <https://github.com/sampsyo/clusterfutures>

clock files, TOA simulation, an interface for Bayesian analysis, χ^2 grids, and a publication output utility.

The new noise characterization framework should improve the task of pulsar timing in two ways. In pulsar timing projects where Bayesian analysis is deemed not worth the cost, noise characterization is either not done at all or is only done in an ad hoc manner. In such projects, our new framework can provide a convenient and inexpensive alternative for noise characterization, ensuring more robust timing solutions. In high-precision applications where Bayesian noise characterization is necessary, our framework can help accelerate the data preparation/combination stage, by integrating noise characterization into interactive pulsar timing workflows and pulsar timing pipelines without needing a Bayesian analysis step. The quick parameter estimates and model comparisons provided by our framework can quicken the iterative refinement of noise models. Further, the frequentist noise parameter estimates can act as independent cross-checks for Bayesian results and as starting points for MCMC samplers to help them converge faster.

Acknowledgments

This work has been carried out as part of the NANOGrav collaboration, which receives support from the National Science Foundation (NSF) Physics Frontiers Center award numbers 1430284 and 2020265. Portions of this work performed at NRL were supported by ONR 6.1 basic research funding. T.T.P. acknowledges support from the Extragalactic Astrophysics Research Group at Eötvös Loránd University, funded by the Eötvös Loránd Research Network (ELKH), which was used during the development of this research. S.M. R. is a CIFAR Fellow. M.B. was supported in part by PRIN TEC INAF 2019 “SpecTemPolar!—Timing analysis in the era of high-throughput photon detectors” and CICLOPS—Citizen Computing Pulsar Search, a project supported by *POR FESR Sardegna 2014–2020 Asse 1 Azione 1.1.3* (code RICER-CA_1C-181), call for proposal “Aiuti per Progetti di Ricerca e Sviluppo 2017,” managed by Sardegna Ricerche. H.T.C. acknowledges funding from the U.S. Naval Research Laboratory. A.S. thanks David Nice and Bhal Chandra Joshi for helpful discussions.

Data Availability

The NANOGrav 9 yr data set of PSR B1855+09 is available at <https://nanograv.org/science/data/9-year-pulsar-timing-array-data-release>. It is also distributed as an example data set along with the PINT source code at <https://github.com/nanograv/PINT>. The simulated data sets used in this paper and the Jupyter notebooks used to create and analyze them are available at <https://github.com/abhiskckl/pint-noise>.

Software: numpy (Harris et al. 2020), scipy (Virtanen et al. 2020), astropy (Price-Whelan et al. 2022), matplotlib (Hunter 2007), corner (Foreman-Mackey 2016), numdifftools,²⁸ PINT (Luo et al. 2021), ENTERPRISE (Johnson et al. 2024), enterprise_extensions (Johnson et al. 2024), PTMCMCSampler (Johnson et al. 2024), emcee (Foreman-Mackey et al. 2013), Jupyter.²⁹

²⁸ <https://github.com/pbrod/numdifftools>

²⁹ <https://jupyter.org/>

Appendix System and Frequency-dependent Delays

The phenomenological model used to account for frequency profile evolution is given by

$$\Delta_{\text{FD}} = \sum_p \mathfrak{F}_p \left[\log \left(\frac{\nu}{1 \text{ GHz}} \right) \right]^p, \quad (\text{A1})$$

where p is an index and \mathfrak{F}_p is known as a frequency-dependent (FD) parameter. In large data sets containing TOAs obtained using many different observing systems, it is often insufficient to model the frequency-dependent profile evolution using global FD parameters that affect all TOAs. The need for system-dependent FD parameters can arise due to: (a) the data reduction procedures used for different observing systems being different; and (b) different template profiles being used to compute TOAs for different systems, etc. The system-dependent FD delay is given by

$$\Delta_{\text{FDJump}} = \sum_{pq} \mathfrak{F}_{pq} \left[\log \left(\frac{\nu}{1 \text{ GHz}} \right) \right]^p \mathcal{M}_q, \quad (\text{A2})$$

where \mathfrak{F}_{pq} is known as a system-dependent FD parameter (also known as an FD jump), and \mathcal{M}_q is a TOA selection mask. An alternative model for Δ_{FDJump} , originally implemented in tempo2, is also available:

$$\Delta_{\text{FDJump}} = \sum_{pq} \mathfrak{F}_{pq} \left(\frac{\nu}{1 \text{ GHz}} \right)^p \mathcal{M}_q. \quad (\text{A3})$$

The two expressions above can be toggled using a Boolean parameter FDJUMPLOG.

ORCID iDs

- Abhimanyu Susobhanan  <https://orcid.org/0000-0002-2820-0931>
 David L. Kaplan  <https://orcid.org/0000-0001-6295-2881>
 Anne M. Archibald  <https://orcid.org/0000-0003-0638-3340>
 Jing Luo  <https://orcid.org/0000-0001-5373-5914>
 Paul S. Ray  <https://orcid.org/0000-0002-5297-5278>
 Timothy T. Pennucci  <https://orcid.org/0000-0001-5465-2889>
 Scott M. Ransom  <https://orcid.org/0000-0001-5799-9714>
 Gabriella Agazie  <https://orcid.org/0000-0001-5134-3925>
 William Fiore  <https://orcid.org/0000-0001-5645-5336>
 Bjorn Larsen  <https://orcid.org/0000-0001-6436-8216>
 Patrick O’Neill  <https://orcid.org/0009-0009-2567-1533>
 Rutger van Haasteren  <https://orcid.org/0000-0002-6428-2620>
 Akash Anumarlapudi  <https://orcid.org/0000-0002-8935-9882>
 Matteo Bachetti  <https://orcid.org/0000-0002-4576-9337>
 Deven Bhakta  <https://orcid.org/0000-0002-7965-3076>
 Chloe A. Champagne  <https://orcid.org/0000-0001-5438-540X>
 H. Thankful Cromartie  <https://orcid.org/0000-0002-6039-692X>
 Paul B. Demorest  <https://orcid.org/0000-0002-6664-965X>
 Ross J. Jennings  <https://orcid.org/0000-0003-1082-2342>
 Matthew Kerr  <https://orcid.org/0000-0002-0893-4073>
 Sasha Levina  <https://orcid.org/0000-0003-1241-7615>
 Alexander McEwen  <https://orcid.org/0000-0001-5481-7559>
 Brent J. Shapiro-Albert  <https://orcid.org/0000-0002-7283-1124>
 Joseph K. Swiggum <https://orcid.org/0000-0002-1075-3837>

References

- Agazie, G., Antoniadis, J., Anumarlapudi, A., et al. 2024a, *ApJ*, **966**, 105
- Agazie, G., Anumarlapudi, A., Archibald, A., et al. 2024b, *nanograv/PINT*, v1.0.0, Zenodo, doi:[10.5281/zenodo.11396241](https://doi.org/10.5281/zenodo.11396241)
- Agazie, G., Anumarlapudi, A., Archibald, A. M., et al. 2023a, *ApJL*, **951**, L10
- Agazie, G., Anumarlapudi, A., Archibald, A. M., et al. 2023b, *ApJL*, **952**, L37
- Alam, M. F., Arzoumanian, Z., Baker, P. T., et al. 2021, *ApJS*, **252**, 5
- Antoniadis, J., Arumugam, P., Arumugam, S., et al. 2023, *A&A*, **678**, A50
- Arzoumanian, Z., Brazier, A., Burke-Spolaor, S., et al. 2015, *ApJ*, **813**, 65
- Ashton, G., Bernstein, N., Buchner, J., et al. 2022, *NRvMP*, **2**, 39
- Backer, D. C., & Hellings, R. W. 1986, *ARA&A*, **24**, 537
- Blandford, R., Narayan, R., & Roman, R. W. 1984, *JApA*, **5**, 369
- Blandford, R., & Teukolsky, S. A. 1976, *ApJ*, **205**, 580
- Burnham, K. P., & Anderson, D. R. 2004, *Sociol. Method Res.*, **33**, 261
- Caballero, R. N., Guo, Y. J., Lee, K. J., et al. 2018, *MNRAS*, **481**, 5501
- Coles, W., Hobbs, G., Champion, D. J., Manchester, R. N., & Verbiest, J. P. W. 2011, *MNRAS*, **418**, 561
- Cromartie, H. T., Fonseca, E., Ransom, S. M., et al. 2020, *NatAs*, **4**, 72
- Damour, T., & Deruelle, N. 1986, *AIHPA*, **44**, 263
- Damour, T., & Taylor, J. H. 1992, *PhRvD*, **45**, 1840
- Davis, J. L., Herring, T. A., Shapiro, I. I., Rogers, A. E. E., & Elgered, G. 1985, *RaSc*, **20**, 1593
- de Groot, C. 2020, The Concurrent Futures Library (Berlin: Springer), 12
- Demorest, P. B., Ferdman, R. D., Gonzalez, M. E., et al. 2012, *ApJ*, **762**, 94
- Deng, X. P., Coles, W. A., Hobbs, G. B., et al. 2012, *MNRAS*, **424**, 244
- Dickey, J. M. 1971, *Ann. Math. Statist.*, **42**, 204
- Donner, J. Y., Verbiest, J. P. W., Tiburzi, C., et al. 2020, *A&A*, **644**, A153
- Edwards, R. T., Hobbs, G. B., & Manchester, R. N. 2006, *MNRAS*, **372**, 1549
- Fiore, W., Levin, L., McLaughlin, M. A., et al. 2023, *ApJ*, **956**, 40
- Fonseca, E., Cromartie, H. T., Pennucci, T. T., et al. 2021, *ApJL*, **915**, L12
- Foreman-Mackey, D. 2016, *JOSS*, **1**, 24
- Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J. 2013, *PASP*, **125**, 306
- Foster, R. S., & Backer, D. C. 1990, *ApJ*, **361**, 300
- Freire, P. C. C., & Wex, N. 2010, *MNRAS*, **409**, 199
- Harris, C. R., Millman, K. J., van der Walt, S. J., et al. 2020, *Natur*, **585**, 357
- Hazboun, J. S., Simon, J., Madison, D. R., et al. 2022, *ApJ*, **929**, 39
- Hee, S., Handley, W. J., Hobson, M. P., & Lasenby, A. N. 2015, *MNRAS*, **455**, 2461
- Hobbs, G. 2014, TEMPO2 Examples, https://www.jb.man.ac.uk/~pulsar/Resources/tempo2_examples_ver1.pdf
- Hobbs, G., Guo, L., Caballero, R. N., et al. 2019, *MNRAS*, **491**, 5951
- Hobbs, G. B., Edwards, R. T., & Manchester, R. N. 2006, *MNRAS*, **369**, 655
- Hotan, A. W., van Straten, W., & Manchester, R. N. 2004, *PASA*, **21**, 302
- Hunter, J. D. 2007, *CSE*, **9**, 90
- Johnson, A. D., Meyers, P. M., Baker, P. T., et al. 2024, *PhRvD*, **109**, 103012
- Jones, G. L., & Qin, Q. 2022, *AnRSA*, **9**, 557
- Keith, M. J., & Nitu, I. C. 2023, *MNRAS*, **523**, 4603
- Kopeikin, S. M. 1995, *ApJL*, **439**, L5
- Kopeikin, S. M. 1996, *ApJL*, **467**, L93
- Kramer, M., Stairs, I. H., Manchester, R. N., et al. 2006, *Sci*, **314**, 97
- Kramer, M., Stairs, I. H., Manchester, R. N., et al. 2021, *PhRvX*, **11**, 041050
- Laal, N., Lamb, W. G., Romano, J. D., et al. 2023, *PhRvD*, **108**, 063008
- Lange, C., Camilo, F., Wex, N., et al. 2001, *MNRAS*, **326**, 274
- Lentati, L., Alexander, P., Hobson, M. P., et al. 2014, *MNRAS*, **437**, 3004
- Lorimer, D. R., & Kramer, M. 2012, Handbook of Pulsar Astronomy (Cambridge: Cambridge Univ. Press)
- Luo, J., Ransom, S., Demorest, P., et al. 2021, *ApJ*, **911**, 45
- Madison, D. R., Cordes, J. M., Arzoumanian, Z., et al. 2019, *ApJ*, **872**, 150
- Manchester, R. N. 2017, *JApA*, **38**, 42
- Nice, D., Demorest, P., Stairs, I., et al. 2015 Tempo: Pulsar Timing Data Analysis, Astrophysics Source Code Library, ascl:[10.5281/zenodo.1509.002](https://doi.org/10.5281/zenodo.1509.002)
- Niell, A. E. 1996, *JGR*, **101**, 3227
- Park, R. S., Folkner, W. M., Williams, J. G., & Boggs, D. H. 2021, *AJ*, **161**, 105
- Pennucci, T. T. 2019, *ApJ*, **871**, 34
- Pennucci, T. T., Demorest, P. B., & Ransom, S. M. 2014, *ApJ*, **790**, 93
- Powell, M. J. D. 1964, *CompJ*, **7**, 155
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 1992, Numerical Recipes in C: The Art of Scientific Computing (2nd ed.; Cambridge: Cambridge Univ. Press), 32
- Price-Whelan, A. M., Lim, P. L., Earl, N., et al. 2022, *ApJ*, **935**, 167
- Rafikov, R. R., & Lai, D. 2006, *PhRvD*, **73**, 063003
- Ransom, S. M. 2001, PhD thesis, Harvard University, Massachusetts
- Ray, P. S., Guillot, S., Ho, W. C. G., et al. 2019, *ApJ*, **879**, 130
- Reardon, D. J., Zic, A., Shannon, R. M., et al. 2023, *ApJL*, **951**, L6
- Sazhin, M. V. 1978, *SvA*, **22**, 36
- Shapiro, I. I. 1964, *PhRvL*, **13**, 789
- Susobhanan, A., Gopakumar, A., Joshi, B. C., & Kumar, R. 2018, *MNRAS*, **480**, 5260
- Tarafdar, P., Nobleson, K., Rana, P., et al. 2022, *PASA*, **39**, e053
- Taylor, J. H. 1992, *RSPTA*, **341**, 117
- Taylor, J. H., & Weisberg, J. M. 1989, *ApJ*, **345**, 434
- Tiburzi, C., Shaifullah, G. M., Bassa, C. G., et al. 2021, *A&A*, **647**, A84
- Vallisneri, M. 2020 libstempo: Python Wrapper for Tempo2, Astrophysics Source Code Library, ascl:[2002.017](https://doi.org/10.5281/zenodo.2002.017)
- Vallisneri, M., Taylor, S. R., Simon, J., et al. 2020, *ApJ*, **893**, 112
- van Haasteren, R., & Levin, Y. 2012, *MNRAS*, **428**, 1147
- van Haasteren, R., & Vallisneri, M. 2014a, *MNRAS*, **446**, 1170
- van Haasteren, R., & Vallisneri, M. 2014b, *PhRvD*, **90**, 104012
- van Straten, W., & Bailes, M. 2011, *PASA*, **28**, 1
- Verbiest, J. P. W., Lentati, L., Hobbs, G., et al. 2016, *MNRAS*, **458**, 1267
- Virtanen, P., Gommers, R., Oliphant, T. E., et al. 2020, *NatMe*, **17**, 261
- Weisberg, J. M., & Huang, Y. 2016, *ApJ*, **829**, 55
- Wolszczan, A., & Frail, D. A. 1992, *Natur*, **355**, 145
- Xu, H., Chen, S., Guo, Y., et al. 2023, *RAA*, **23**, 075024
- Yoo, A. B., Jette, M. A., & Grondona, M. 2003, in Job Scheduling Strategies for Parallel Processing, ed. D. Feitelson, L. Rudolph, & U. Schwiegelshohn (Berlin: Springer), 44
- You, X. P., Coles, W. A., Hobbs, G. B., & Manchester, R. N. 2012, *MNRAS*, **422**, 1160
- You, X. P., Hobbs, G. B., Coles, W. A., Manchester, R. N., & Han, J. L. 2007, *ApJ*, **671**, 907
- Zhu, W. W., Desvignes, G., Wex, N., et al. 2018, *MNRAS*, **482**, 3249