

# Tempo2 Tutorial - IPTA 2023

Ryan Shannon & Aditya Parthasarathy

September 26, 2023

## Abstract

In this two-part tutorial, we will first use PSRCHIVE to produce arrival times and then use the TEMPO2 software to analyse timing residuals of the MeerKAT Pulsar MSP J1903–7051. TEMPO2 is a high-precision pulsar timing software and provides a generic set of tools to analyse and visualise pulsar timing residuals. For background reading, please refer to ?. The software repository is hosted here:

<https://bitbucket.org/psrsoft/tempo2/src/master/>

The data can for the tutorial can be found here:

<https://cloudstor.aarnet.edu.au/plus/s/TQTH4uS5JQ5XTpb>

The data can be downloaded here:

```
curl -O "ftp://elwood.ru.ac.za/pub/geyer/NWU_pulsartiming/data/Session2_timing.tgz"
```

## 1 Introduction

Pulsars spin with remarkable rotational stability, which allows us to use them as highly accurate clocks, distributed across the galaxy. Pulsar timing is a powerful technique which has enabled tests of gravity, plasma physics and nuclear physics – extreme conditions that are not achievable in Earth-based laboratories. It can also be feasible to use an array of ultra-stable millisecond pulsars to detect nanohertz-frequency gravitational waves.

We have learned the basics of pulsar data sets. In the first part, we will be forming arrival times analysing pulsar timing residuals. Pulsar timing residuals show us how well we are able to predict the arrival times of a pulsar’s pulses. We will use data on the PSR J1903–7051 obtained with the MeerKAT radio telescope, as part of the MeerKAT Pulsar Array.

## 2 Estimation of arrival times

### 2.1 Inspect the profiles

Arrival times are made by cross correlating a template (standard) with individual observations. Before forming the arrival times it is useful to look at the data. This can be accomplished by using the tools discussed in previous activities. We will start by looking at fully frequency averaged together. Before we start forming TOAs it is useful to get a general sense of the quality of data and assess what we might expect to achieve in terms of timing precision.

The achievable timing precision is directly proportional to the pulse width and inversely proportional to the observation signal to noise ratio:

$$\sigma_{\text{TOA}} = \frac{W}{\text{S/N}} \quad (1)$$

In practice we need to use a measurement of pulse width that accounts for the sharpness of a pulse profile.

What is the range in signal to noise ratios in the observations? What is the median value?

What is the pulse width for the pulsar?

What is expected median timing precision?

## 2.2 Add data together

The first step is to create a template. Before we do that we need to add a set of observations together. This can be accomplished using the *psradd* tool in psrchive.

```
psradd -o add.ar <cleaned_archives>
```

(Adding all of the archives together might take some time - you may wish to only consider adding together a subset of observations, for example those with high signal to noise ratio).

## 2.3 Generating template using *psrsmooth*

Before creating the arrival time estimates, we can use *psrsmooth* to smooth the template.

Typically the templates are generated to be noise-free.

```
psrsmooth -W add.ar
```

This generates a new file (with *.sm* extension) after using a wavelet smoothing routine on the input template profile.

## 2.4 Generating template using *paas*

An alternative way to generate a template is interactively through the *paas* utility. This graphical tool allows you to fit a set of von Mises (Gaussian) components to an average pulse profile.

```
paas -i -d 1/xs add.ar
```

The tool will allow you to iteratively allow you to fit a model for the pulse profile. After you are satisfied with the model type  $s$ , to save and then  $q$  to quite and you should have a new standard  $paas.std$  that can also be used when forming arrival times.

## 2.5 Generating TOAs using *pat*

Once we have a smoothed template and cleaned profiles, we can use *pat* to generate pulse arrival times.

```
pat -jFTp -A FDM -s <standard> -f 'tempo2' <cleaned_archives>
```

The above command produces arrival times after frequency, time and polarization integration (*-jFTp*). It uses a Fourier-domain Markov chain Monte Carlo algorithm (*-A FDM*) to compute the ToAs. The *-s* flag is used to specify the standard and *-f*, the output format.

Redirect the output of this command to a text file:

```
pat -jFTp -A FDM -s <standard> -f 'tempo2' <cleaned_archives> > J1903-7051.tim
```

We will be using this to generate the timing residuals. The columns in the arrival time show information used by *tempo2* and *enterprise* used in pulsar timing analysis.

1. A string of characters (usually the file name used to produce the TOA)
2. The centre frequency of the arrival time
3. The arrival time (measured in modified julian days; MJD)
4. The arrival time certainty (measured in  $\mu\text{s}$ ).
5. The observatory site

What is the median arrival time uncertainty? How does it compare to the prediction from Section 2.1?

You can also use the *pat* utility to see how well the standard matches the observation. To do this we use the *-t* option and the *-K* option to

```
pat -K 1/xs -t -A FDM -s <standard> -f 'tempo2' <cleaned_archives>
```

How well does the standard match the observations?

What happens to the arrival time uncertainties if you use a bad model for the data

### 3 Pulsar ephemerides

We previously learned that a time of arrival (TOA) is computed by comparing the observed pulse profile with a noise-free reference template. This process is called ‘template matching’.

These arrival times are then compared with the predicted arrival times to generate the timing residuals for a pulsar. A pulsar model is used to predict arrival times and as our understanding of the pulsar improves, so does the prediction of the arrival times. Unmodelled or imprecise parameters produce structures in the timing residuals providing insights into missing parameters in the model or the presence of other signals such as a gravitational-wave background.

A pulsar model consists of several parameters that describe the pulsar system. Take a look at the *J1903-7051.par* in the main tutorial directory.

The first set of parameters as shown below provide information on the position of the pulsar on the sky (RAJ, DECJ), it’s rotational frequency (F0), the rate at which it slows down due to dissipation of energy (F1), higher-order frequency derivatives (like F2) and the dispersion measure of the pulsar (DM). If the pulsar is in a binary system, then there are additional parameters that describe the orbital properties.

The first column is the parameter, the second column is the value, the third column describes if the parameter is fitted for in TEMPO2 and the final column provides the uncertainty in the measurement of that parameter. The ephemeris for PSR J1903–7051, from the ATNF pulsar catalogue is listed below.

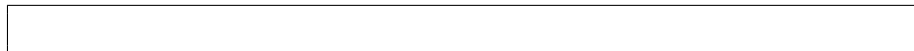
PSRJ	J1903-7051		
RAJ	19:03:38.7935		3.000e-04
DECJ	-70:51:43.461		2.000e-03
DM	19.6612		2.000e-04
PEPOCH	56526.0		
F0	277.94006243351		8.000e-11
F1	-8.06E-16		4.000e-18
PMRA	-8.8		1.600e+00
PMDEC	-16		2.000e+00
DMEPOCH	58836.74183		
BINARY	ELL1		
PB	11.05079833		2.000e-08
A1	9.938869		2.000e-06
TASC	56027.2292914		7.000e-07
EPS1	2.0261968E-6		5.000e-13
EPS2	1.1855635E-7		5.000e-14
RM	17.61		1.700e-01
EPHVER	2		
UNITS	TDB		

What are the different parameters in the parfile?

## 4 Pulsar times of arrival

The pulsar times of arrival are stored in a *.tim* file.

This file records the arrival time of every pulse profile of the pulsar along with information about the recording system, the length of each observation, the backend, etc. Read each column of this file carefully and if you have any questions, please ask.



## 5 Using tempo2

Once we have generated the arrival times and have an initial model (*.par* file) for the pulsar, we can compute the timing residuals using TEMPO2.

To run tempo2, on the command line type

```
tempo2 -gr plk -f J1903-7051.par J1903-7051.tim
```

The *-gr plk* command launches a graphical interface that allows interactive visualisation and analysis of the timing residuals. The tool has a lot of features and is very powerful for diagnosing pulsar timing data sets.

On running the above command, you should see a window as shown in the top panel of Figure 1.

- The parameters listed on the top describe the pulsar model (as listed in the *.par* file).
- The highlighted parameters (with red squares) indicate those that are fitted for.
- The plot with green squares show the timing residuals. When the pulsar model accurately describes the properties of the pulsar, the timing residuals would be scattered around a mean value of 0 – which is not the case here.
- The weighted root mean square value (Wrms) listed above the timing residuals describes the goodness of the fit and in this case, it is  $\sim 1$  microsecond.
- The various buttons on the left side of the timing residuals allow us to interactively view and analyse the data.

Try plotting the timing residuals as a function of:

- ToA error
- Serially
- Orbital phase

Apart from the graphical interface, TEMPO2 displays a wealth of information on the terminal. Were you able to spot it? If so, try answering the following questions:

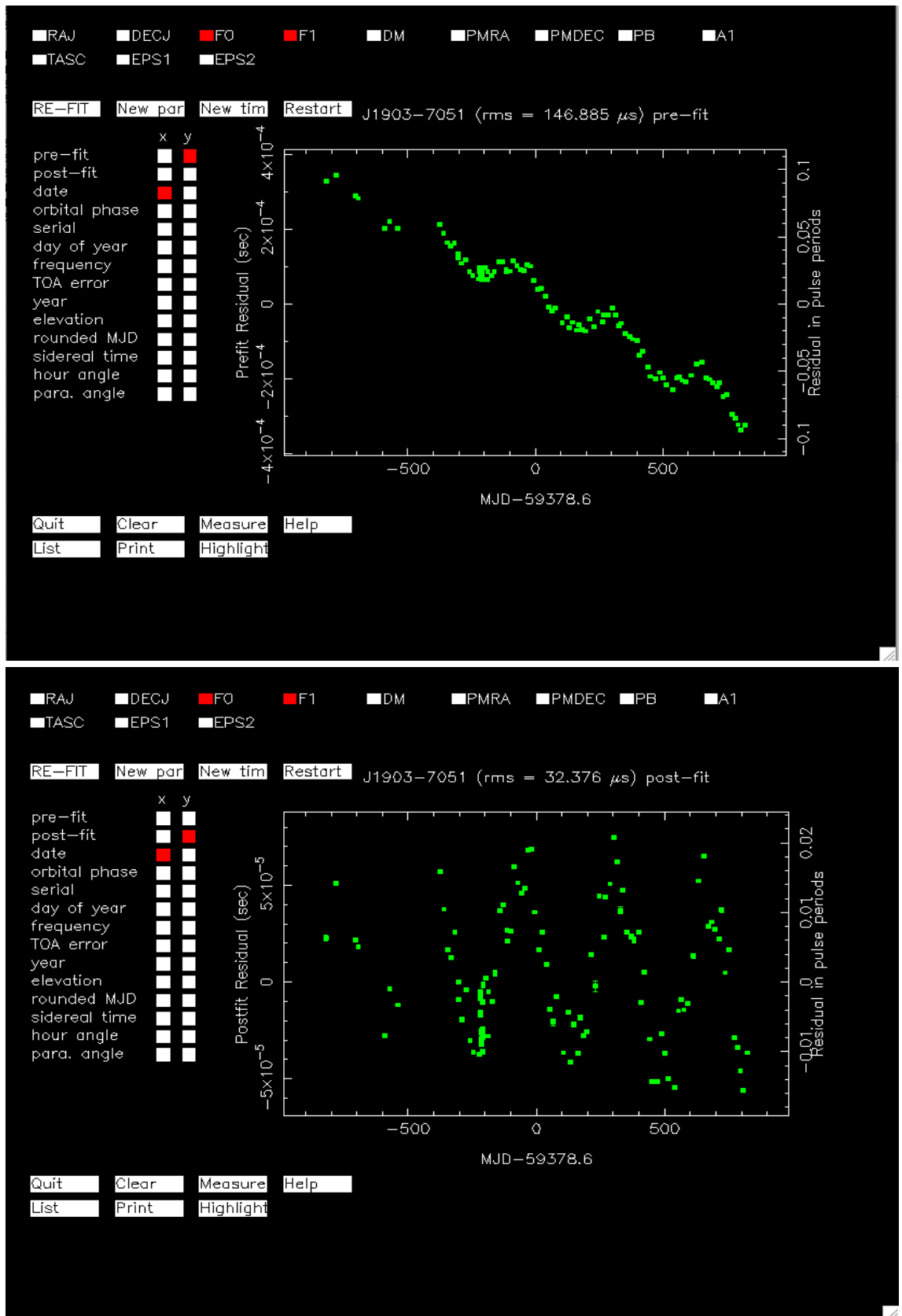


Figure 1: The TEMPO2 graphical interface. The top panel shows the pre-fit residuals and the bottom panel shows the post fit residuals, after fitting for spin frequency (F0) and spin down (F1).

- What is the time span of the data set?
- What is the inferred age of the pulsar?
- What is the reduced chi-square value (the goodness of fit)?

Time span of the data:  
 Age of the pulsar (in years):  
 Reduced chi-square value of the fit:  
 Number of fitted parameters:  
 Number of data points in the fit:

## 6 Trends in the timing residuals

There is clear structure in the timing residuals. The different effects can be identified by seeing how the TOAs vary over time, day of year, and orbital phase. In order to update the residuals it is necessary to fit fit parameters and potentially add new parameters to the model.

To fit for parameters you can hit the *Fit* button in the *plk* panel type *x* on the keyboard.

- First fit for F0 and F1.
- Then fit for RAJ, DECJ.
- Then fit for PMRA and PMDEC.

How did the values of the parameters change?

If there are additional unmodelled parameters they can be included in the parfile as additional lines.

Add the parameter *XDOT* (apparent change in the size of the semimajor axis) to the parfile:

XDOT 0 1

Is the parameter significant?

There is still a signature in the residuals. This is best visualised by plotting the residuals as a function of orbital phase.. To fit for the Shapiro signature, add these two lines to the parfile:

```
M2 0
SINI 0.9
```

From the Shapiro delay it is possible to measure the companion mass, and the system inclination angle. This enables the pulsar mass to be determined.

What is the pulsar mass?

How would you improve the measurement of the pulsar mass?

Save your parfile by clicking the *New Par* button in the *plk* window and entering a name for the file in the terminal (e.g., `J1903-7051_new.par`)

## 7 Exploring multi-frequency data

We have so far considered only looking at data that has been fully frequency averaged (F-scrunched). In order to undertake the most precise pulsar timing it is necessary to make pulsar timing measurements at multiple frequencies. For some projects, this involves observing pulsars at different frequencies with different receivers with the same telescope or even with different telescopes. MeerKAT benefits from having wide band systems, so it is possible to time pulsars over a 2:1 fractional bandwidth with a single system.

The tutorial data directory includes data produced from the same observations as the fully F-scrunched observations as used previously. The data have 16 channels, similar to that which is used in the MeerKAT Pulsar Timing Array observations.

Used one of the standards above to produce multil-frequency TOAs

```
pat -K 1/xs -t -A FDM -s <standard> -f 'tempo2' <cleaned_archives>
    > <J1903-7051_F16.tim>
```

Have a look at the contents of the new .tim file. How does the the new .tim file differ from the one produced from the f-scrunched data?

Now that a new data set has been produced, the first step is to inspect the data in tempo2. We will use the good .parfile made at the end of Section 5.

```
tempo2 -gr plk -f J1903-7051\_new.par J1903-7051_F16.tim
```



You will notice that there a number of outliers in the *plk* window. These are likely associated with low S/N sub-banded observations. (The pulsar varies in flux density in time and frequency due to scintillation, and it is likely that the pulsar had scintillated down in flux. It is important to identify *why* an observation is an outlier in pulsar timing data sets. There are a couple of ways to interactively remove the observations:

1. Click on the outlying observation using the middle mouse button (or equivalent)
2. After focusing on the plk window type shift-Z on your key board, this will result in the ability to draw a box in the *plk* window. Any TOA within the box will be removed.

**Save a new .tim file by clicking the New Tim, button on the *plk* window, and entering a name for the clean data in the terminal, e.g. J1903-7051\_F16\_clean.tim.** The "zapped" TOAs will remain in the new .tim file, but the first character of the TOA line will be C, indicating the TOA is commented out.

Now open the clean .tim file in plk.

```
tempo2 -gr plk -f J1903-7051_new.par J1903-7051_F16_clean.tim
```

What do the TOAs look like as a function of frequency?

After fitting for DM is there still structure in the residuals as a function of frequency?

Plot the residuals serially. Are there any trends?

## 8 Further work

When timing pulsars over broad bands, it is important to account for pulse profile evolution. Various tools have been devised to account for this. The most mature is referred to as *pulseportraiture*.

We have only considered the initial fitting of the pulsar data. In order to achieve the most accurate description of the pulsar (and use it to search for gravitational waves) it is necessary to develop models for all the noise processes in the data. These will be discussed in subsequent activities.