

# TEMPO2 user manual

George Hobbs, Russell Edwards

Australia Telescope National Facility, CSIRO, PO Box 76, Epping NSW 1710, Australia

*(documentation V2.0)*

# Contents

<b>1</b>	<b>INTRODUCTION</b>	<b>6</b>
<b>2</b>	<b>Obtaining and installing tempo2</b>	<b>6</b>
<b>3</b>	<b>A simple example of using tempo2</b>	<b>6</b>
<b>4</b>	<b>Required files</b>	<b>8</b>
4.1	Clock correction files . . . . .	8
4.1.1	Updating clock corrections . . . . .	9
4.2	Earth Orientation Parameters . . . . .	9
4.3	Time ephemeris . . . . .	9
4.4	Planetary ephemeris . . . . .	10
4.5	Observatory definitions . . . . .	10
<b>5</b>	<b>Parameter files</b>	<b>10</b>
5.0.1	tpo file format . . . . .	14
5.0.2	Changing the model epoch . . . . .	14
5.1	Jumps . . . . .	15
5.2	Removing Timing Noise . . . . .	15
5.3	Default values . . . . .	15
<b>6</b>	<b>Observation files</b>	<b>15</b>
6.1	Setting dispersion measures . . . . .	17
6.2	Jodrell Bank Observatory binary format . . . . .	17
<b>7</b>	<b>Atmospheric propagation delays</b>	<b>17</b>
<b>8</b>	<b>Command-line arguments</b>	<b>17</b>
8.1	The TEMPO1 emulation mode . . . . .	18
8.2	Filtering . . . . .	18
8.3	Aliases . . . . .	19
<b>9</b>	<b>The bootstrap fitting algorithm</b>	<b>19</b>
<b>10</b>	<b>Predictive mode</b>	<b>19</b>
10.1	TEMPO2 format . . . . .	19
10.2	TEMPO1 format . . . . .	19
<b>11</b>	<b>Global parameters</b>	<b>20</b>
<b>12</b>	<b>Output formats</b>	<b>20</b>
12.1	general . . . . .	21
12.2	general2 . . . . .	21
12.3	list . . . . .	22
12.4	stats . . . . .	22
<b>13</b>	<b>Graphical interfaces</b>	<b>23</b>
13.1	basic . . . . .	23
13.2	calcDM . . . . .	23
13.3	chisq . . . . .	23
13.4	compare . . . . .	24
13.5	compareRes . . . . .	24
13.6	delays . . . . .	24
13.7	errors . . . . .	25
13.8	fake . . . . .	25
13.9	gorilla . . . . .	26
13.10	plk . . . . .	27

13.11polytest . . . . .	29
13.12splk . . . . .	29
13.13stridefit . . . . .	29
13.14transform . . . . .	30
<b>14 Developing the software</b>	<b>31</b>
14.1 Creating a plug-in . . . . .	31
14.1.1 A new output format . . . . .	31
14.1.2 A new graphical interface . . . . .	32
14.2 The main source code . . . . .	34
<b>15 Tempo2 error and warning messages</b>	<b>34</b>
15.1 Error messages . . . . .	34
15.2 Warning messages . . . . .	34
<b>16 Common questions</b>	<b>35</b>
<b>17 In progress</b>	<b>35</b>
<b>18 Acknowledgements</b>	<b>35</b>

## List of Figures

1	a) pre-fit timing residuals for the test data-set and b) post-fit timing residuals. . . . .	8
2	An example of the ‘basic’ graphical interface. A $P-\dot{P}$ diagram is produced with the pulsar being analysed highlighted. . . . .	23
3	Example of the chisq plug-in for determining the most likely values of the orbital period, PB, and longitude of periastron, OM. . . . .	24
4	The solar system Shapiro delay for PSR J1810–2005 shown using the DELAYS graphical interface plug-in . . . . .	25
5	Simulating pulsar timing residuals for a pulsar with a proper motion in right ascension of 200 mas/yr using the FAKE plugin. . . . .	27
6	An example of the plk graphical interface in use. The post-fit residuals for PSR J0437–4715 are plotted with 20 observations shown in green and 10 cm observations in red. The white circles and black lines indicate arrival times between orbital phases 0.4 and 0.6. . . . .	28
7	Example of the SPLK graphical interface plug-in for showing the pre- and post-fit timing residuals for multiple pulsar simultaneously. . . . .	29
8	Example of the STRIDEFIT graphical interface plug-in which calculates the pulsar’s dispersion measure as a function fo time. . . . .	30

## List of Tables

1	Observatory details . . . . .	11
2	Pulsar parameters that can be entered in a parameter file . . . . .	12
3	Binary parameters that can be entered in a parameter file . . . . .	13
4	Flags in arrival time files . . . . .	16
5	Commands that may be included in an arrival time file . . . . .	17

# 1 INTRODUCTION

TEMPO2 is a new version of the TEMPO pulsar timing software. An overview of the software is provided in Hobbs, Edwards & Manchester (2006; MNRAS 369 655). A second paper provides mathematical details of the algorithms used in the software (Edwards, Hobbs & Manchester 2006; currently available from astro-ph). A third paper will describe how TEMPO2 can be used to simulate the effects of gravitational waves on pulsar timing residuals. A summary of the basic features can also be found in Hobbs, Edwards & Manchester (2006; in press CHJAA).

This document provides full usage instructions for TEMPO2.

## 2 Obtaining and installing tempo2

The TEMPO2 software and documentation can be obtained from <http://www.atnf.csiro.au/research/pulsar/ppta/tempo2>. Click on the “Download” option to download the software to your local machine. This software requires compilation before it will run. Up-to-date installation instructions are available in a README file with the download.

Note: TEMPO2 makes heavy use of “long double” precision in its calculations. Most compiler/architecture combinations support long doubles of 80 or 128 bits in size, which is sufficient. TEMPO2 has been successfully tested under Linux-gcc/x86, Solaris/SPARC and Mac OS 10.4-gcc/PowerPC. Unfortunately, some systems only provide 64-bit long doubles (i.e. identical to a standard “double”); these include Mac OS 10.3.9 and earlier, and many Windows compilers. While parts of the source code make reference to a software quad-precision library, this feature is no longer functional.

## 3 A simple example of using tempo2

The TEMPO2 website (<http://www.atnf.csiro.au/research/pulsar/tempo2>) provides a set of example files for use to test the software. PSR1.PAR contains the catalogued parameters for PSR J0437–4715 in standard TEMPO2 format (see §5). PSR1.TIM contains a set of simulated observations of this pulsar over a 10 yr period with an rms residual of 100 ns. See §6 for details of the contents of this file. Running

```
tempo2 -f psr1.par psr1.tim
```

should provide some information about clock correction files and also

```
Results for PSR J0437-4715
```

```
RMS pre-fit residual = 0.096 (us), RMS post-fit residual = 0.096 (us)
Number of points in fit = 367
```

PARAMETER	Pre-fit	Post-fit	Uncertainty	Difference	Fit
RAJ (rad)	1.20978853473707	1.20978853473707	0	0	N
RAJ (hms)	04:37:15.7865145	04:37:15.7865145	0	0	
DECJ (rad)	-0.824709094464799	-0.824709094464799	0	0	N
DECJ (dms)	-47:15:08.46158	-47:15:08.46158	0	0	
FO (s <sup>-1</sup> )	173.687946306032	173.687946306032	0	0	N
F1 (s <sup>-2</sup> )	-1.7283139464043e-15	-1.7283139464043e-15	0	0	N
PEPOCH (MJD)	51194.0001248168	51194.0001248168	0	0	N
POSEPOCH (MJD)	51194.0001248168	51194.0001248168	0	0	N
DMEPOCH (MJD)	51194	51194	0	0	N
DM (cm <sup>-3</sup> pc)	2.64690012312213	2.64690012312213	0	0	N
PMRA (mas/yr)	121.43799811708	121.43799811708	0	0	N
PMDEC (mas/yr)	-71.4379988923397	-71.4379988923397	0	0	N
PX (mas)	7.19	7.19	0	0	N
SINI	0.6788	0.6788	0	0	N
PB (d)	5.74104608901605	5.74104608901605	0	0	N
TO (MJD)	51194.6240248265	51194.6240248265	0	0	N
A1 (lt-s)	3.36669162220122	3.36669162220122	0	0	N
OM (deg)	1.2	1.2	0	0	N
ECC	1.9186e-05	1.9186e-05	0	0	N

PBDOT (10 <sup>-12</sup> )	3.64	3.64	0	0	N
OMDOT (deg/yr)	0.0159999997519168	0.0159999997519168	0	0	N
M2	0.236	0.236	0	0	N
START (MJD)	50640.9281162413	49350.5129309451	0	-1290.4	N
FINISH (MJD)	52088.8971386924	53000.5197060478	0	911.62	N
TRACK (MJD)	0	0	0	0	N
TZRMJD	51204.6438924841	51204.6438924841	0	0	N
TZRFRQ (MHz)	1413.39997808495	1413.39997808495	0	0	N

Binary model: DD  
 Mass function = 0.001243113190 +- 0.000000000000 solar masses  
 Minimum companion mass = 0.1403 solar masses  
 Median companion mass = 0.1637 solar masses  
 Maximum companion mass = 0.3493 solar masses  
 MTOT derived from sin i and m2 = 1.8186068413766  
 Inclination angle (deg) = 42.74994182955 (+ 0 - 0)

PSR1.2.PAR is similar to PSR1.PAR except that the parameter values have been changed slightly from their true values. As above, running

tempo2 -f psr1\_2.par psr1.tim

should produce

Results for PSR J0437-4715

RMS pre-fit residual = 6.895 (us), RMS post-fit residual = 0.096 (us)  
 Number of points in fit = 367

PARAMETER	Pre-fit	Post-fit	Uncertainty	Difference	Fit
RAJ (rad)	1.2097885336826	1.20978853474746	2.1206e-11	1.0649e-09	Y
RAJ (hms)	04:37:15.7865	04:37:15.7865146	2.916e-07	1.4643e-05	
DECJ (rad)	-0.824709094076948	-0.824709094473468	1.5157e-11	-3.9652e-10	Y
DECJ (dms)	-47:15:08.4615	-47:15:08.46158	3.1264e-06	-8.1788e-05	
FO (s <sup>-1</sup> )	173.68794630603	173.687946306032	9.5579e-15	2.3306e-12	Y
F1 (s <sup>-2</sup> )	-1.728e-15	-1.72831367406148e-15	2.3325e-22	-3.1367e-19	Y
PEPOCH (MJD)	51194.0001248168	51194.0001248168	0	0	N
POSEPOCH (MJD)	51194.0001248168	51194.0001248168	0	0	N
DMEPOCH (MJD)	51194	51194	0	0	N
DM (cm <sup>-3</sup> pc)	2.64690012312213	2.64690012312213	0	0	N
PMRA (mas/yr)	121.43799811708	121.43799811708	0	0	N
PMDEC (mas/yr)	-71.4379988923397	-71.4379988923397	0	0	N
PX (mas)	7.19	7.19	0	0	N
SINI	0.6788	0.6788	0	0	N
PB (d)	5.741046089	5.74104608900813	1.0437e-11	8.1286e-12	Y
TO (MJD)	51194.6240248265	51194.6240248265	0	0	N
A1 (lt-s)	3.36669162220122	3.36669162220122	0	0	N
OM (deg)	1.2	1.2	0	0	N
ECC	1.9186e-05	1.9186e-05	0	0	N
PBDOT (10 <sup>-12</sup> )	3.64	3.64	0	0	N
OMDOT (deg/yr)	0.0159999997519168	0.0159999997519168	0	0	N
M2	0.236	0.236	0	0	N
START (MJD)	50640.9281162413	49350.5129309451	0	-1290.4	N
FINISH (MJD)	52088.8971386924	53000.5197060478	0	911.62	N
TRACK (MJD)	0	0	0	0	N
TZRMJD	51204.6438924841	51204.6438924841	0	0	N
TZRFRQ (MHz)	1413.39997808495	1413.39997808495	0	0	N

Binary model: DD  
 Mass function = 0.001243113190 +- 0.000000047151 solar masses  
 Minimum companion mass = 0.1403 solar masses  
 Median companion mass = 0.1637 solar masses  
 Maximum companion mass = 0.3493 solar masses  
 MTOT derived from sin i and m2 = 1.818606841374  
 Inclination angle (deg) = 42.74994182955 (+ 0 - 0)

The residuals can be inspected using

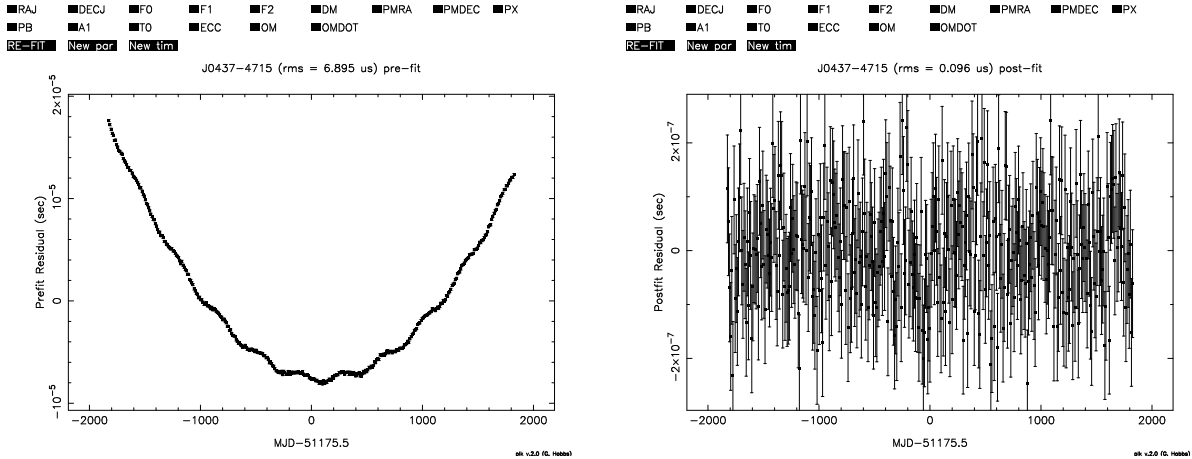


Figure 1: a) pre-fit timing residuals for the test data-set and b) post-fit timing residuals.

```
tempo2 -gr plk -f psr1_2.par psr1.tim
```

which should produce a plot of the pre-fit residuals as shown in Figure 1a. Pressing '2' on the plot will show the post-fit residuals (Figure 1b).

## 4 Required files

TEMPO2 requires certain files in order to run correctly. These files are provided with the download and are discussed in the following sections.

### 4.1 Clock correction files

Times of arrival provided to TEMPO2 are recorded against local observatory clocks. These times differ from those recorded against a uniform clock, firstly because observatory clocks are typically maintained in approximate synchrony with Coordinated Universal Time (UTC), which itself is not uniform, and secondly because they deviate from ideal UTC owing to deviations in uniformity in the underlying frequency standard (usually a hydrogen maser). The ultimate aim of the clock correction process is to transform all site arrival times to a chosen realisation of TT (Terrestrial Time), which in an ideal realisation is a uniform clock ticking SI seconds on the geoid. By default this is TT(TAI), which (since 1971) differs from UTC by a constant offset plus an integer number of leap seconds. Alternative realisations of TT can be specified using the the CLOCK keyword in the parameter file.

The clock correction process proceeds entirely on the basis of linear interpolation of user-supplied tabulations of the difference between named pairs of clocks, as a function of Modified Julian Day (MJD)<sup>1</sup>. These files reside in the directory \$TEMPO2/clock. Lines beginning with the hash character (#) are treated as comments. The first line must be a comment specifying the name of the clock to convert from, the name of the clock to convert to, and an optional “badness” value (which defaults to 1). For example, the following specifies that the values in the file can be added to times measured against the Parkes clock (“UTC(PKS)”) to transform them to the frame of the Global Positioning System (GPS) clock (“UTC(GPS)”).

```
# UTC(PKS) UTC(GPS) 10
```

Non-comment lines consist of a sequence of pairs of MJDs and offsets (in seconds), specifying the difference between the second and first clocks as a function of date. For example:

```
50844.72917 -7.49068e-07
50845.77083 -7.47637e-07
50846.81250 -7.46650e-07
...
```

<sup>1</sup>The frame in which the MJD is measured is not specified: it is assumed that clock offsets and drift rates are small enough that if  $t' = t + f(t)$  then  $t \simeq t' - f(t')$ .



The spacing of the dates need not be any specific value, or even be regular. For most purposes roughly daily values are suitable.

All files ending in `.clk` in `$TEMPO2/clock` are read by TEMPO2 when it starts executing. Then, given a TOA to transform, it obtains the name of the clock against which it was measured based upon name specified in the observatory database (§4.5), given the observatory code recorded in the TOA file. Given the source and destination clocks, TEMPO2 must then choose a selection of clock correction tables (from `.clk` files) to use for the transformation. This is firstly attempted by consulting the list of pre-defined transformation paths, which are defined using `CLK_CORR_CHAIN` entries in the parameter file. For example, the following tells TEMPO2 to convert from UTC(PKS) to TT(TAI) using tables defined in `pks2gps.clk`, `gps2utc.clk`, `utc2tai.clk` and `tai2tt_tai.clk`:

```
CLK_CORR_CHAIN pks2gps gps2utc utc2tai tai2tt_tai
```

This parameter may be specified multiple times. TEMPO2 will attempt to apply each path in the order in which they were specified (which may fail if the MJD of the TOA is outside the range of component tables).

If no applicable pre-defined paths are found, TEMPO2 find the “best” possible path using all of the available tables. Here “best” means the path for which the sum of badness values is minimized. Tie-breaking is arbitrary. This path is then appended to the global list of pre-defined paths. Since TEMPO2 always checks this list before attempting automatic path construction, subsequent transformations will always use this path if it is applicable, even if the MJDs of some TOAs would have allowed for a “better” path. Caution is therefore advised in using the automatic path construction feature when multiple paths exist.

#### 4.1.1 Updating clock corrections

The distribution of TEMPO2 includes several useful files containing corrections based on the BIPM’s Circular T (offsets between UTC and its various realisations, as well as the GPS clock) and the IERS Bulletin C (announcing leap seconds). A suite of ancillary software is available on the TEMPO2 website, which provides among other things a means for parsing Circular T to update the relevant clock correction files (`update_clkcorr`). This program can also parse clock monitoring data from the Parkes Observatory. Interested parties are invited to contribute code for the parsing of clock data from other sources.

## 4.2 Earth Orientation Parameters

To compute the Roemer delay, the position of the observatory must be known. This depends not only on the Earth’s orbit, but on the Earth’s orientation and rotation. The necessary parameters are obtained by interpolation of the “C05” series of Earth Orientation Parameters (EOPs) from the IERS. The file `$TEMPO2/earth/README` specifies the web address for downloading the latest EOPs. The user may optionally select to emulate the algorithm of `tempo` (which neglects polar motion and uses an out of date precession/nutation model) for transforming the observatory coordinates to the celestial frame, using the `T2C_METHOD` parameter; in this case `$TEMPO2/clock/ut1.dat` (in the same format as the corresponding file for `tempo`) is used.

## 4.3 Time ephemeris

The pulse arrival times at the observatory at transformed to the arrival time at the solar system barycentre (SSB). In this transformation the *Einstein delay*, which describes the combined effect of gravitational redshift and time dilation due to the motion of the Earth and other bodies, must be taken in to account. This transformation converts the site arrival time from TT to a coordinate time at the SSB, known as Barycentric Coordinate Time (TCB). Optionally, for backward compatibility with `tempo` the user may also choose to use a scaled version of this frame in which the mean drift relative to TT is divided out: this is nominally (but incorrectly; see Paper II) referred to as TDB. This is accomplished by specifying “UNITS TDB” in the parameter file.

The Einstein delay is computed using a polynomial approximation to the numerical evaluation of the time dilation integral as provided by Irwin & Fukushima (1999). It lives in `$TEMPO2/ephemeris/TIMEEPH_short.te405`. For reproducing results obtained with `tempo`, the user may also chose to use the Fairhead & Bretagnon (1990) version of this integral (stored at `$TEMPO2/ephemeris/TDB.1950.2050`) by specifying “TIMEEPH FB90” in the pulsar parameter file.

## 4.4 Planetary ephemeris

In order to correct the arrival time to the solar system's barycentre, TEMPO2 requires a solar system ephemeris. By default the JPL ephemeris DE200 is chosen. Different JPL ephemerides may be selected using the EPHEM\_FILE command in the parameter file. For example,

```
EPHEM_FILE /pulsar/psr/runtime/tempo//tempo_ephem/DE405.1950.2050
```

would select the DE405 JPL ephemeris. If the full-path is defined from \$TEMPO2/ephemeris then the DE405 ephemeris could be selected from

```
EPHEM DE405
```

## 4.5 Observatory definitions

It is necessary for TEMPO2 to know the coordinates of the observatory. In the original TEMPO a file, OBSYS.DAT, was used that contained the coordinates of each observatory and a single-character identifying code. This code was used in the arrival time file. Unfortunately, different users used different codes for the same observatory and therefore the arrival time files were not transferable between different installations of TEMPO. To avoid this, TEMPO2 provides a **read-only** database of observatories, each identified by a short, non-cryptic mnemonic. This resides in \$TEMPO/observatory/observatories.dat. In addition, for backwards compatibility or for experimental use, further definitions can be placed in extra files: TEMPO2 parses every file in \$TEMPO/observatory/. Each line should contain 5–6 whitespace-separated parameters. These are, in order, the x, y and z geocentric coordinates (in metres), a one-word name for the observatory, a few-character mnemonic and optionally the name of the clock associated with the observatory (used to refer to the relevant clock-correction tables). If not supplied, the clock name is constructed as UTC(xxx) where xxx is the observatory mnemonic.

For full accuracy, observatory coordinates should be specified in the International Terrestrial Reference System. Geodetic coordinates (as optionally used by TEMPO, given as latitude and longitude in degrees in the form dddmmss.ss, and height in metres) may be specified, in which case TEMPO2 will detect this and convert them to the ITRF on the assumption that they refer to the GRS80 geoid. The converted coordinates are displayed and execution is halted for the user to add the converted coordinates to the observatories database (or not! the accuracy of the conversion and the assumption of GRS80 may be dubious).

*NOTE: The mnemonics in observatories.dat have not been finalised. Please let G. Hobbs or R. Edwards know if you prefer another mnemonic for your observatory.*

The current observatory file is listed in Table 1.

## 5 Parameter files

The parameter files have the same design as in the earlier tempo implementations. Each of the pulsar parameters has a label, a value and may have an uncertainty on the value and a flag indicating whether TEMPO2 should fit for this parameter or whether this parameter should be held constant (0 = default = hold constant; 1 = fit). These labels are described in Table 2. An example of a parameter file for PSR J0437–4715 taken from the catalogue and fitting for various parameters:

PSRJ	J0437-4715		
RAJ	04:37:15.7865145	1	7.000e-07
DECJ	-47:15:08.461584	1	8.000e-06
DM	2.6469		1.000e-04
PEPOCH	51194.000		
F0	173.6879489990983	1	3.000e-13
F1	-1.728314E-15	1	1.600e-20
PMRA	121.438		6.000e-03
PMDEC	-71.438		7.000e-03
BINARY	DD		
PB	5.741046	1	3.000e-06
ECC	1.9186E-5	1	5.000e-09
A1	3.36669157	1	1.400e-07
TO	51194.6239	1	8.000e-04

Table 1: Observatory details

x	y	z	Mnemonic	Clock
882589.65	-4924872.32	3943729.348	GBT	gbt
-4752329.7000	2790505.9340	-3200483.7470	NARRABRI	atca
2390490.0	-5564764.0	1994727.0	ARECIBO	ao
-228310.702	4631922.905	4367064.059	NANSHAN	nanshan
-4460892.6	2682358.9	-3674756.0	DSS_43	tid43
-4554231.5	2816759.1	-3454036.3	PARKES	pks
3822252.643	-153995.683	5086051.443	JODRELL	jb
-1601192.	-5041981.4	3554871.4	VLA	vla
4324165.81	165927.11	4670132.83	NANCAJ	ncy
4033949.5	486989.4	4900430.8	EFFELSBURG	eff
3822252.643	-153995.683	5086051.443	JODRELLM4	jbm4
881856.58	-4925311.86	3943459.70	GB300	gb300
882872.57	-4924552.73	3944154.92	GB140	gb140
882315.33	-4925191.41	3943414.05	GB853	gb853
383395.727	-173759.585	5077751.313	MKIII	j
3817176.557	-162921.170	5089462.046	TABLEY	k
3828714.504	-169458.987	5080647.749	DARNHALL	l
3859711.492	-201995.082	5056134.285	KNOCKIN	m
3923069.135	-146804.404	5009320.570	DEFFORD	n
0.0	1.0	0.0	COE	coe
3822473.365	-153692.318	5085851.303	JB_MKII	jbm2
3822294.825	-153862.275	5085987.071	JB_42ft	jb42
1719555.576	5327021.651	3051967.932	LA_PALMA	p

OM	1.20	1	5.000e-02
OMDOT	0.016		1.000e-02
START	50640.928		
FINISH	52088.897		
CLK	UTC(NIST)		
EPHEM	DE200		
PBDOT	3.64E-12		2.000e-13
TZRMJD	51204.64376750220085		
TZRFRQ	1413.400		
TZRSITE	7		
RM	+1.5		5.000e-01
PX	7.19		1.400e-01
SINI	0.6788		1.200e-03
M2	0.236		1.700e-02

In more detail, for a pulsar which a spin period of  $P0 = 1.23456$  s, with no fitting required:

```
P0 1.23456
```

To fit to this parameter, use

```
P0 1.23456 1
```

or with an uncertainty (which is ignored by TEMPO2)

```
P0 1.23456 1 0.00003
```

Other commands may be given in parameter files that control the algorithms used by TEMPO2. TEMPO2 only requires the following parameters: PSRJ, DM, F0, PEPOCH, RAJ and DECJ. If no period epoch is provided then the position epoch is assumed to be the same as the period epoch.

It is also possible to provide the pulsar parameters in the old-style tempo format where the arrival times and the parameters are given in the same file. In this mode TEMPO2 is called only using one file, e.g.

Table 2: Pulsar parameters that can be entered in a parameter file

Label	Description	Units
PSRJ, PSRB or PSR	Pulsar name	—
FX (e.g F0, F1, F2)	The X'th time derivative of the rotational frequency	$(s^{-(X-1)})$
P0 or P	Spin period of pulsar	sec.
P1 or Pdot	Spin down rate of pulsar	$(\times 10^{-15})$
PEPOCH	Epoch of period determination	MJD
RAJ or RA	J2000 right ascension	hh:mm:ss.sss
DECJ or DEC	J2000 declination	dd:mm:ss.sss
ELONG or LAMBDA	Ecliptic longitude	deg
ELAT or BETA	Ecliptic latitude	deg
POSEPOCH	Epoch of position measurement. If not present a warning is given and PEPOCH is used instead	MJD
PMLAMBDA or PMELONG	Proper motion in ecliptic longitude	mas/yr
PMBETA or PMELAT	Proper motion in ecliptic latitude	mas/yr
PMRA	proper motion in right ascension	mas/yr
PMDEC	proper motion in declination	mas/yr
DMEPOCH	Epoch of DM measurement	MJD
DM	Dispersion measure	$\text{cm}^{-3}\text{pc}$
DMX	X'th time derivative of the dispersion measure	$\text{cm}^{-3}\text{pcs}^{-X}$
FDD	Frequency-dependent delay	
PX	Parallax	mas
PMRV	Radial velocity	
GLEP_X	Glitch epoch	MJD
GLPH_X	Glitch phase increment	
GLF0_X	Glitch permanent pulse frequency increment	Hz
GLF1_X	Glitch permanent frequency deriv increment	$s^{-2}$
GLF0D_X	Glitch pulse frequency increment	Hz
GLTD_X	Glitch Decay time constant	Hz
WAVE_OM	Frequency of fundamental sinusoid for whitening	
WAVEX	Amplitude of sine and cosine for the X'th harmonic for whitening	
TEMPO1	Run in tempo emulation mode: e.g. TDB units	
UNITS	Set units to SI or TDB	
MODE	Fitting with errors MODE 1, or without MODE 0	
JUMP	Add a constant offset between specified TOAs	
CLK	Definition of clock to use	
TRES	Rms timing residual	$(\mu s)$
NOTRACK	Switch off tracking mode	
NO_SS_SHAPIRO	Switch off the calculation of the Solar system Shapiro delay	
IPM	= 0 to switch off calculation of the interplanetary medium	
NITS	Number of iterations for the fitting routines	
DILATEFREQ	Whether or not to apply gravitational redshift and time dilation to observing frequency (Y/N)	
IBOOT	Number of iterations used in the bootstrap fitting method	
PLANET_SHAPIRO		
CORRECT_TROPOSPHERE	Whether or not to apply tropospheric delay corrections	
NE1AU	The electron density at 1 AU due to the solar wind	
TIMEEPH	Which time ephemeris to use (IF99/FB90)	
T2CMETHOD	Method for transforming from terrestrial to celestial frame (IAU2000B/TEMPO)	
CLK_CORR_CHAIN	Clock correction chain(s) to use (Sect. 4.1)	
EPHEM	Which solar system ephemeris to use	
TZRMJD		
TZRSITE		
NSPAN or TSPAN		
TZRFRQ		
START		
FINISH		
EPHVER		
TRACK		

Table 3: Binary parameters that can be entered in a parameter file

Label	Description	Units
BINARY	Binary model (BT/ELL1/DD/MSS)	
A1	Projected semi-major axis of orbit	lt-sec
PB	Orbital period	days
ECC or E	Eccentricity of orbit	—
T0	Epoch of periastron	MJD
OM	Longitude of periastron	degrees
TASC	Epoch of ascending node	MJD
EPS1	$ECC \times \sin(OM)$ for ELL1 model	—
EPS2	$ECC \times \cos(OM)$ for ELL1 model	—
OMDOT	Rate of advance of periastron	(deg/yr)
PBDOT	1st time derivative of binary period	( $10^{-12}$ )
A1DOT or XDOT	Rate of change of projected semi-major axis	( $10^{-12}$ )
SINI	Sine of inclination angle	
M2	Companion mass (solar masses)	
XPBDOT	Rate of change of orbital period minus GR prediction	
A1DOT		
EDOT or ECCDOT	Rate of change of eccentricity	
OMDOT	Periastron advance	degrees/yr
PBDOT	1st time derivative of binary period	
PBX	X'th time derivative of binary period	
GAMMA	post-Keplerian 'gamma' term	s
DR	Relativistic deformation of the orbit	
DTH	Relativistic deformation of the orbit	
A0	Aberration parameter A0	
B0	Aberration parameter B0	
BP	Tensor multi-scalar parameter beta-prime	
BPP	Tensor multi-scalar parameter beta-prime-prime	
DTHETA	Relativistic deformation of the orbit	
XOMDOT	Rate of periastron advance minus GR prediction	deg/yr
AFAC		
A1DOT or XDOT		
TASC	Epoch of ascending node	MJD
EPS1DOT		
EPS2DOT		
KOM		
KIN		
SHAPMAX		
MTOT	Total system mass	solar masses
BPJEP_X		
BPJPH_X		
BPJA1_X		
BPJEC_X		
BPJOM_X		
BPJPB_X		

```
tempo2 -gr plk myfile.tempo
```

where the first line in the file contains flags with '1' to indicate that the parameter should be fit and '0' for not fitting:

Column	Parameter
1	Phase
2	P0
3	P1
4	P2
5	RAJ
6	DECJ
7	PMRA
8	PMDEC
9	A1
10	ECC
11	T0
12	PB
13	OM
14	OMDOT
15	GAMMA
16	DM

17	PX
18	PBDOT
19	M1 (not implemented)
20	SINI
21	MTOT
22	M2
23	DTHETA
24	BP (not implemented)
25	BPP (not implemented)
27	Binary model: 0 = none, 1 = BT, 2 = EH, 3 = DD, 4 = DDGR, 5 = H88, 6 = BT+, 7 = DDT, 8 = DD+, 9 = 2 BT orbits

The actual parameter values are given on three extra lines. The second line in the file gives

Columns	Parameter
1-12	Pulsar name
21-40	RAJ in the form hhmmss.ss
41-60	DECJ in the form ddmss.ss
61-70	PMRA
71-80	PMDEC

The third line gives

Columns	Parameter
2-20	P0
21-40	P1
41-60	PEPOCH as Julian date
61-70	P2
71-80	PX

The fourth line gives

Columns	Parameter
9-20	DM

Further lines provide the binary parameters (not implemented yet).

### 5.0.1 tpo file format

Both the parameters and the arrival times may be provided in a single file. The first line of this file must be

HEAD

followed by the parameters in the same format as above. These are followed by

TOAS

and a list of arrival times.

### 5.0.2 Changing the model epoch

It is possible to convert the period epoch from that given in the parameter file using the `-epoch` command on the command line. For instance `-epoch 52000` will update the period epoch, rotational frequency and its first derivative to the new epoch using the inputted values of F0, F1 and F2. Can use `-epoch centre` (or `-epoch center`) to set to the centre of the data span. `-epoch left` will set the epoch to the first observation.

## 5.1 Jumps

It is often necessary to fit for a constant offset between two sets of arrival times. For instance, the templates used to determine arrival times at different frequencies may not be perfectly aligned or an offset may exist between the arrival times provided by different observatories. The command `JUMP` in the parameter file is used to define jumps.

`JUMP MJD v1 v2` will provide a jump between all TOAs with MJDs between `v1` and `v2` compared to all the other TOAs.

`JUMP FREQ v1 v2` will section all TOAs with observing frequencies between `v1` and `v2` MHz.

`JUMP TEL id` will section all TOAs observed with telescope ‘`id`’.

`JUMP NAME str` will section all TOAs with observation IDs that contain the string ‘`str`’.

`JUMP flag val` will select all TOAs with specified flag (e.g. `-o`) and value `val`.

## 5.2 Removing Timing Noise

Even with accurate spin and positional parameters the residuals for some (particularly the young) pulsars contain remnant structures. Some of these structures are understood: cusps, for instance, signify sudden changes in the pulsar’s spin rate during a glitch, sinusoidal oscillations can represent unmodelled companions (such as planets) or the pulsar precessing. However, many of the structures seen in the residuals are still not understood and are known as “timing noise”. To obtain the most accurate pulsar’s positional and proper motion parameters (and dispersion measure) it is essential to remove this timing noise. This has traditionally been carried out by fitting higher order pulsar rotational derivative terms. More recently, Hobbs et al. (2004) described a method for fitting harmonically related sinusoids.

## 5.3 Default values

# 6 Observation files

For each pulsar an arrival time file must be created that contains all the site-arrival-times<sup>2</sup> (i.e. the pulse arrival time at the observatory for each observation<sup>3</sup>). These files can take the form of the old Parkes- or Jodrell-style TEMPO files or may use a new formatting structure. The current TEMPO2 format is identified with

```
FORMAT 1
```

at the start of the observation file. Each observation line can be entered in a “free-format” manner (there is no limit on the number of decimal places or characters supplied for each parameter). The file has the following form:

```
file freq sat satErr siteID <flags>
```

where the flags are listed in Table 4. It is important for the TOAs to be given to high precision. TEMPO2 reads all parameters with ‘long double’ precision. Other undefined flags may be used in the arrival time files. These flags can provide any information and could be used, for example, in determining colours used for plotting with a personal graphical interface. An example observation file is given below.

```
FORMAT 1
```

```
C ../archives/w040205_062810.cFTp 3072.52800000 53040.27037033597299853 10.07000 7 -i WBC_10
../archives/w040206_070831.cFTp 3092.99900000 53041.31851839551620031 1.16000 7 -i WBC_10
C ../archives/w040206_084839.cFTp 3068.03800000 53041.38807867413299846 9.97000 7 -i WBC_10
../archives/w040206_111139.cFTp 3105.49900000 53041.47201962440929890 1.15000 7 -i WBC_10
../archives/w040207_070619.cFTp 3092.99900000 53042.30720476755460169 1.21000 7 -i WBC_10
../archives/w040207_081328.cFTp 1367.99900000 53042.35109949197099866 1.09000 7 -i WBC_20
../archives/w040207_084227.cFTp 1415.14600000 53042.36843163794929978 0.98000 7 -i WBC_20
../archives/w040207_115804.cFTp 1431.21700000 53042.50986685147659827 0.98000 7 -i WBC_20
../archives/w040207_142934.cFTp 1431.43500000 53042.60951958276880092 0.99000 7 -i WBC_20
../archives/w040208_081840.cFTp 1563.91900000 53043.34710640092290035 0.82000 7 -i WBC_20
../archives/w040208_083501.cFTp 1432.49900000 53043.36886551809849877 0.78000 7 -i WBC_20
../archives/n2004-06-27-04:13:58.FTp 685.24900000 53183.18848980008860039 0.06000 7 -i nCPSR2_50
```

<sup>2</sup>site-arrival-times can be obtained from pulse profiles and a standard template using software packages such as `pat`.

<sup>3</sup>note: the “pulse” arrival time is measured by the summation of many thousands of individual pulses from the pulsar.

Table 4: Flags in arrival time files

Flag	Parameter	Type
dm	DM ( $\text{cm}^{-3}\text{pc}$ )	float
p	Phase offset	float
t	Telescope identifier	string

```

../archives/n2004-07-15-18:33:23.FTp 685.24900000 53201.79496983790510001 0.08000 7 -i nCPSR2_50
../archives/n2004-07-15-19:39:47.FTp 685.24900000 53201.81959222764389850 0.22000 7 -i nCPSR2_50
../archives/n2004-07-15-19:42:31.FTp 685.24800000 53201.84208218378509869 0.19000 7 -i nCPSR2_50

```

The PAT software in the PSRCHIVE package has been updated to output arrival time files using this (or earlier) formats.

A list of available commands that can be included in an arrival time file are provided in table 5. Note that, when writing out a new arrival time file from TEMPO2 (using a plugin such as plk, for example - see Section 13.10), most of these commands will not be replicated, but are instead directly executed. (EFLOOR, EMIN, EMAX and EQUAD for example will be absorbed in the TOA uncertainties.) The exception to this is EFAC, which is written to the tim-file without affecting the uncertainties. (GLOBAL\_EFAC, T2EFAC and EFAC are combined into a single EFAC value, though.)

The old Parkes-style structure contains a label indicating the observation (such as a file name containing the folded profile), the observing frequency (MHz), the site-arrival-time (MJD), a phase offset ( $\mu\text{s}$ ), the uncertainty on the TOA ( $\mu\text{s}$ ) and an identification flag for the telescope used. These identification flags are defined by the user and differ between users. For example,

```
f981016_092219.FT 0 0 1374.000 51102.3925603118473 0.00 138.00 7
```

TEMPO2 can read arrival times in the “Parkes”, “Princeton” and “ITOA” formats. All these formats are “fixed-format” and are defined as:

#### Princeton Format

columns	item
1-1	Observatory code (1-character)
2-2	Must be blank
16-24	Observing frequency (MHz)
25-44	TOA (decimal point must be in column 30 or 31)
45-53	TOA uncertainty
69-78	DM correction (NOT IMPLEMENTED IN TEMPO2)

#### Parkes Format

columns	item
1-1	Must be blank
26-34	Observing frequency (MHz)
35-55	TOA (decimal point must be in column 42)
56-63	Phase offset (fraction of P0, added to TOA)
64-71	TOA uncertainty
80-80	Observatory (1-character)

#### ITOA Format

columns	item
1-2	ignored, but must not be blank
10-28	TOA (decimal point must be in column 15)
29-34	TOA uncertainty
35-45	Observing frequency (MHz)
46-55	DM correction (NOT IMPLEMENTED IN TEMPO2)
58-59	Observatory (2-characters)



Table 5: Commands that may be included in an arrival time file

Command	Meaning
EFAC <i>x</i>	Multiply uncertainties by <i>x</i>
T2EFAC -backend dfb <i>x</i>	Multiply TOA uncertainties with flag “-backend dfb” by <i>x</i>
GLOBAL_EFAC <i>x</i>	Multiply all TOA uncertainties by <i>x</i> . If for some or all of the TOAs an “EFAC <i>y</i> ” is present as well, then those TOAs will be multiplied by $x \times y$ .
EMAX <i>x</i>	Ignore TOAs with uncertainties greater than <i>x</i> $\mu$ s
EMIN <i>x</i>	Ignore TOAs with uncertainties less than <i>x</i> $\mu$ s
EFLOOR <i>x</i>	Put uncertainties of less than <i>x</i> $\mu$ s to <i>x</i> $\mu$ s
END	Ignore all remaining lines in the arrival time file
EQUAD <i>x</i>	Additional uncertainty in us, added in quadrature
FMAX <i>x</i>	Ignore TOAs at frequencies greater than <i>x</i>
FMIN <i>x</i>	Ignore TOAs at frequencies less than <i>x</i>
INCLUDE <i>x</i>	Include the arrival times in file <i>x</i>
INFO <i>x</i>	Identify all following points with a given highlighting code
MODE	MODE 0 (default) implies that the TOA error is not taken into account during the fitting procedure. MODE 1 uses the uncertainties (see section on fitting)
NOSKIP	End of SKIP statement
PHASE <i>x</i>	Add phase jump
SIGMA <i>x</i>	Set uncertainties of following TOAs to <i>x</i> $\mu$ s
SKIP	Skip all lines until NOSKIP is read
TIME <i>x</i>	Add <i>x</i> seconds to following TOAs
TRACK <i>x</i>	Tracks phase wrap-arounds if time step is less than <i>x</i> days

## 6.1 Setting dispersion measures

In some situations the dispersion measure is measured accurately for every observation. This information can be provided to tempo2 by adding `-dm DMval` on the end of each arrival time line in the `.TIM` file. If this is not present then the dispersion measure in the `.PAR` file is used instead.

## 6.2 Jodrell Bank Observatory binary format

It is also possible to provide pulse arrival times in the Jodrell Bank format binary file that containing barycentric arrival times. For instance a typical usage would be:

```
tempo2 -f psrav.eph psrav.bat -jbo -delete psrav.del
```

## 7 Atmospheric propagation delays

(to be written)

## 8 Command-line arguments

TEMPO2 has a number of options that can be controlled from the command line. Options include:

- `-debug`, provides output useful for identifying problems with the software
- `-delete fname`, delete the observations listed in the file `fname` by their site-arrival times or observations name.
- `-clock x`, define the clock in conversion to TT as X
- `-epoch x`, set the epoch of the parameter file to be MJD *x*

- `-f x.par y.tim` specifies the parameter (.par) and arrival time (.tim) files to use for subsequent processing. If only a .tim file is present (without -f option) then the parameter file will be assumed to be y.par.
- `-filter x`, filters the set of observations (see below).
- `-fit x`, turn on fitting for parameter 'x' (this command-line option can be repeated for multiple parameters)
- `-gr x`, use the 'x' graphical interface
- `-h`, displays help information
- `-jbo`, expect input in the Jodrell Bank psrav.bat format
- `-list` lists information about the residuals, time delays etc. that have been used by TEMPO2.
- `-machine x` define the processor being used
- `-model ...`
- `-name x` define the pulsar name to be 'x' - ignoring what is in the parameter file.
- `-newpar` produces a new .par file from the fitted parameters
- `-nobs` re-defines the maximum number of observations to be stored simultaneously in memory.
- `-nofit` switch off the fitting algorithms
- `-noWarnings` switch off displaying warnings.
- `-npsr` re-defines the maximum number of pulsars to be stored simultaneously in memory.
- `-output name` uses the 'name' plugin instead of displaying the standard TEMPO2 output.
- `-polyco` runs TEMPO2 in predictive mode.
- `-pre` only calculates the pre-fit residuals
- `-residuals` outputs the residuals to a file called "residuals.dat".
- `-set X a`, set the parameter X to the value 'a' (ignoring the value in the parameter file)
- `-tempo1` enables tempo1 compatibility mode

## 8.1 The tempo1 emulation mode

In the TEMPO1 compatibility mode the following parameters are automatically set (these can also be set manually in the parameter file):

- UNITS TDB
- TIMEEPH FB90.TIMEEPH
- DILATEFREQ N
- PLANET\_SHAPIRO N
- T2CMETHOD TEMPO
- CORRECT\_TROPOSPHERE N
- NE\_SW 9.961
- ECLIPTIC\_OBLIQUITY 84381.412

## 8.2 Filtering

It is often useful to be able to filter the observation file before processing. For instance, an observer may use multiple back-end systems or observatories. These can be defined using "flags":

```
../archives/n2003-01-10-17:07:45.FTp 1340.749 52649.7257990374280 0.67 7 -i nCPSR2_20
../archives/w040206_150608.cFTp 3067.999 53041.6404217220093 34.43 7 -i WBC_10
../archives/w040206_160503.cFTp 3067.999 53041.6813940348552 42.73 7 -i WBC_10
../archives/w040207_133403.cFTp 1421.439 53042.5667819871316 6.92 7 -i WBC_20
../archives/w040207_134155.cFTp 1417.998 53042.5819726383262 4.59 7 -i WBC_20
```

If all the WBC\_10 and nCPSR2.20 observations should be ignored then use

```
tempo2 -f mypar.par mytim.tim -filter "-i nCPSR2_20 -i WBC_10"
```

### 8.3 Aliases

Occasionally the user must repeatedly run the same `tempo2` command which contains multiple arguments. This can be simplified using aliases. Aliases are placed in a file called `$TEMPO2/alias.dat` as follows:

```
-jodrell -f psrav.eph psrav.bat -jbo -del psrav.del  
-quick -newpar -gr plk
```

The user can then type

```
tempo2 -jodrell -gr plk
```

instead of

```
tempo2 -f psrav.eph psrav.bat -jbo -del psrav.del -gr plk
```

## 9 The bootstrap fitting algorithm

Bootstrapping fitting methods can produce more realistic parameter values and uncertainties when significant correlations between parameters are present. The bootstrapping method implemented in `TEMPO2` estimates the uncertainty on a parameter by 1) randomly selecting observations to produce a new dataset of the same length as the original (the observations are selected with replacement; ie. in the new dataset some of the original observations will be omitted while others will be replicated), 2) recalculating the parameter and 3) repeating as many times as possible.

## 10 Predictive mode

For on-line and off-line folding of pulsar data `TEMPO2` can produce a set of polynomial coefficients to predict the pulse phase at any given time. After a standard `TEMPO2` fit has been carried out, the parameters will include the following

- `tzrmjd`: a reference TOA calculated as the first site-arrival-time with an MJD greater than `PEPOCH`. The residual after fitting is subsequently removed from `tzrmjd` to produce a site-arrival-time that produces zero residual,
- `tzrfreq`: The frequency of the arrival time corresponding to `tzrmjd`,
- `tzrsite`: The telescope site code corresponding to `tzrmjd`.

`TEMPO2` can be used to produce predictions in a new format or in a new format.

### 10.1 Tempo2 format

*(To be written)*

### 10.2 Tempo1 format

Example usage:

```
tempo2 -f 0437-4715.par -polyco "53000 53001 300 12 8 pks 1400.0" -tempo1
```

will request that `TEMPO2` makes a prediction for J0437-4715 from MJDs 53000 to 53001 with each divided into segments each of `nspan = 300` minutes. The number of coefficients for use in the fitting, `ncoeff = 12`. The maximum hour angle range for the prediction, `maxha = 8`. The observatory site for the prediction is `site_code = pks` at an observing frequency of `freq = 1400.0`. Therefore, the format used in defining the prediction is

```
-polyco "mjd1 mjd2 nspan ncoeff maxha site_code freq"
```

(note the use of the quote marks around the parameters). TEMPO2 should produce a file (POLYCO\_NEW.DAT) that takes the form of TEMPO1. For instance,

```
0437-4715 27-Dec-03 123000.00 53000.52083333330 2.646966 0.269 -7.575
27109839749.228820 173.687948999098 atca 960 12 600.000 0.5593 0.1742
-5.42287549188530393e-08 1.29656588048219029e-01 -8.73567925939434345e-05
-3.77430102018454689e-08 3.61675384452316159e-11 -5.30580886494090104e-16
-2.11398949708346726e-17 7.50678735870179551e-22 7.17204723641224429e-24
-2.20834139169245321e-28 -1.38834331804078410e-30 6.03780300115982370e-35
```

TEMPO2 also produces a file (NEWPOLYCO.DAT) which has the same parameters, but listed to more decimal places. Each parameter is listed on an individual line.

**note: the tempo1 software switches off clock corrections in predictive mode. To emulate this the CLK flag in the parameter file should be set to CLK UNCORR.**

## 11 Global parameters

(To be written)

## 12 Output formats

The default output format provides the pre- and post-fit rms residuals, the number of points in the fit and, if a weighted fit has been carried out, the reduced  $\chi^2$ -value of the fit. For each parameter, the values of the pre- and post-fit parameters are listed alongside the uncertainty in the post-fit value and the difference between the pre- and post-fit values. A flag indicates whether the parameter was included in the fit. For binary systems, the default output format also provides details on the binary model and lists, if possible, the mass function, minimum, median and maximum companion masses, the total system mass and the inclination angle. An example is given below,

Results for PSR J0437-4715

RMS pre-fit residual = 6.895 (us), RMS post-fit residual = 0.096 (us)  
Number of points in fit = 367

PARAMETER	Pre-fit	Post-fit	Uncertainty	Difference	Fit
RAJ (rad)	1.2097885336826	1.20978853474746	2.1206e-11	1.0649e-09	Y
RAJ (hms)	04:37:15.7865	04:37:15.7865146	2.916e-07	1.4643e-05	
DECJ (rad)	-0.824709094076948	-0.824709094473468	1.5157e-11	-3.9652e-10	Y
DECJ (dms)	-47:15:08.4615	-47:15:08.46158	3.1264e-06	-8.1788e-05	
FO (s <sup>-1</sup> )	173.68794630603	173.687946306032	9.5579e-15	2.3306e-12	Y
F1 (s <sup>-2</sup> )	-1.728e-15	-1.72831367406148e-15	2.3325e-22	-3.1367e-19	Y
PEPOCH (MJD)	51194.0001248168	51194.0001248168	0	0	N
POSEPOCH (MJD)	51194.0001248168	51194.0001248168	0	0	N
DMEPOCH (MJD)	51194	51194	0	0	N
DM (cm <sup>-3</sup> pc)	2.64690012312213	2.64690012312213	0	0	N
PMRA (mas/yr)	121.43799811708	121.43799811708	0	0	N
PMDEC (mas/yr)	-71.4379988923397	-71.4379988923397	0	0	N
PX (mas)	7.19	7.19	0	0	N
SINI	0.6788	0.6788	0	0	N
PB (d)	5.741046089	5.74104608900813	1.0437e-11	8.1286e-12	Y
TO (MJD)	51194.6240248265	51194.6240248265	0	0	N
A1 (lt-s)	3.36669162220122	3.36669162220122	0	0	N
OM (deg)	1.2	1.2	0	0	N
ECC	1.9186e-05	1.9186e-05	0	0	N
PBDOT (10 <sup>-12</sup> )	3.64	3.64	0	0	N
OMDOT (deg/yr)	0.0159999997519168	0.0159999997519168	0	0	N
M2	0.236	0.236	0	0	N
START (MJD)	50640.9281162413	49350.5129309451	0	-1290.4	N
FINISH (MJD)	52088.8971386924	53000.5197060478	0	911.62	N
TRACK (MJD)	0	0	0	0	N
TZRMJD	51204.6438924841	51204.6438924841	0	0	N

TZRFRQ (MHz)	1413.39997808495	1413.39997808495	0	0	N
--------------	------------------	------------------	---	---	---

---

```

Binary model: DD
Mass function          = 0.001243113190 +- 0.000000047151 solar masses
Minimum companion mass = 0.1403 solar masses
Median companion mass  = 0.1637 solar masses
Maximum companion mass = 0.3493 solar masses
MTOT derived from sin i and m2 = 1.818606841374
Inclination angle (deg) = 42.74994182955 (+ 0 - 0)

```

Other output formats have been developed and can be used as plug-ins, e.g.

```
tempo2 -output NAME -f ...
```

where NAME is the name of the output format. The plug-ins provided with TEMPO2 are listed below.

## 12.1 general

The GENERAL output format allows the user to determine, in a general manner, the presentation of the fitted parameters. For instance,

```
tempo2 -output general -s "\{NORAD\}{ALL\_f}{TAB 20}\& {ALL\_p} {TAB 50}\\\\\\n" -f psr.par psr.tim
```

will produce output in a  $\LaTeX$  format:

```

RAJ (rad)           & 18:02:05.33496(12)      \\
DECJ (rad)          & -21:24:03.72(6)                        \\
FO (s^-1)           & 79.066424253435(18)                       \\
F1 (s^-2)           & -4.574E-16(13)                             \\
PEPOCH (MJD)        & 52855                                        \\
POSEPOCH (MJD)      & 52855                                        \\
DMEPOCH (MJD)       & 52855                                        \\
DM (cm^-3 pc)      & 149.666066                                  \\
PB (d)              & 0.69888924320(20)                          \\
A1 (lt-s)           & 3.718853847                                 \\
TASC (MJD)          & 52595.795078543                            \\
EPS1                & 1.0412e-06                                  \\
EPS2                & 2.3924e-06                                  \\
START (MJD)         & 52605.162                                   \\
FINISH (MJD)        & 53565.334                                   \\
TRACK (MJD)         & 0                                             \\
TZRMJD              & 52883.440856822                            \\
TZRFRQ (MHz)       & 1390                                         \\

```

Options are also available to increase the uncertainties by a given factor. To change the parameters listed, to change the string displayed if a parameter has not been set and to change the number of decimal places output.

## 12.2 general2

This output format is similar to the GENERAL output described above. However, this output provides access to values calculated for each observation:

```

{sat}   site-arrival-times
{bat}   barycentric arrival times
{clock0 -> clock4} various clock correction values
{shapiro} the solar Shapiro delay
{shapiroJ} the Shapiro delay due to Juptier
{shapiroS} the Shapiro delay due to Saturn
{shapiroV} the Shapiro delay due to Venus
{shapiroU} the Shapiro delay due to Uranus
{shapiroN} the Shapiro delay due to Neptune
{tropo} the tropospheric delay
{roemer} the solar system Roemer delay
{tt}    correction to TT

```

```

{tt2tb} correction from TT to TB
{earth_ssb} magnitude of vector from Earth to barycentre
{earth_ssb1} magnitude of 'x' component from Earth to barycentre
{earth_ssb2} magnitude of 'y' component from Earth to barycentre
{earth_ssb3} magnitude of 'z' component from Earth to barycentre
{sun_earth1} magnitude of 'x' component from Sun to Earth
{sun_earth2} magnitude of 'y' component from Sun to Earth
{sun_earth3} magnitude of 'z' component from Sun to Earth
{ism}      interstellar medium dispersion delay
{elev}     Source elevation
{npulse}   pulse number
{clock}    complete clock corrections to TT
{ipm}      interplanetary medium dispersion delay
{freq}     observing frequency
{pre}      prefit timing residual in seconds
{pre_phase} prefit timing residual in phase
{post}     postfit timing residual in seconds
{post_phase} postfit timing residual in phase
{err}      TOA error
{binphase} binary phase

```

For example, to display the barycentric arrival times, the Shapiro delay due to Jupiter and the post-fit residual, use

```
tempo2 -output general2 -f par.par tim.tim -s "{bat} {shapiroJ} {post}\n"
```

### 12.3 list

The LIST output lists the basic parameters that are being used in the TEMPO2 calculations. For example: `tempo2 -output list -f msp1.par msp1.tim` will list

1. the site arrival times
2. pre- and post-fit residuals
3. clock correction to UTC
4. barycentric arrival time
5. other clock corrections
6. the solar system Shapiro delay
7. the dispersion measure time delays due to the interstellar and planetary medium
8. the Roemer delay
9. ephemeris values for the position of the Sun with respect to the solar system barycentre
10. ephemeris values for the earth-moon barycentre with respect to the solar system barycentre
11. ephemeris values for the moon with respect to the Earth
12. the position of the observatory with respect to the centre of the Earth
13. a 3-vector pointing at the pulsar from the observatory

### 12.4 stats

Provides information on the residuals and observations such as the median TOA uncertainty for different observing frequencies.

For example: `tempo2 -output stats -f gh.par gh.tim` will give

```

Number of TOAs in fit = 227
Residual = 0.000985 (us)
Earliest arrival time at   MJD 53041.318518
Most recent arrival time at MJD 53248.801042
Span = 207.5 days
Observing frequencies = 1429.29   3099.00

```

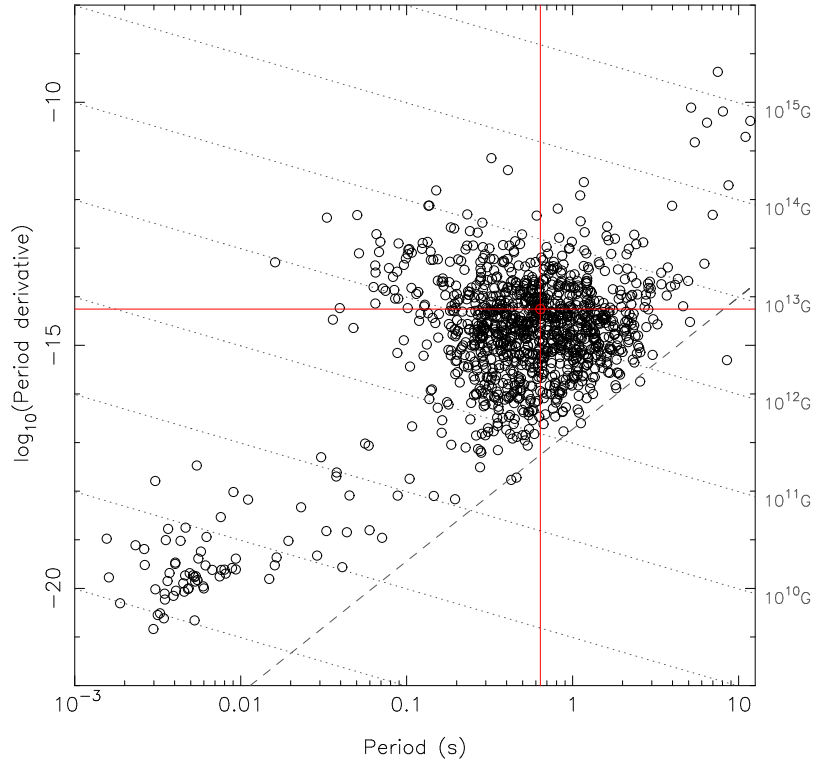


Figure 2: An example of the ‘basic’ graphical interface. A  $P - \dot{P}$  diagram is produced with the pulsar being analysed highlighted.

Freq. (MHz)	N	Minimum Uncertainty (us)	Maximum Uncertainty (us)	Mean Uncertainty (us)	Median Uncertainty (us)	RMS (us)
All	226	0.17	2.70	1.23	0.30	9.8452e-12
1429.29	95	0.97	2.70	2.54	2.57	
3099.00	131	0.17	1.27	0.29	0.28	

## 13 Graphical interfaces

### 13.1 basic

Plots a  $P - \dot{P}$  diagram and calculates the pulsar’s characteristic age (see Figure 2 for an example: `tempo2 -gr basic -f my`). The plugin also provides information such as the pulsar’s spin period, its derivative and the characteristic age  $\tau = P/(2\dot{P})$ . It is possible to display only pulsars within a certain declination range and to list all the pulsars currently being displayed. This allows the user to obtain a list of pulsars that have similar rotational characteristics to the pulsar currently being analysed by TEMPO2.

### 13.2 calcDM

*(To be written)*

### 13.3 chisq

This graphical interface is used to create analyse the  $\chi^2$  value of the fit or the post-fit rms residual for a one- or two-dimensional grid of parameter values. The interface repeatedly carries out the fitting routines for specific values of one or two parameters and displays the results as a graph, contour or gray-scale plot.

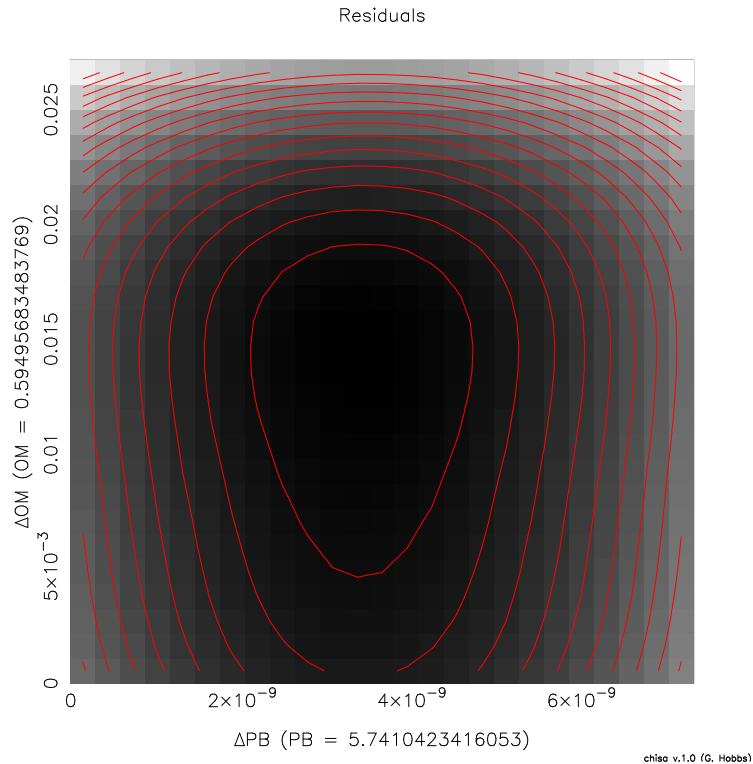


Figure 3: Example of the `chiso` plug-in for determining the most likely values of the orbital period, PB, and longitude of periastron, OM.

This plugin is based on the “`parmap`” program developed by Swinburne University.

### 13.4 compare

Graphical interface used to compare the residuals obtained using two different parameter (`.par`) files. For instance, different solar system ephemeris could be defined in the parameter files and the results compared using this interface.

### 13.5 compareRes

*(To be written)*

### 13.6 delays

This interface allows the user to inspect the clock corrections and propagation time delays that TEMPO2 has applied in order to obtain barycentric arrival times. See Figure 4 (`tempo2 -gr delays -f mypar.par mytim.tim`). The following key strokes are possible

<code>q</code>	<code>quit</code>
<code>y-1</code> ('y' followed by '1')	<code>plot first clock correction</code>
<code>y-2</code> ('y' followed by '2')	<code>plot second clock correction</code>
<code>y-3</code>	<code>plot third clock correction</code>
<code>y-4</code>	<code>plot fourth clock correction</code>
<code>y-5</code>	<code>plot fifth clock correction</code>
<code>y-6</code>	<code>plot UT1</code>
<code>y-7</code>	<code>plot Shapiro delay due to Sun</code>
<code>y-8</code>	<code>plot dispersion delay in solar system</code>
<code>y-9</code>	<code>plot dispersion delay in ISM</code>
<code>y-0</code>	<code>plot Roemer delay</code>
<code>y-a</code>	<code>plot pre-fit residuals</code>
<code>y-b</code>	<code>plot post-fit residuals</code>
<code>y-c</code>	<code>plot Shapiro delay due to Jupiter</code>



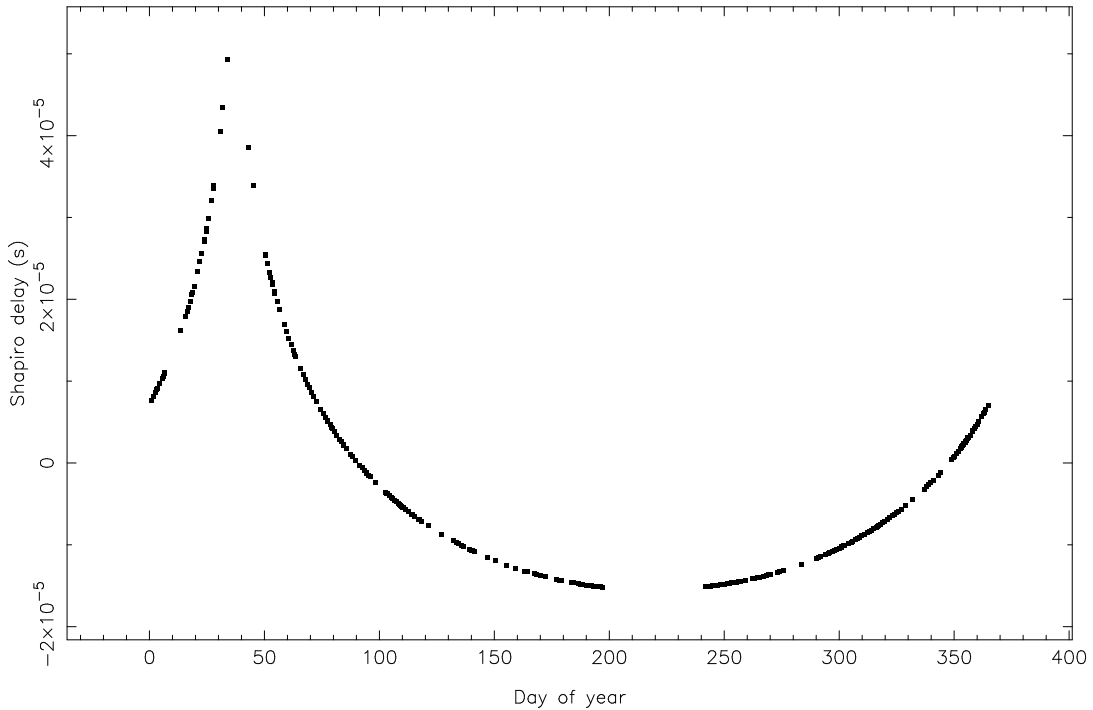


Figure 4: The solar system Shapiro delay for PSR J1810–2005 shown using the DELAYS graphical interface plug-in

```

y-d          plot Shapiro delay due to Saturn
y-e          plot Shapiro delay due to Uranus
y-f          plot Shapiro delay due to Neptune
y-g          plot total planetary Shapiro delay
y-h          plot tropospheric propagation delay
1           plot TOA number on x-axis
2           plot MJD day on x-axis
3           plot observing frequency on x-axis
4           plot day-of-year on x-axis
5           plot binary phase on x-axis
6           plot year on x-axis
x           redo fit
h           display this help
g           change the graphics terminal
right mouse button  delete TOA
left mouse button   identify TOA
c               clear highlights
s               start zoom region
f               finish zoom region
t               simulate new arrival times
u               unzoom

```

### 13.7 errors

*(To be written)*

### 13.8 fake

It is often necessary to simulate timing residuals that would be expected to be measured for a particular pulsar. TEMPO2 contains a plug-in package to do this. FAKE automatically generates a set of TOAs between dates specified by the user. The arrival times are defined to be at transit and so only the pulsar's right ascension is used in this calculation. The pre-fit timing residuals are then formed using a parameter file (containing the simulated pulsar's position, spin parameters, binary parameters etc.) and

subtracted from the original simulated arrival times. This procedure is iterated until the residuals are zero. These arrival times can subsequently be modified by 1) the addition of Gaussian noise and 2) the addition of simulated timing noise. Upon running the fake plugin, the user is asked to provide

- The number of days between observations
- The number of observations on a given day
- The maximum absolute hour angle allowed
- Whether the user required random or regular hour angle coverage
- The initial MJD for the simulated TOAs
- The final MJD for the simulated TOAs
- The rms of Gaussian noise to be added to the TOAs
- Whether red noise should be added to simulate timing noise

If the red noise option is selected, the user is also requested to provide

- The power law index for the red noise
- The power law amplitude
- A random number seed
- Whether the residuals should be smoothed
- Whether a cubic should be added to the residuals

The red noise is simulated as a shot-noise process obtained by summing many sinusoids with random phase, but with amplitudes given by the requested power-law spectrum.

The following example simulates a pulsar with a large proper motion:

1. Produce a parameter file similar to (called testfake.par)

```

PSRJ           J1730-2304
RAJ            17:30:21.6483
DECJ          -23:04:31.4
POSEPOCH      51500.0
PMRA          200
PEPOCH        51500.0
F0            123.110289179797
F1            -3.0631E-16
DM            9.611
CLK UTC(NIST)
EPHEM DE200

```

2. Run `tempo2 -gr fake testfake.par`
3. Follow on screen instructions
4. The fake plug-in will produce an arrival time file called `testfake.simulate`
5. In the parameter file change the PMRA back to zero
6. Run `tempo2 -gr plk -f testfake.par testfake.simulate`

The result should be similar to that shown in Figure 5.

*note: this code was based on the fake software originally developed by Duncan Lorimer and updated by Simon Johnston*

## 13.9 gorilla

*(To be written)*

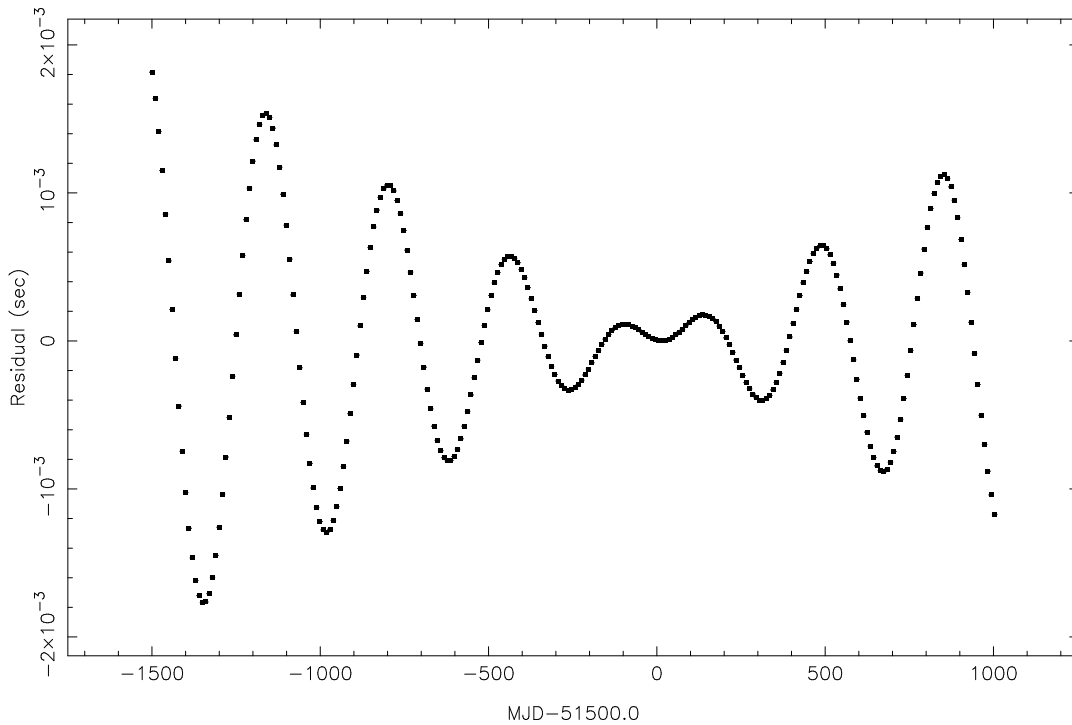


Figure 5: Simulating pulsar timing residuals for a pulsar with a proper motion in right ascension of 200 mas/yr using the FAKE plugin.

### 13.10 plk

PLK provides the user with a graphical interface that plots pre-fit and post-fit timing residuals versus parameters such as day, TOA number, binary phase or observing frequency. It is based on the plk standalone package written for the original TEMPO, but has been significantly enhanced. The profiles corresponding to TOAs may be viewed and the TOAs deleted or identified using mouse clicks. Phase jumps can easily be added and the data refit in order to improve the timing model. A simple menu system allows the user to change the fitted parameters and to produce new arrival time and parameter files. This graphical interface only works in single-pulsar mode. An example of the output is shown in Figure 6 (usage: `tempo2 -gr plk -f mypar.par mytim.tim`).

Options available are:

```

b          Bin TOAs within certain time bin
c          Change fitting parameters
C          run unix command with filenames for highlighted observations
ctrl-c    Toggle between period epoch and centre for the reference epoch
d (or right mouse) delete point
D (or middle mouse) view profile
ctrl-D    delete highlighted points
e         multiply all TOA errors by given amount
f         finish of zoom section
F         run FITWAVES
ctrl-F    remove FITWAVES curve from residuals
g         change graphics device
G         change gridding on graphics device
h         this help file
H         highlight points with specific flag in .tim file
i (or left mouse) identify point
j         draw line between each points
l         list all data points in zoomed region
L         add label to plot
ctrl-l    add line to plot
m         measure distance between two points
M         toggle removing mean from the residuals

```

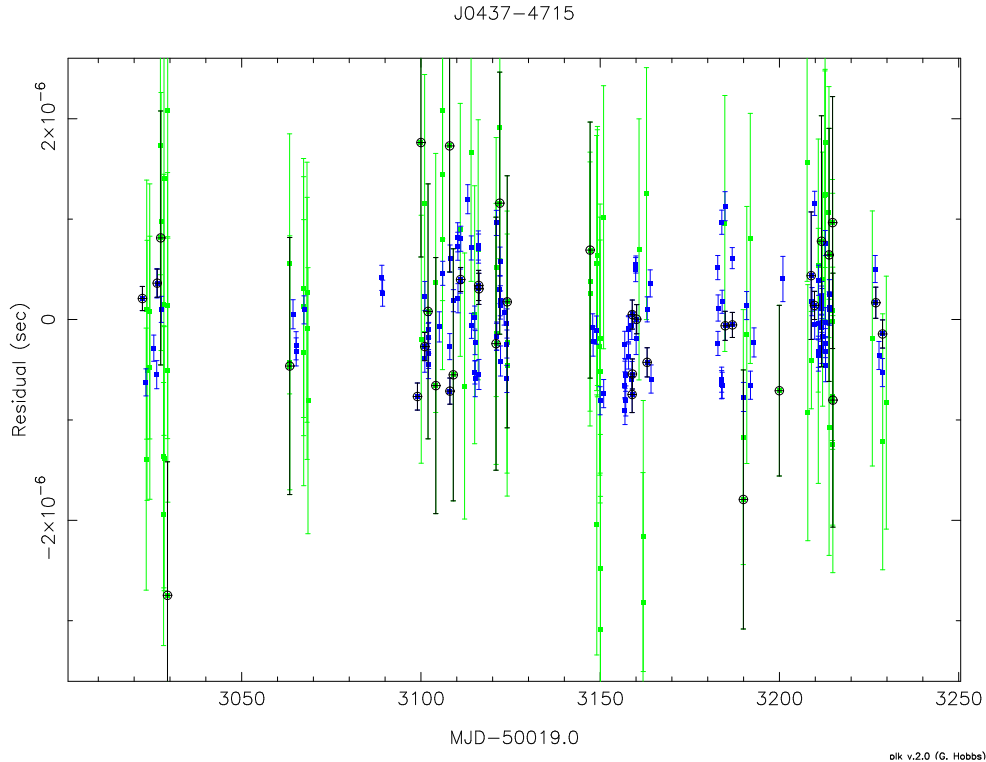


Figure 6: An example of the plk graphical interface in use. The post-fit residuals for PSR J0437-4715 are plotted with 20 observations shown in green and 10 cm observations in red. The white circles and black lines indicate arrival times between orbital phases 0.4 and 0.6.

```

ctrl-m    toggle menu bar
o         obtain/highlight all points currently in plot
p         Change model parameter values
P         write new .par file
ctrl-P    Toggle fitting versus pulse phase
q         quit
r         Reset (reload .par and .tim file)
s         start of zoom section
S         save a new .tim file
ctrl-S    Overplot Shapiro delay
u         unzoom
U         unhighlight selected points
v         view profiles for highlighted points
V         define the user parameter
ctrl-V    for pre-fit plotting, decompose the timing model fits
w         toggle fitting using weights
x         redo fit using post-fit parameters
ctrl-X    place periodic marks on the x-scale
y         Rescale y-axis only
z         Zoom using mouse
+         add positive phase jump
-         add negative phase jump
ctrl-=    add period to add residuals above cursor
<         in zoom mode include previous observation
>         in zoom mode include next observation
1         plot pre-fit residuals vs date
2         plot post-fit residuals vs date
3         plot pre-fit residuals vs orbital phase
4         plot post-fit residuals vs orbital phase
5         plot pre-fit residuals serially
6         plot post-fit residuals serially
7         plot pre-fit residuals vs day of year
8         plot post-fit residuals vs day of year
9         plot pre-fit residuals vs frequency
a         plot post-fit residuals vs frequency

```

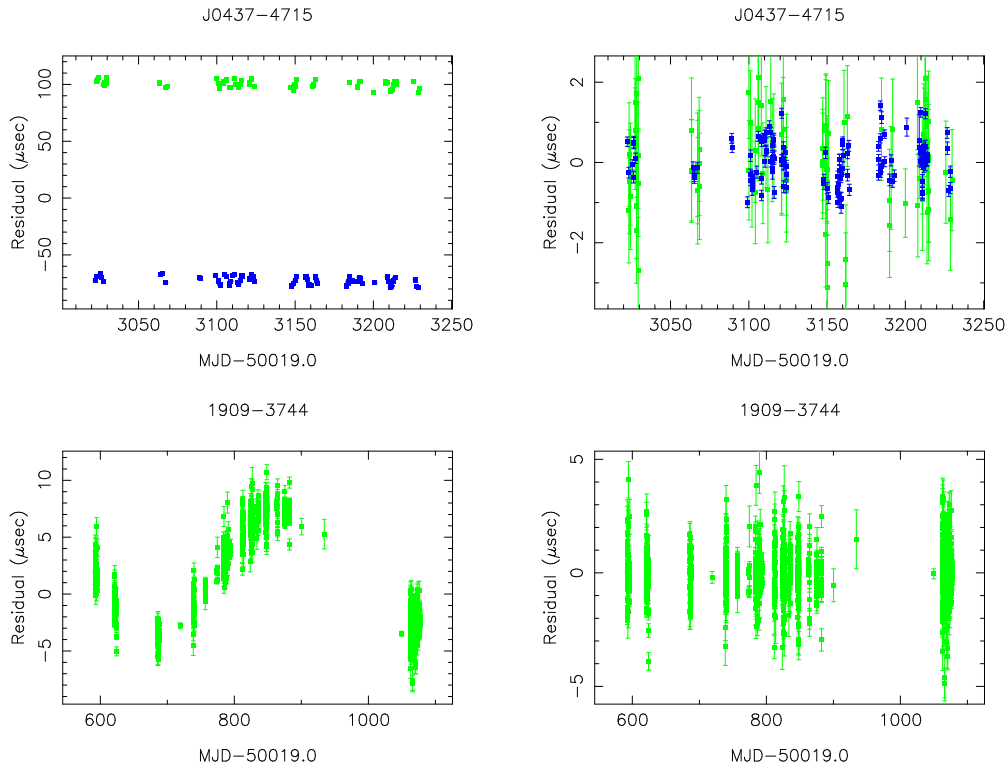


Figure 7: Example of the SPLK graphical interface plug-in for showing the pre- and post-fit timing residuals for multiple pulsars simultaneously.

```
!      plot pre-fit  residuals vs TOA error
@      plot post-fit residuals vs TOA error
#      plot pre-fit  residuals vs user values
$      plot post-fit residuals vs user values
%      plot pre-fit  residuals vs year
~      plot post-fit residuals vs year
```

### 13.11 polytest

*(To be written)*

### 13.12 splk

A simple graphical interface that displays pre-fit and post-fit timing residuals for multiple pulsars (Figure 7). Key-strokes enable the user to plot all the pulsars using the same x and y axes (or to scale each axis separately) and the timing residuals can be output in an ASCII format to a file.

```
tempo2 -gr splk -f mypar1.par mytim1.tim -f mypar2.par mytim2.tim -f mypar3.par mytim3.tim
```

### 13.13 stridefit

The STRIDEFIT plugin is used to fit for a pulsar parameter (commonly DM) using small sections of data and plotting the variation in that parameter versus time. Example usage is:

```
tempo2 -gr stridefit -f psr.par psr.tim
```

The plugin will subsequently ask for the parameter for study (e.g. DM, PMRA, PMDEC ...). The STRIDEFIT works by running a window with a width entered, from a start time to an end time with a given time step. These can all be entered on the command line:

```
tempo2 -gr stridefit -f psr.par psr.tim -param DM -start 50000 -end 52000
      -width 300 -dt 100
```

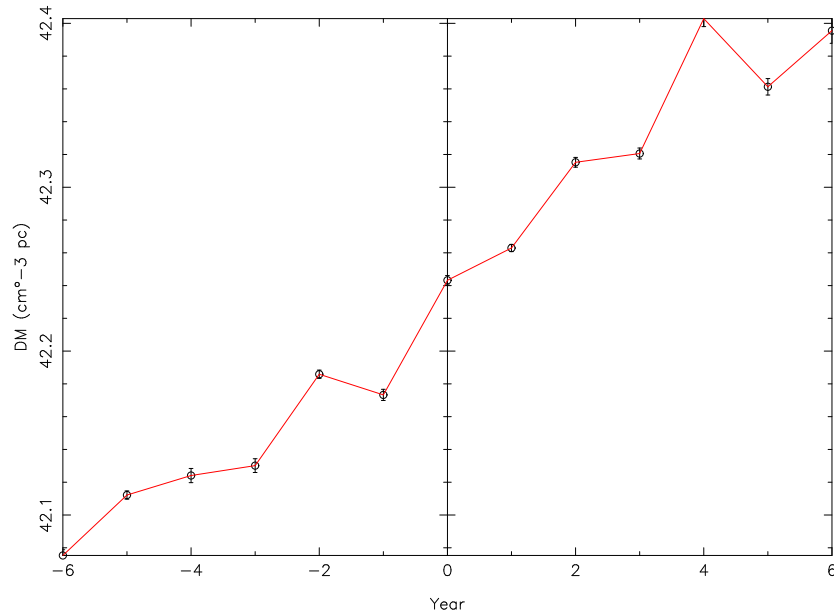


Figure 8: Example of the STRIDEFIT graphical interface plug-in which calculates the pulsar's dispersion measure as a function of time.

The STRIDEFIT plugin will produce a graph of the parameter versus time. Clicking on any point will list the points that were used in deriving the parameter. If there is reason to believe that any particular point is in error (for instance trying to obtain a DM value with only one observing frequency) then the point can be deleted using the right mouse button.

An example STRIDEFIT output is shown in Figure 8.

After plotting the following key strokes may be used

```

h   provides help
1   zero x-value for first point
2   centre points
3   toggle plotting day/year
u   rescale the axes
q   quit
f   fit a straight line to a region selected by the mouse
F   fit and remove a straight line
r   redo the stridefit
e   re-enter the MJD of the end point
s   re-enter the MJD of the start point
S   change the font-size for plotting
n   enter the minimum number of observations required for a point
g   set the graphical device
w   re-enter the width of the fit region
t   re-enter the time step
A (or left mouse) identify closest point
X (or right mouse) delete closest point
z   zoom using the mouse
l   list all the values
L   list all the values and output to file

```

### 13.14 transform

Transforms a TEMPO1 parameter file into a TEMPO2-compatible file.

```
tempo2 -gr transform tempo1.par tempo2.par [back]
```

where the `back` command is used to convert from TEMPO2 to TEMPO1.

## 14 Developing the software

The plug-in features of TEMPO2 implies that users can add to the functionality of TEMPO2 by producing their own plug-in. In this section, we first discuss how such plug-ins can be produced. We also provide a few notes for users who are interested in developing the actual main source code and become part of the TEMPO2 development team.

### 14.1 Creating a plug-in

Plug-ins are relatively easy to create in C or C++. These plug-ins may use other libraries such as PGPLOT, TCL/TK or QT. As described above, plug-ins can be created to change the output format or as a graphical interface.

#### 14.1.1 A new output format

When an “output-format” plug-in is used, the TEMPO2 software reads in the specified parameter and observation files, carries out the standard techniques to obtain pre- and post-fit timing residuals and their corresponding parameter values and then passes control to the output-format software to display the results. An example output format (in C) is as follows:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "tempo2.h" /* Essential for a tempo2 plugin */

extern "C" int tempoOutput(pulsar *psr,int npsr,char timFile[] [MAX_FILELEN],char parFile[] [MAX_FILELEN])
{
    int i;
    printf("Pulsar name = %s\n",psr[0].name);
    printf("Pulsar spin-frequency = %Lg\n",psr[0].param[param_f].val[0]);
    for (i=0;i<psr[0].nobs;i++)
        printf("Residual = %.14f\n", (double)psr[0].obsn[i].residual);
}
```

This plug-in displays the pulsar’s name (obtained from the parameter file), its post-fit spin-frequency and the post-fit timing residuals on the screen. The software is built around a STRUCTURE called PULSAR which is defined in TEMPO2.H. The most important definitions in this structure are

```
char *name;          /* The pulsar name */
char *binaryModel;  /* The binary model (e.g. T2/BT/ELL1 ...) */
double ne_sw;       /* The electron density at 1AU due to the solar wind */
int fitMode;        /* = 1 if fitting with errors, 0 = not fitting with errors */
int nobs;           /* Number of observations */
int nits;           /* Number of iterations used for the fit */
int ipm;            /* = 1 if the interplanetary DM correction is used, 0 otherwise */
```

An array of pulsars is defined and therefore, in order to display the name of the first three pulsars analysed:

```
printf("pulsar 1 = %s\n",psr[0].name);
printf("pulsar 2 = %s\n",psr[1].name);
printf("pulsar 3 = %s\n",psr[2].name);
```

The structure also includes an array of parameters (PARAM) which are defined as

```
param_raj,param_decj,param_f,param_pepoch,param_posepoch,
param_dmepoch,param_dm,param_pmra,param_pmdec,param_px,
param_sini,param_pb,param_t0,param_a1,param_om,param_pmrsv,
param_ecc,param_edot,param_xpbdot,param_pbdot,param_a1dot,
```

```

param_omdot,param_tasc,param_eps1,param_eps2,param_m2,param_gamma,
param_mtot,param_glep,param_glph,param_glf0,param_glf1,
param_glf0d,param_gltd,param_start,param_finish,param_track,param_bp,param_bpp,
param_tzrmjd,param_tzrfrq,param_fdd,param_dr,param_dtheta,param_tspan,
param_bpjep,param_bpjph,param_bpja1,param_bpjec,param_bpjom,param_bpjpb,
param_wave_om,param_kom,param_kin,param_shapmax,param_dth,param_a0,
param_b0,param_xomdot,param_afac,param_eps1dot,param_eps2dot,param_tres

```

For each parameter, the user can obtain

```

char **label;          /* Label about this parameter          */
char **shortlabel;    /* Label about this parameter without units */
longdouble *val;      /* Value of parameter                    */
longdouble *err;      /* Uncertainty on parameter value        */
int *fitFlag;         /* = 1 if fitting required                */
int *paramSet;        /* = 1 if parameter has been set         */
longdouble *prefit;   /* Pre-fit value of the parameter        */
longdouble *prefitErr; /* Pre-fit value of the uncertainty      */
int aSize;           /* Number of elements in the array for this parameter */

```

Each parameter is stored as an array with each element of the array typically storing time derivatives of the parameter. For instance, to obtain the value of the spin-frequency and its first two derivatives:

```

printf("values = %Lg %Lg %Lg\n",psr[0].param[param_f].val[0],
       psr[0].param[param_f].val[1],psr[0].param[param_f].val[2]);

```

(Note: 'C' requires the symbol '%Lg' or '%LF' to display values in 'long double' precision. The 'pulsar' structure also contains the set of observations and corresponding parameters. The observation structure contains (other less-common parameters are defined in the TEMPO2.H file)

```

longdouble sat;          /* Site arrival time                      */
longdouble bat;          /* Barycentric arrival time                */
int deleted;            /* = 1 if observation has been deleted     */
longdouble prefitResidual; /* Pre-fit residual                        */
longdouble residual;    /* residual                                 */
double freq;            /* Frequency of observation (in MHz)       */
double freqSSB;         /* Frequency of observation in barycentric frame (in Hz) */
double toaErr;          /* Error on TOA (in us)                   */
char fname[MAX_FILELEN]; /* Name of data file giving TOA           */
char telID[100];        /* Telescope ID                             */
longdouble correctionTT_TB; /* Correction to TDB/TCB                   */
double einsteinRate;    /* Derivative of correctionTT_TB          */
longdouble correctionTT_Teph; /* Correction to Teph                       */
longdouble correctionUT1; /* Correction from site TOA to UT1        */
double shapiroDelaySun; /* Shapiro Delay due to the Sun            */
double shapiroDelayJupiter; /* Shapiro Delay due to Jupiter          */
double shapiroDelaySaturn; /* Shapiro Delay due to Saturn            */
double shapiroDelayUranus; /* Shapiro Delay due to Uranus            */
double shapiroDelayNeptune; /* Shapiro Delay due to Neptune           */
double troposphericDelay; /* Delay due to neutral refraction in atmosphere */
double tdis1;           /* Interstellar dispersion measure delay   */
double tdis2;           /* Dispersion measure delay due to solar system */
longdouble roemer;     /* Roemer delay                            */

```

For instance, in order to display the site arrival-time, the barycentric arrival-time, and the Solar Shapiro delay for each observation:

```

for (i=0;i<psr[0].nobs;i++)
  printf("values = %Lf %Lf %f\n",psr[0].obsn[i].sat,psr[0].obsn[i].bat,psr[0].obsn[i].shapiroDelaySun);

```

### 14.1.2 A new graphical interface

When a graphical interface is used, TEMPO2 checks the command-line arguments, initialises the memory and then passes control to the graphical interface. The reading of the parameter and observation files, calculating barycentric arrival times, obtaining residuals, fitting and displaying the output must all be done within the graphical interface. The following TEMPO2 functions are commonly called:



```

readParfile(...)      /* Read the parameter file          */
readTimfile(...)      /* Read the observation file           */
preProcess(...)       /* Needs to be called after reading the
                        parameter and observation files */
formBatsAll(...)      /* Forms the barycentric arrival times */
formResiduals(...)    /* Forms the timing residuals          */
doFit(...)            /* Calls the fitting algorithms        */
textOutput(...)       /* The standard output display         */

```

An example of a simple interface would be

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"

using namespace std;

/* The main function called from the TEMPO2 package is 'graphicalInterface' */
/* Therefore this function is required in all plugins                          */
extern "C" int graphicalInterface(int argc, char *argv[], pulsar *psr, int *npsr)
{
    char parFile[MAX_PSR][MAX_FILELEN];
    char timFile[MAX_PSR][MAX_FILELEN];
    int i;
    double globalParameter;

    *npsr = 1; /* For a graphical interface that only shows results for one pulsar */

    printf("Graphical Interface: name\n");
    printf("Author:                author\n");
    printf("Version:                version number\n");

    /* Obtain the .par and the .tim file from the command line */
    if (argc==4) /* Only provided .tim name */
    {
        strcpy(timFile[0],argv[3]);
        strcpy(parFile[0],argv[3]);
        parFile[0][strlen(parFile[0])-3] = '\0';
        strcat(parFile[0],"par");
    }

    /* Obtain all parameters from the command line */
    for (i=2;i<argc;i++)
    {
        if (strcmp(argv[i],"-f")==0)
        {
            strcpy(parFile[0],argv[i+1]);
            strcpy(timFile[0],argv[i+2]);
        }
    }

    readParfile(psr,parFile,timFile,*npsr); /* Load the parameters          */
    readTimfile(psr,timFile,*npsr); /* Load the arrival times          */
    preProcess(psr,*npsr,argc,argv);

    for (i=0;i<2;i++) /* Do two iterations for pre- and post-fit residuals*/
    {
        formBatsAll(psr,*npsr); /* Form the barycentric arrival times */
        formResiduals(psr,*npsr,0.0); /* Form the residuals          */
        if (i==0) doFit(psr,*npsr,&globalParameter,0,0); /* Do the fitting          */
        else textOutput(psr,*npsr,globalParameter,0,0,0,""); /* Display the output */
    }

    /* Now you have the parameters and residuals */
    /* which can be displayed or plotted          */

    return 0;
}

```

}

## 14.2 The main source code

The TEMPO2 software can be obtained from the ATNF CVS repository `SOFT_ATNF/TEMPO2` (please contact members of the ATNF pulsar group for more information). To install type

```
> make install
```

The plug-ins provided with the source code are stored in `SOFT_ATNF/TEMPO2/PLUGIN` and the documentation in `SOFT_ATNF/TEMPO2/DOCUMENTATION`.

## 15 Tempo2 error and warning messages

Errors result from serious problems with the pulsar analysis which may not be able to complete at all. Warnings indicate possible problems in the analysis. Generally, the analysis will keep going.

### 15.1 Error messages

File handling errors

- ERROR [FILE1]: No .tim file given on command line
- ERROR [FILE2]: No .par file given on command line
- ERROR [FILE3]: Unable to open parfile X for pulsar Y
- ERROR [FILE4]: Unable to open timefile X
- ERROR [FILE5]: Unable to open file OBSERVATORY\_CLOCK\_2.UTC\_NIST
- ERROR [FILE6]: Unable to open the leap second file
- ERROR [FILE7]: Unable to open OBSYS.DAT

Parameter file errors

- ERROR [PAR1]: Have not set a period epoch in X

Clock errors

- ERROR: Too many lines in ut1 file -- increase array size in tai2tdb
- ERROR [CLK2] Must increase MAX\_CLKCORR
- ERROR [CLK3], must increase size of MAX\_LEAPSEC
- ERROR [CLK4]: Date X out of range of TDB-TDT table

Ephemeris errors

- ERROR [EPHEM1]: Ephemeris file X not loaded

### 15.2 Warning messages

Parameter file warnings

- WARNING [PAR2]: Have not set a position epoch in X. The period epoch will be used instead

Clock warnings

- Warning [CLK1]: MJD later than last entry in time.dat. Imprecise clock corrections will be applied

## 16 Common questions

Our web-page provides a feedback form that allows the user to alert the software developers of any problems in the software.

- 1. How do I cite tempo2 in my publication?** For basic TEMPO2 usage, please give a reference to Hobbs, Edwards & Manchester (2006; submitted to MNRAS). For details of the timing model please refer to Edwards, Hobbs & Manchester (2006; in preparation).
- 2. How do I increase the number of observations that can be loaded in tempo2?** Use “-NOBS X” on the command line where X is the number of observations required (the current default is 10000 observations).

## 17 In progress

TEMPO2 is continually being developed. We are currently developing the following

- A set of plug-ins to simulate the effects of gravitational wave sources on pulsar timing residuals and to place limits on the existence of any such gravitational wave sources in real pulsar timing data.

## 18 Acknowledgements

The TEMPO2 package is based on the original TEMPO Fortran code. This software was developed over many years by multiple authors including J. Taylor, R. Manchester, D. Nice, W. Peters, J. Weisberg, A. Irwin and N. Wex. We thank the Australian pulsar community, J. Weisberg, D. Lorimer and M. Kramer for suggestions and comments on TEMPO2.