Tempo2

Generated by Doxygen 1.8.10

Thu Aug 11 2016 13:57:23

Contents

Chapter 1

Main Page

- User Guide
- Developer Guide
- Directory structure

Chapter 2

User Guide

2.1 Tempo2 User Manual

2.1.1 About tempo2

Tempo2 is a pulsar timing package, based on the old fortran tempo code to address some shortcomings in that code for high precision pulsar timing. Over the years tempo2 has been expanded my many developers, and has grown to become the premier package for all kinds of pulsar timing experiments.

For more details on pulsar timing in general, you may wish to read the Tempo2 paper series:

- I. An overview `http://adsabs.harvard.edu/abs/2006MNRAS.369..655H`
- II. The timing model and precision estimates `http://adsabs.harvard.edu/abs/2006MNRAS.372.1549E`
- III. Gravitational wave simulation `http://adsabs.harvard.edu/abs/2009MNRAS.394.1945H`

There is also a lot of useful information on the tempo2 wiki, `http://www.atnf.csiro.au/research/pulsar/tempo2/in` `php?n=Main.HomePage`. Some of the details are outdated as of 2015, but the general principles are sound. **The wiki is the best place for tutorials and basic introduction to tempo2.**

2.1.2 Terminology and basic usage

This documentation will focus on providing some basic overview of the many functions of tempo2 and is mostly intended as a reference for those who have mastered the basics. However, for completeness, here we will cover the most basic functions of tempo2.

Tempo2 brings together time-of-arival measurements (ToAs), stored in a `.tim` file, and a pulsar ephemeris stored in a `.par` file to produce the difference between the pulsar ephemeris model and the actual arrival times. This step is generally known as "forming residuals", and depends on having accurate models of the Earth ephemeris and of the clocks used to measure the ToAs. If all is well, these differences will be consistent with the uncertanty in the measurements. This is not generally the case, therefore the second part of tempo2 is a fitting routine that attempts to update the model parameters to get the best-fit model.

The basic usage of tempo2 is to feed in a `.par` and a `.tim` file, form residuals, do the fit and write out the best-fit parameters.

```
tempo2 -f example1.par example1.tim -newpar
```

This will write out `new.par` file with the updated parameters, as well as printing to the console the pre and post-fit parameters. Note any warnings that are printed. One of the side-effects of tempo2 is that it sometimes prints a lot of warnings, some you can ignore and some that you can't, so you have to read them!

If you compiled the `pgplot` plugins, you can run the graphical interface `plk`

```
tempo2 -gr plk -f example1.par example1.tim
```

Running plugins

There are many, many plugins. Some plugins are better supported than others. To get a list of the plugins you have installed try `tempo2 -h`. The majority of plugins are "graphical" plugins, even if they do not use graphics. This is to do with the way that the plugin is called, rather than anything to do with it being graphical. Graphical plugins are run with the `-gr` option. A few other types of plugins exist:

- `-gr <plugin_name>` for so-called "graphical" plugins. This is most plugins.
- `-output <plugin_name>` for "output" plugins, like `general` and `general2`
- `-fitfunc <plugin_name>` for alternative fit routines. This is likely to be removed in a future release.
- `-select <plugin_name>` for ToA filtering plugins.

You may have to review the source code if you can't find documentation for the plugin you desire. See the Plugin Documentation for more details on the avaliable plugins.

Chapter 3

Core Developers

Tempo2 development team

Tempo2 was origianaly written by George Hobbs and Rusell Edwards.

Core package maintainers

- George Hobbs [GH]george.hobbs@csiro.au
  - Core tempo2 development.
  - Gravitational wave codes.
  - Binary models.
- Michael Keith [MJK]mkeith@pulsarastronomy.net
  - C++ code maintainence.
  - Linear algebra and least-squares algorithms.
  - Build system maintainence.
  - Unit testing.

Active contributors

- Joris Verbiest
- Lindley Lentati
- Ryan Shannon
- Paul Demorest
- Lucas Guillemot
- Stefan Oslowski
- Willem van Straten
- Rutger van Haasteren
- Anne Archibald

Past Contributors

- Russell Edwards
- Aiden Hotan
- Ankur Chaudhary
- Ingrid Stairs

Chapter 4

Developer Guide

4.1 Tempo2 Developer Guide

4.1.1 About this guide

This guide has been developed to encourage development of tempo2, and to improve the consistency between developers. The majority of this guide has been written by MJK, although all are welcome to contribute.

4.1.2 General code guidelines

Tempo2 is, for historical reasons, mostly written in C but compiled using a C++ compiler. However, be aware that a few parts of tempo2 use C++ clases or other C++ extensions. There is no particular C or C++ version in use, but for now assume that we are using C++98 with GNU extensions (i.e. –std=gnu++98)

Todo determine if we should migrate to C++ 11. It has lots of good features, but we need to check that all compilers support it.

Core tempo2 code

As a general rule, we try to minimise the libraries needed to build the core of tempo2 (not plugins). This means you can't link against libfftw, libpgplot, etc. from the core code. Some linear algebra features from BLAS/LAPACK are made avaliable to the code code via the T2toolkit, and fallback routines have been generated to ensure that the code still works without BLAS/LAPACK. These routines are being expanded all the time.

plugins

For plugins, the rules are much less strict. Currently we compile plugins with links to cfitsio, fftw and pgplot as part of the main plugin distribution.

libt2toolkit

MJK is attempting to introduce a little more rigour in the coding standards for the code that makes up libt2toolkit, but in general this is treated exactly the same as code temo2.

4.1.3 Development workflow

Recommended workflow

The recommended workflow is as follows.

Step 1: create a new branch:

```
git checkout -b myfeature
```

Step 2: Make and commit your changes to that branch

```
git commit -a
```

Step 3: Build, test, run your code.

```
make
make check
```

Step 4: If the new features seem good, promote them to the "master" branch.

```
# if the first time
git push --set-upstream origin docs
# otherwise
git push origin
```

and go to https://bitbucket.org/mkeith/tempo2/pull-requests/new to make a new pull request. The code will be reviewed by the core developers to check that the changes do not break any important features. If the modification is accepted (almost always) then it will be merged.

Alternative workflow

If you can't be bothered with branches, you can simply work directly on the "dev" branch:

```
git checkout dev
```

And commit as you want.

```
git commit -a && git push origin
```

The dev branch will be merged into master, after code review, as and when required. The drawbacks of this method are that you have to deal with conflicts yourself.

4.1.4 Coding style

Tempo2 does not have a strict coding style. However, it is recommended to adopt the following practice, as illustrated by the snippet below:

```
// copyright statement up here.
#ifdef HAVE_CONFIG_H
#include <config.h> // make sure to include config.h
#endif

#include <cstdint>   // standard libries are included first
#include <fftw.h>    // then external libraries
#include "TKlog.h"  // then internal libraries


// functions are prefarably camelCase with small first letter.
// strings should be declared as const char* (or std::string) as they are immutable.
void myFunction(int anInt, const char *str, double **matrix) {
    // indent is 4 spaces.

    // use stdint types where possible to avoid confusion on 32-bit vs 64-bit machines.
    // use unsigned types whre sutable
    // use const when a variable will not change
    const uint64_t myconst = 1024;
    // keywords have a space before parenthesis (e.g. if, for, while).
    if (anInt < 10) { // always use braces, even if one line!
        // use TKlog for logging debug messages and warnings.

        // debug for statements that are to be printed when debug flag is set
        logdbg("anInt = %d",anInt);

        // warnings when problem might be an issue but can continue
        logwarn("anInt should be less than 10"); // adds a message to the warning stack

        // messages always appear
        logmsg("Print to terminal")

        // errors for when the operation is likely to fail.
        logerr("aborting because anInt was too large (%d)",anInt);

        // prefer to return on error rather than exit
        return;
    }

    // best to declare variables in for loops, but give them a proper name (not i, j, k) if possible.
    for (size_t iVal = 0; ival < myconst; ival++) {
        // ...
    }

}
```

Note

Core tempo2 code should be copyright George Hobbs and Russell Edwards until we decide to change this.

Headers should declare the functions and have documentation! Please avoid globals as much as possible, but sometimes they are required. Use any doxygen markup required to document the interface, ESPECIALLY if it is to be called from outside tempo2.

```
// use defines to prevent double declaration
#ifndef myHeader_h
#define myHeader_h


/*!
 *  @brief A brief description of the function
 *  @param anInt[in]   description of this parameter
 *  @param str[in]     description of this parameter
 *  @param matrix[out] descrition, note if it is an "output" parameter!
 *
 *  More description if required
 */


void myFunction(int anInt, const char* str, double** matrix);
#endif
```

Chapter 5

Directory structure

The tempo2 directory structure:

```
.
+-- autoconf.boot
+-- documentation
+-- mpack_lite
+-- plugin
+-- sofa
+-- t2runtime
+-+ tests
  +-- gtest-1.7.0
  +-- test_data
+-- unsupported_plugins
```

autoconf.boot

This directory contains the .m4 files used by autoconf to build the configure script. It is copied to autoconf/ by the bootstrap script.

documentation

Includes this documentation

mpack_lite

Source code for multi-precision lapack/blas. This is a subset of the mplapack package from http://mplapack.sourceforge.net/

plugin

Source code for plugins

sofa

Source code for the 3rd party fortran SOFA library.

T2runtime

This directory contains the runtime files for tempo2, i.e. the contents of this directory should be reached at $TEMPO2 This includes the clock correction files, observatory parameters and earth ephemerdies, etc.

tests

Source code for the unit tests, and the gtest library. Also contains a number of data files in the test_data subdirectory used by the tests.

unsupported_plugins

Source code for other plugins that are for whatever reason not part of the main distribution.

Chapter 6

Plugin Documentation

6.1 Tempo2 Plugins

[TOC]

Chapter 7

README

![Build Status](https://drone.io/bitbucket.org/psrsoft/tempo2/status.png)

Git INSTALLATION README

0. Contents

1. What this package is
2. Quick Guide
3. Requirements
4. Detailed instalation guide
5. Plugins
6. Changes from old makefile
7. Installation troubleshooting
8. Bugs and feature requests

1. What this package is

You (or someone else) have checked out tempo2 from the Git (https://bitbucket.org/psrsoft/tempo2)

This is the best way to get the latest/cutting edge version, and develop your own additions to the tempo2 code or via plugins.

For more information on tempo2 see: http://www.atnf.csiro.au/research/pulsar/tempo2/

This requires the gnu autotools. If you don't have or don't want to install autotools, we recommend you install the latest distributed release from http://www.atnf.csiro.au/research/pulsar/tempo2/ or use PSRSOFT to install tempo2: http://www.pulsarastronomy.net/wiki/Software/PSRSoft

2. Quick Guide

Bootstrap the build system:

```
./bootstrap
```

setup the tempo2 runtime dir

```
cp -r T2runtime /usr/share/tempo2/
export TEMPO2=/usr/share/tempo2/
```

Configure:

```
./configure [[--prefix=/your/install/path]]
```

use –prefix to set the path you want to install the binaries and libraries

Make and install...

```
make && make install
```

You will probably want to build the default plugins (plk, etc). Do this with:

```
make plugins && make plugins-install
```

And you're done.

3. Requirements

Tempo2 requires the following:

- A fortran 77 compiler (tested with gfortran).
- A C compiler (tested with gcc).

Plugins may have other requirements, notably PGPLOT.

5. Plugins

The bootstrap command will create suitible makefiles for the default set of plugins. This is controled by the contents of the files in ./plugin/plugin_lists/

- vanilla.plugins lists plugins to install which have no dependancies.
- pgplot.plugins lists plugins to install that are dependant on PGPLOT.
- gsl.plugins lists plugins to install that are dependant on the GSL.

5.1 Building your own plugin

The easiest way to compile your own plugins is:

```
g++ {$CFLAGS} {$LDFLAGS} -fPIC -shared -o {$TEMPO2}/plugins/{$PLG_NAME}_{$LOGIN_ARCH}_plug.t2 {$SRCLIST}
```

where:

- `{$PLG_NAME}` is the name of your plugin
- `{$SRCLIST}` is your plugin's source code.
- `{$LOGIN_ARCH}` is the result of `` `uname` `` (usualy Linux).
- `{$CFLAGS}` are the compiler flags your plugin needs... remeber to add a -I option to point to the location of tempo2.h
- `{$LDFLAGS}` are any linking options you need, e.g. pgplot, etc.
- `{$TEMPO2}` is the tempo2 runtime dir

For example, to compile a basic plugin called '`foo`' on linux, you might do

```
g++ -I/usr/src/tempo2 -fPIC -shared -o $TEMPO2/plugins/foo_{$LOGIN_ARCH}_plug.t2 foo_plug.C
```

5.2 Adding a new plugin to the default build list

If your plugin has dependances that are already covered by the lists above, just add the name to the appropriate list, and name your plugin source file as:

name_plug.C

6. Changes from the old Make system.

At the start of 2010, tempo2 moved over to an autotools based make system, replacing the old hand written makefiles. This may confuse some people!

Important notes:

- Tempo2 plugins now have a .t2 extention, rather than the old .so This is to ensure reduce confusion on MacOSx and to allow the old make system and the new make system to co-exist for a while.
- Any 3rd party plugins will still work as before. Indeed, to update a plugin, just change the .so extention to a .t2 extention. e.g. mv general_Linux_plug.so general_Linux_plug.t2

7. Installation Troubleshooting

7.1 Can't find PGPLOT

Download pgplot from: http://www.astro.caltech.edu/~tjp/pgplot/

Or use PSRSOFT to manage the installation. http://www.pulsarastronomy.net/wiki/↩Software/PSRSoft

If you have pgplot installed, but it is not detected by the configure script, check:

- You have got at least libpgplot.a and libcpgplot.a in your LDFLAGS
- Check you have `$PGPLOT_DIR` pointing to the folder with grfont.dat and rgb.txt
- Check that you have `$F77` set to the same compiler that compiled PGPLOT (e.g. setenv F77 gfortran, if you used gfortran for PGPLOT)

7.2 Incompatible C and Fortran compilers

Check that you are using the same build of gcc and gfortran (or whatever compiler you are using).

Note that on MacOSX there is often an issue where the default compiler is incompatible with gfortran. The gfortran compatible version is often called gcc-4 and gxx-4 or similar. Use this with:

```
export CC=gcc-4
export CXX=g++-4
```

and reconfigure.

8. Bugs and feature requests

Please submit bug reports here: https://bitbucket.org/psrsoft/tempo2/issues/new

Note that it is very helpful if you can upload a small example demonstrating the bug!

Chapter 8

Todo List

Page Developer Guide

determine if we should migrate to C++ 11. It has lots of good features, but we need to check that all compilers support it.

Chapter 9

Module Index

9.1 Modules

Here is a list of all modules:

Chapter 10

Class Index

10.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 11

File Index

11.1 File List

Here is a list of all files with brief descriptions:

Chapter 12

Module Documentation

12.1 libt2toolkit API

Files

- file T2toolkit.h

  Set of routines that are commonly used in tempo2 and/or its plugins.

12.1.1 Detailed Description

12.2 libtempo2 External API

Files

- file tempo2.h

  contains the main interface to libtempo2.

12.2.1 Detailed Description

Chapter 13

Class Documentation

13.1 Cheby2D Struct Reference

```
#include <tempo2pred.h>
```

Public Attributes

- int nx
- int ny
- long double $*$ coeff

13.1.1 Member Data Documentation

13.1.1.1 long double$*$ Cheby2D::coeff

13.1.1.2 int Cheby2D::nx

13.1.1.3 int Cheby2D::ny

The documentation for this struct was generated from the following file:

- tempo2pred.h

13.2 ChebyModel Struct Reference

```
#include <tempo2pred.h>
```

Collaboration diagram for ChebyModel:

Public Attributes

- char psrname [64]
- char sitename [64]
- long double mjd_start
- long double mjd_end
- long double freq_start
- long double freq_end
- long double dispersion_constant

- Cheby2D cheby
- Cheby2D frequency_cheby

13.2.1 Member Data Documentation

13.2.1.1 Cheby2D ChebyModel::cheby

13.2.1.2 long double ChebyModel::dispersion_constant

13.2.1.3 long double ChebyModel::freq_end

13.2.1.4 long double ChebyModel::freq_start

13.2.1.5 Cheby2D ChebyModel::frequency_cheby

13.2.1.6 long double ChebyModel::mjd_end

13.2.1.7 long double ChebyModel::mjd_start

13.2.1.8 char ChebyModel::psrname[64]

13.2.1.9 char ChebyModel::sitename[64]

The documentation for this struct was generated from the following file:

- tempo2pred.h

13.3 ChebyModelSet Struct Reference

```
#include <tempo2pred.h>
```

Collaboration diagram for ChebyModelSet:

Public Attributes

- ChebyModel ∗ segments
- int nsegments

13.3.1 Member Data Documentation

13.3.1.1 int ChebyModelSet::nsegments

13.3.1.2 ChebyModel∗ ChebyModelSet::segments

The documentation for this struct was generated from the following file:

- tempo2pred.h

13.4 clock_correction Struct Reference

```
#include <tempo2.h>
```

Public Attributes

- double correction
- char corrects_to [32]

13.4.1 Detailed Description

observation contains an array of these, which getClockCorrections() fills in

13.4.2 Member Data Documentation

13.4.2.1 double clock_correction::correction

13.4.2.2 char clock_correction::corrects_to[32]

The documentation for this struct was generated from the following file:

- tempo2.h

13.5 complexVal Struct Reference

```
#include <TKspectrum.h>
```

Public Attributes

- double real
- double imag

13.5.1 Member Data Documentation

13.5.1.1 double complexVal::imag

13.5.1.2 double complexVal::real

The documentation for this struct was generated from the following file:

- TKspectrum.h

13.6 DynamicArray Struct Reference

```
#include <dynarr.h>
```

Public Attributes

- void ∗ data
- size_t nelem
- size_t elem_size
- size_t nalloced

13.6.1 Member Data Documentation

13.6.1.1 void∗ DynamicArray::data

13.6.1.2 size_t DynamicArray::elem_size

13.6.1.3 size_t DynamicArray::nalloced

13.6.1.4 size_t DynamicArray::nelem

The documentation for this struct was generated from the following file:

- dynarr.h

13.7 FitInfo Struct Reference

contains details of the fit

```
#include <tempo2.h>
```

Collaboration diagram for FitInfo:

Public Attributes

- unsigned nParams
- unsigned nConstraints
- param_label paramIndex [MAX_FIT]
- constraint_label constraintIndex [MAX_FIT]
- int paramCounters [MAX_FIT]
- int constraintCounters [MAX_FIT]
- paramDerivFunc paramDerivs [MAX_FIT]
- constraintDerivFunc constraintDerivs [MAX_FIT]
- paramUpdateFunc updateFunctions [MAX_FIT]

13.7.1 Detailed Description

contains details of the fit

Holds references to the fit functions, as well as references linking the index in the derivative matrix to the actual parameter fit for.

13.7.2 Member Data Documentation

13.7.2.1 int FitInfo::constraintCounters[MAX_FIT]

13.7.2.2 constraintDerivFunc FitInfo::constraintDerivs[MAX_FIT]

13.7.2.3 constraint_label FitInfo::constraintIndex[MAX_FIT]

13.7.2.4 unsigned FitInfo::nConstraints

13.7.2.5 unsigned FitInfo::nParams

13.7.2.6 int FitInfo::paramCounters[MAX_FIT]

13.7.2.7 paramDerivFunc FitInfo::paramDerivs[MAX_FIT]

13.7.2.8 param_label FitInfo::paramIndex[MAX_FIT]

13.7.2.9 paramUpdateFunc FitInfo::updateFunctions[MAX_FIT]

The documentation for this struct was generated from the following file:

- tempo2.h

13.8 gwgeneralSrc Struct Reference

```
#include <GWsim.h>
```

Public Attributes

- longdouble theta_g
- longdouble phi_g
- longdouble omega_g
- longdouble phi_polar_g
- longdouble phase_g
- longdouble aplus_g
- longdouble aplus_im_g
- longdouble across_g
- longdouble across_im_g
- longdouble ast_g
- longdouble ast_im_g
- longdouble asl_g
- longdouble asl_im_g
- longdouble avx_g
- longdouble avx_im_g
- longdouble avy_g
- longdouble avy_im_g
- longdouble phi_bin
- longdouble theta_bin
- longdouble inc_bin
- longdouble dist_bin
- longdouble h [3][3]
- longdouble h_im [3][3]
- longdouble kg [3]

13.8.1 Member Data Documentation

13.8.1.1 longdouble gwgeneralSrc::across_g

13.8.1.2 longdouble gwgeneralSrc::across_im_g

13.8.1.3 longdouble gwgeneralSrc::aplus_g

13.8.1.4 longdouble gwgeneralSrc::aplus_im_g

13.8.1.5 longdouble gwgeneralSrc::asl_g

13.8.1.6 longdouble gwgeneralSrc::asl_im_g

13.8.1.7 longdouble gwgeneralSrc::ast_g

13.8.1.8 longdouble gwgeneralSrc::ast_im_g

13.8.1.9 longdouble gwgeneralSrc::avx_g

13.8.1.10 longdouble gwgeneralSrc::avx_im_g

13.8.1.11 longdouble gwgeneralSrc::avy_g

13.8.1.12 longdouble gwgeneralSrc::avy_im_g

13.8.1.13 longdouble gwgeneralSrc::dist_bin

13.8.1.14 longdouble gwgeneralSrc::h[3][3]

13.8.1.15 longdouble gwgeneralSrc::h_im[3][3]

13.8.1.16 longdouble gwgeneralSrc::inc_bin

13.8.1.17 longdouble gwgeneralSrc::kg[3]

13.8.1.18 longdouble gwgeneralSrc::omega_g

13.8.1.19 longdouble gwgeneralSrc::phase_g

13.8.1.20 longdouble gwgeneralSrc::phi_bin

13.8.1.21 longdouble gwgeneralSrc::phi_g

13.8.1.22 longdouble gwgeneralSrc::phi_polar_g

13.8.1.23 longdouble gwgeneralSrc::theta_bin

13.8.1.24 longdouble gwgeneralSrc::theta_g

The documentation for this struct was generated from the following file:

- GWsim.h

13.9 gwgenSpec Struct Reference

```
#include <GWsim.h>
```

Public Attributes

- double tensor_amp
- double st_amp
- double sl_amp
- double vl_amp
- double tensor_alpha
- double st_alpha

- double sl_alpha
- double vl_alpha

13.9.1 Member Data Documentation

13.9.1.1 double gwgenSpec::sl_alpha

13.9.1.2 double gwgenSpec::sl_amp

13.9.1.3 double gwgenSpec::st_alpha

13.9.1.4 double gwgenSpec::st_amp

13.9.1.5 double gwgenSpec::tensor_alpha

13.9.1.6 double gwgenSpec::tensor_amp

13.9.1.7 double gwgenSpec::vl_alpha

13.9.1.8 double gwgenSpec::vl_amp

The documentation for this struct was generated from the following file:

- GWsim.h

13.10 gwSrc Struct Reference

```
#include <GWsim.h>
```

Public Attributes

- longdouble theta_g
- longdouble phi_g
- longdouble omega_g
- longdouble phi_polar_g
- longdouble phase_g
- longdouble aplus_g
- longdouble aplus_im_g
- longdouble across_g
- longdouble across_im_g
- longdouble phi_bin
- longdouble theta_bin
- longdouble inc_bin
- longdouble dist_bin
- longdouble h [3][3]
- longdouble h_im [3][3]
- longdouble kg [3]

13.10.1 Member Data Documentation

13.10.1.1 longdouble gwSrc::across_g

13.10.1.2 longdouble gwSrc::across_im_g

13.10.1.3 longdouble gwSrc::aplus_g

13.10.1.4 longdouble gwSrc::aplus_im_g

13.10.1.5 longdouble gwSrc::dist_bin

13.10.1.6 longdouble gwSrc::h[3][3]

13.10.1.7 longdouble gwSrc::h_im[3][3]

13.10.1.8 longdouble gwSrc::inc_bin

13.10.1.9 longdouble gwSrc::kg[3]

13.10.1.10 longdouble gwSrc::omega_g

13.10.1.11 longdouble gwSrc::phase_g

13.10.1.12 longdouble gwSrc::phi_bin

13.10.1.13 longdouble gwSrc::phi_g

13.10.1.14 longdouble gwSrc::phi_polar_g

13.10.1.15 longdouble gwSrc::theta_bin

13.10.1.16 longdouble gwSrc::theta_g

The documentation for this struct was generated from the following file:

- GWsim.h

13.11 interpolation_info Struct Reference

```
#include <jpl_int.h>
```

Public Attributes

- double posn_coeff [MAX_CHEBY]
- double vel_coeff [MAX_CHEBY]
- double twot
- unsigned n_posn_avail
- unsigned n_vel_avail

13.11.1 Member Data Documentation

13.11.1.1 unsigned interpolation_info::n_posn_avail

13.11.1.2 unsigned interpolation_info::n_vel_avail

13.11.1.3 double interpolation_info::posn_coeff[MAX_CHEBY]

13.11.1.4 double interpolation_info::twot

13.11.1.5 double interpolation_info::vel_coeff[MAX_CHEBY]

The documentation for this struct was generated from the following file:

- jpl_int.h

13.12 jpl_eph_data Struct Reference

```
#include <jpl_int.h>
```

Collaboration diagram for jpl_eph_data:

Public Attributes

- double ephem_start
- double ephem_end
- double ephem_step
- uint32_t ncon
- double au
- double emrat
- uint32_t ipt [15][3]
- uint32_t ephemeris_version
- uint32_t kernel_size
- uint32_t recsize
- uint32_t ncoeff
- uint32_t swap_bytes
- uint32_t curr_cache_loc
- double pvsun [9]
- double pvsun_t
- double ∗ cache
- struct interpolation_info iinfo
- FILE ∗ ifile

13.12.1 Member Data Documentation

13.12.1.1 double jpl_eph_data::au

13.12.1.2 double∗ jpl_eph_data::cache

13.12.1.3 uint32_t jpl_eph_data::curr_cache_loc

13.12.1.4 double jpl_eph_data::emrat

13.12.1.5 double jpl_eph_data::ephem_end

13.12.1.6 double jpl_eph_data::ephem_start

13.12.1.7 double jpl_eph_data::ephem_step

13.12.1.8 uint32_t jpl_eph_data::ephemeris_version

13.12.1.9 FILE∗ jpl_eph_data::ifile

13.12.1.10 struct interpolation_info jpl_eph_data::iinfo

13.12.1.11 uint32_t jpl_eph_data::ipt[15][3]

13.12.1.12 uint32_t jpl_eph_data::kernel_size

13.12.1.13 uint32_t jpl_eph_data::ncoeff

13.12.1.14 uint32_t jpl_eph_data::ncon

13.12.1.15 double jpl_eph_data::pvsun[9]

13.12.1.16 double jpl_eph_data::pvsun_t

13.12.1.17 uint32_t jpl_eph_data::recsize

13.12.1.18 uint32_t jpl_eph_data::swap_bytes

The documentation for this struct was generated from the following file:

- jpl_int.h

13.13 observation Struct Reference

A struct containing the details of a single obesrvation.

```
#include <tempo2.h>
```

Collaboration diagram for observation:

Public Attributes

- longdouble sat
- longdouble origsat
- longdouble sat_day
- longdouble sat_sec
- longdouble bat
- longdouble batCorr
- longdouble bbat
- longdouble pet
- int clockCorr
- int delayCorr
- int deleted
- longdouble prefitResidual
- longdouble residual
- double addedNoise
- double TNRedSignal
- double TNRedErr
- double TNDMSignal

- double TNDMErr
- double TNGroupSignal
- double TNGroupErr
- double freq
- double freqSSB
- double toaErr
- double toaDMErr
- double origErr
- double phaseOffset
- double averagebat
- double averageres
- double averageerr
- double averagedmbat
- double averagedmres
- double averagedmerr
- char fname [MAX_FILELEN]
- char telID [100]
- clock_correction correctionsTT [MAX_CLK_CORR]
- int nclock_correction
- longdouble correctionTT_TB
- double einsteinRate
- longdouble correctionTT_calcEph
- longdouble correctionTT_Teph
- longdouble correctionUT1
- double sun_ssb [6]
- double sun_earth [6]
- double planet_ssb [9][6]
- double planet_ssb_tmr [9][6]
- double planet_ssb_derv [9][6]
- double jupiter_earth [6]
- double saturn_earth [6]
- double venus_earth [6]
- double uranus_earth [6]
- double neptune_earth [6]
- double earthMoonBary_ssb [6]
- double earthMoonBary_earth [6]
- double earth_ssb [6]
- double observatory_earth [6]
- double psrPos [3]
- double zenith [3]
- double nutations [6]
- double siteVel [3]
- longdouble shklovskii
- double shapiroDelaySun
- double shapiroDelayJupiter
- double shapiroDelaySaturn
- double shapiroDelayVenus
- double shapiroDelayUranus
- double shapiroDelayNeptune
- double troposphericDelay
- double tdis1
- double tdis2
- longdouble roemer
- longdouble torb
- longdouble nphase

- longdouble phase
- long long pulseN
- char flagID [MAX_FLAGS][MAX_FLAG_LEN]
- char flagVal [MAX_FLAGS][MAX_FLAG_LEN]
- int nFlags
- int jump [MAX_FLAGS]
- int obsNjump
- double efac
- double equad

13.13.1 Detailed Description

A struct containing the details of a single obesrvation.

13.13.2 Member Data Documentation

13.13.2.1 double observation::addedNoise

13.13.2.2 double observation::averagebat

13.13.2.3 double observation::averagedmbat

13.13.2.4 double observation::averagedmerr

13.13.2.5 double observation::averagedmres

13.13.2.6 double observation::averageerr

13.13.2.7 double observation::averageres

13.13.2.8 longdouble observation::bat

Infinite frequency barycentric arrival time

13.13.2.9 longdouble observation::batCorr

13.13.2.10 longdouble observation::bbat

Arrival time at binary barycentre

13.13.2.11 int observation::clockCorr

= 1 for clock corrections to be applied, = 0 for BAT

13.13.2.12 clock_correction observation::correctionsTT[MAX_CLK_CORR]

chain of corrections from site TOA to chosen realisation of TT

13.13.2.13 longdouble observation::correctionTT_calcEph

13.13.2.14 longdouble observation::correctionTT_TB

Correction to TDB/TCB

13.13.2.15 longdouble observation::correctionTT_Teph

Correction to Teph

13.13.2.16 longdouble observation::correctionUT1

Correction from site TOA to UT1

13.13.2.17 int observation::delayCorr

= 1 for time delay corrections to be applied, = 0 for BAT

13.13.2.18 int observation::deleted

= 1 if observation has been deleted, = -1 if not included in fit

13.13.2.19 double observation::earth_ssb[6]

Centre of Earth w.r.t. SSB

13.13.2.20 double observation::earthMoonBary_earth[6]

Position of Earth-Moon barycentre with respect to Earth (sec) (RBE)

13.13.2.21 double observation::earthMoonBary_ssb[6]

Ephem values for Earth-Moon barycentre wrt SSB (sec) (RCB)

13.13.2.22 double observation::efac

Error multiplication factor

13.13.2.23 double observation::einsteinRate

Derivative of correctionTT_TB

13.13.2.24 double observation::equad

Value to add in quadrature

13.13.2.25 char observation::flagID[MAX_FLAGS][MAX_FLAG_LEN]

Flags in .tim file

13.13.2.26 char observation::flagVal[MAX_FLAGS][MAX_FLAG_LEN]

13.13.2.27 char observation::fname[MAX_FILELEN]

Name of data file giving TOA

13.13.2.28 double observation::freq

Frequency of observation (in MHz)

13.13.2.29 double observation::freqSSB

Frequency of observation in barycentric frame (in Hz)

13.13.2.30 int observation::jump[MAX_FLAGS]

Jump region

13.13.2.31 double observation::jupiter_earth[6]

Ephemeris values for Jupiter w.r.t. Earth centre (sec)

13.13.2.32 int observation::nclock_correction

13.13.2.33 double observation::neptune_earth[6]

Ephemeris values for Neptune w.r.t. Earth centre (sec)

13.13.2.34 int observation::nFlags

13.13.2.35 longdouble observation::nphase

allows the pulse number to be determined

13.13.2.36 double observation::nutations[6]

13.13.2.37 double observation::observatory_earth[6]

Observatory site with respect to Earth centre (sec) (REA)

13.13.2.38 int observation::obsNjump

Number of jumps for this observation

13.13.2.39 double observation::origErr

Original error on TOA after reading tim file (in us)

13.13.2.40 longdouble observation::origsat

13.13.2.41 longdouble observation::pet

Pulsar emission time

13.13.2.42 longdouble observation::phase

13.13.2.43 double observation::phaseOffset

Phase offset

13.13.2.44 double observation::planet_ssb[9][6]

Ephemeris values for all planets w.r.t. SSB (sec)

13.13.2.45 double observation::planet_ssb_derv[9][6]

13.13.2.46 double observation::planet_ssb_tmr[9][6]

13.13.2.47 longdouble observation::prefitResidual

Pre-fit residual

13.13.2.48 double observation::psrPos[3]

Unit vector giving position of the pulsar at observation time from Earth

13.13.2.49 long long observation::pulseN

Pulse number

13.13.2.50 longdouble observation::residual

residual

13.13.2.51 longdouble observation::roemer

Roemer delay

13.13.2.52 longdouble observation::sat

Site arrival time

13.13.2.53 longdouble observation::sat_day

13.13.2.54 longdouble observation::sat_sec

13.13.2.55 double observation::saturn_earth[6]

Ephemeris values for Saturn w.r.t. Earth centre (sec)

13.13.2.56 double observation::shapiroDelayJupiter

Shapiro Delay due to Jupiter

13.13.2.57 double observation::shapiroDelayNeptune

Shapiro Delay due to Neptune

13.13.2.58 double observation::shapiroDelaySaturn

Shapiro Delay due to Saturn

13.13.2.59 double observation::shapiroDelaySun

Shapiro Delay due to the Sun

13.13.2.60 double observation::shapiroDelayUranus

Shapiro Delay due to Uranus

13.13.2.61 double observation::shapiroDelayVenus

Shapiro Delay due to Venus

13.13.2.62 longdouble observation::shklovskii

Shklovskii delay term

13.13.2.63 double observation::siteVel[3]

Observatory velocity w.r.t. geocentre

13.13.2.64 double observation::sun_earth[6]

Ephemeris values for Sun w.r.t Earth (sec)

13.13.2.65 double observation::sun_ssb[6]

Ephemeris values for Sun w.r.t SSB (sec) (RCS)

13.13.2.66 double observation::tdis1

Interstellar dispersion measure delay

13.13.2.67 double observation::tdis2

Dispersion measure delay due to solar system

13.13.2.68 char observation::telID[100]

Telescope ID

13.13.2.69 double observation::TNDMErr

Error on Model DM signal from temponest fit

13.13.2.70 double observation::TNDMSignal

Model DM signal from temponest fit

13.13.2.71 double observation::TNGroupErr

Error on Model Group Noise signal from temponest fit

13.13.2.72 double observation::TNGroupSignal

Model Group Noise signal from temponest fit

13.13.2.73 double observation::TNRedErr

Error on Model red noise signal from temponest fit

13.13.2.74 double observation::TNRedSignal

Model red noise signal from temponest fit

13.13.2.75 double observation::toaDMErr

Error on TOA due to DM (in us)

13.13.2.76 double observation::toaErr

Error on TOA (in us)

13.13.2.77 longdouble observation::torb

Combined binary delays

13.13.2.78 double observation::troposphericDelay

Delay due to neutral refraction in atmosphere

13.13.2.79 double observation::uranus_earth[6]

Ephemeris values for Uranus w.r.t. Earth centre (sec)

13.13.2.80 double observation::venus_earth[6]

Ephemeris values for Venus w.r.t. Earth centre (sec)

13.13.2.81 double observation::zenith[3]

Zenith vector, in BC frame. Length=geodetic height

The documentation for this struct was generated from the following file:

- tempo2.h

13.14 observatory Struct Reference

```
#include <tempo2.h>
```

Public Attributes

- double x
- double y
- double z
- double longitude_grs80
- double latitude_grs80
- double height_grs80
- char name [32]
- char code [16]
- char clock_name [16]

13.14.1 Member Data Documentation

13.14.1.1 char observatory::clock_name[16]

13.14.1.2 char observatory::code[16]

13.14.1.3 double observatory::height_grs80

13.14.1.4 double observatory::latitude_grs80

13.14.1.5 double observatory::longitude_grs80

13.14.1.6 char observatory::name[32]

13.14.1.7 double observatory::x

13.14.1.8 double observatory::y

13.14.1.9 double observatory::z

The documentation for this struct was generated from the following file:

- tempo2.h

13.15 parameter Struct Reference

Holds the values for a parameter.

```
#include <tempo2.h>
```

Public Attributes

- char ∗∗ label
- char ∗∗ shortlabel
- longdouble ∗ val
- longdouble ∗ err
- int ∗ fitFlag
- int ∗ paramSet
- longdouble ∗ prefit
- longdouble ∗ prefitErr
- int aSize
- int linkFrom [5]
- int linkTo [5]
- int nLinkTo
- int nLinkFrom

13.15.1 Detailed Description

Holds the values for a parameter.

May include multiple values, for e.g. F0, F1, F2,...

Note

> If this structure is modified - must update copyParam in tempo2Util.C

13.15.2 Member Data Documentation

13.15.2.1 int parameter::aSize

Number of elements in the array for this parameter

13.15.2.2 longdouble∗ parameter::err

Uncertainty on parameter value

13.15.2.3 int∗ parameter::fitFlag

= 1 if fitting required, = 2 for global fit

13.15.2.4 char∗∗ parameter::label

Label about this parameter

13.15.2.5 int parameter::linkFrom[5]

13.15.2.6 int parameter::linkTo[5]

13.15.2.7 int parameter::nLinkFrom

13.15.2.8 int parameter::nLinkTo

13.15.2.9 int∗ parameter::paramSet

= 1 if parameter has been set

13.15.2.10 longdouble∗ parameter::prefit

Pre-fit value of the parameter

13.15.2.11 longdouble∗ parameter::prefitErr

Pre-fit value of the uncertainty

13.15.2.12 char∗∗ parameter::shortlabel

Label about this parameter without units

13.15.2.13 longdouble∗ parameter::val

Value of parameter

The documentation for this struct was generated from the following file:

- tempo2.h

13.16 pulsar Struct Reference

contains the details for a single pulsar.

```
#include <tempo2.h>
```

Collaboration diagram for pulsar:

Public Attributes

- char name [100]
- char eopc04_file [MAX_FILELEN]
- int fixedFormat
- parameter param [MAX_PARAMS]
- char rajStrPre [100]
- char decjStrPre [100]
- char rajStrPost [100]
- char decjStrPost [100]
- char binaryModel [100]
- double ∗∗ ToAextraCovar
- int dmoffsDMnum
- int dmoffsCMnum
- double dmoffsDM_mjd [MAX_IFUNC]
- double dmoffsDM [MAX_IFUNC]
- double dmoffsDM_error [MAX_IFUNC]
- double dmoffsDM_weight [MAX_IFUNC]
- double dmoffsCM_mjd [MAX_IFUNC]
- double dmoffsCM [MAX_IFUNC]
- double dmoffsCM_error [MAX_IFUNC]
- double dmoffsCM_weight [MAX_IFUNC]
- double gwsrc_ra
- double gwsrc_dec
- double gwsrc_aplus_r

- double gwsrc_aplus_i
- double gwsrc_across_r
- double gwsrc_across_i
- double gwsrc_aplus_r_e
- double gwsrc_aplus_i_e
- double gwsrc_across_r_e
- double gwsrc_across_i_e
- double gwsrc_epoch
- double gwsrc_psrdist
- double cgw_h0
- double cgw_cosinc
- double cgw_angpol
- double cgw_mc
- double gwm_raj
- double gwm_decj
- double gwm_epoch
- double gwm_phi
- double gwm_dphase
- double gwb_epoch
- double gwb_width
- double gwb_raj
- double gwb_decj
- double gwb_geom_c
- double gwb_geom_p
- double gwecc_ra
- double gwecc_dec
- double gwecc_m1
- double gwecc_m2
- double gwecc_e
- double gwecc_inc
- double gwecc_theta_nodes
- double gwecc_nodes_orientation
- double gwecc_theta_0
- double gwecc_orbital_period
- double gwecc_distance
- double gwecc_redshift
- double gwecc_epoch
- double gwecc_psrdist
- int gwecc_pulsarTermOn
- double posPulsar [3]
- double velPulsar [3]
- longdouble phaseJump [MAX_JUMPS]
- int phaseJumpDir [MAX_JUMPS]
- int phaseJumpID [MAX_JUMPS]
- int nPhaseJump
- double dmOffset
- double ne_sw
- int nCompanion
- int eclCoord
- int nJumps
- char fjumpID [16]
- double jumpVal [MAX_JUMPS]
- int fitJump [MAX_JUMPS]
- double jumpValErr [MAX_JUMPS]
- char jumpStr [MAX_JUMPS][MAX_STRLEN]

- char filterStr [MAX_STRLEN]
- char passStr [MAX_STRLEN]
- double tOffset [MAX_TOFFSET]
- double tOffset_f1 [MAX_TOFFSET]
- double tOffset_f2 [MAX_TOFFSET]
- double tOffset_t1 [MAX_TOFFSET]
- double tOffset_t2 [MAX_TOFFSET]
- char tOffsetSite [MAX_TOFFSET][100]
- char tOffsetFlags [MAX_TOFFSET][1000]
- int nToffset
- int ndmx
- double fitChisq
- int fitNfree
- int globalNfit
- int globalNoConstrain
- int nFit
- int nParam
- int nGlobal
- int fitParamGlobalI [MAX_FIT]
- int fitParamGlobalK [MAX_FIT]
- int fitParamI [MAX_FIT]
- int fitParamK [MAX_FIT]
- int fitMode
- char robust
- int rescaleErrChisq
- double offset
- double offset_e
- double ∗∗ covar
- int calcShapiro
- int planetShapiro
- int jboFormat
- observation ∗ obsn
- int nobs
- int units
- int setUnits
- int tempo1
- int dilateFreq
- int timeEphemeris
- int t2cMethod
- int correctTroposphere
- int noWarnings
- char sorted
- char clock [16]
- char clockFromOverride [64]
- char JPL_EPHEMERIS [MAX_FILELEN]
- char ephemeris [MAX_FILELEN]
- int useCalceph
- storePrecision storePrec [MAX_STOREPRECISION]
- int nStorePrecision
- int bootStrap
- char tzrsite [100]
- double rmsPre
- double rmsPost
- char deleteFileName [100]
- int nits

- int ipm
- int swm
- double wave_sine [MAX_WHITE]
- double wave_sine_err [MAX_WHITE]
- double wave_cos [MAX_WHITE]
- double wave_cos_err [MAX_WHITE]
- double wave_sine_dm [MAX_WHITE]
- double wave_sine_dm_err [MAX_WHITE]
- double wave_cos_dm [MAX_WHITE]
- double wave_cos_dm_err [MAX_WHITE]
- int nWhite
- int nWhite_dm
- double waveScale
- double quad_aplus_r [MAX_QUAD]
- double quad_aplus_r_e [MAX_QUAD]
- double quad_aplus_i [MAX_QUAD]
- double quad_aplus_i_e [MAX_QUAD]
- double quad_across_r [MAX_QUAD]
- double quad_across_r_e [MAX_QUAD]
- double quad_across_i [MAX_QUAD]
- double quad_across_i_e [MAX_QUAD]
- double quadEpoch
- double quadRA
- double quadDEC
- int nQuad
- double ifuncT [MAX_IFUNC]
- double ifuncV [MAX_IFUNC]
- double ifuncE [MAX_IFUNC]
- double ifunc_weights [MAX_IFUNC]
- int ifuncN
- double clk_offsT [MAX_TEL_CLK_OFFS]
- double clk_offsV [MAX_TEL_CLK_OFFS]
- double clk_offsE [MAX_TEL_CLK_OFFS]
- int clkOffsN
- double quad_ifuncT_p [MAX_IFUNC]
- double quad_ifuncV_p [MAX_IFUNC]
- double quad_ifuncE_p [MAX_IFUNC]
- int quad_ifuncN_p
- double quad_ifuncT_c [MAX_IFUNC]
- double quad_ifuncV_c [MAX_IFUNC]
- double quad_ifuncE_c [MAX_IFUNC]
- int quad_ifuncN_c
- double quad_ifunc_p_RA
- double quad_ifunc_p_DEC
- double quad_ifunc_c_RA
- double quad_ifunc_c_DEC
- double quad_ifunc_geom_p
- double quad_ifunc_geom_c
- int nTelDX
- int setTelVelX
- double telDX_t [MAX_TEL_DX]
- double telDX_v [MAX_TEL_DX]
- double telDX_e [MAX_TEL_DX]
- double telDX_vel [MAX_TEL_DX]
- double telDX_vel_e [MAX_TEL_DX]

- int nTelDY
- int setTelVelY
- double telDY_t [MAX_TEL_DY]
- double telDY_v [MAX_TEL_DY]
- double telDY_e [MAX_TEL_DY]
- double telDY_vel [MAX_TEL_DY]
- double telDY_vel_e [MAX_TEL_DY]
- int nTelDZ
- int setTelVelZ
- double telDZ_v [MAX_TEL_DZ]
- double telDZ_t [MAX_TEL_DZ]
- double telDZ_e [MAX_TEL_DZ]
- double telDZ_vel [MAX_TEL_DZ]
- double telDZ_vel_e [MAX_TEL_DZ]
- int nT2efac
- int nT2equad
- char T2efacFlagID [MAX_T2EFAC][MAX_FLAG_LEN]
- char T2efacFlagVal [MAX_T2EFAC][MAX_FLAG_LEN]
- double T2efacVal [MAX_T2EFAC]
- char T2equadFlagID [MAX_T2EQUAD][MAX_FLAG_LEN]
- char T2equadFlagVal [MAX_T2EQUAD][MAX_FLAG_LEN]
- double T2equadVal [MAX_T2EQUAD]
- double T2globalEfac
- int nTNEF
- int nTNEQ
- int nTNSQ
- int nTNECORR
- char TNEFFlagID [MAX_TNEF][MAX_FLAG_LEN]
- char TNEFFlagVal [MAX_TNEF][MAX_FLAG_LEN]
- double TNEFVal [MAX_TNEF]
- double TNGlobalEF
- char TNEQFlagID [MAX_TNEQ][MAX_FLAG_LEN]
- char TNEQFlagVal [MAX_TNEQ][MAX_FLAG_LEN]
- double TNEQVal [MAX_TNEQ]
- double TNGlobalEQ
- double addTNGlobalEQ
- char TNSQFlagID [MAX_TNSQ][MAX_FLAG_LEN]
- char TNSQFlagVal [MAX_TNSQ][MAX_FLAG_LEN]
- double TNSQVal [MAX_TNSQ]
- char TNECORRFlagID [MAX_TNECORR][MAX_FLAG_LEN]
- char TNECORRFlagVal [MAX_TNECORR][MAX_FLAG_LEN]
- double TNECORRVal [MAX_TNECORR]
- double TNRedAmp
- double TNRedGam
- int TNRedC
- double TNRedCoeffs [200]
- double TNRedFLow
- double TNRedCorner
- double TNDMAmp
- double TNDMGam
- int TNDMC
- double TNDMCoeffs [200]
- int TNsubtractDM
- int TNsubtractRed
- int AverageResiduals

- int AverageDMResiduals
- char AverageFlag [MAX_FLAG_LEN]
- float AverageEpochWidth
- int outputTMatrix
- int useTNOrth
- double TNBandDMAmp
- double TNBandDMGam
- int TNBandDMC
- int nTNBandNoise
- double TNBandNoiseLF [MAX_TNBN]
- double TNBandNoiseHF [MAX_TNBN]
- double TNBandNoiseAmp [MAX_TNBN]
- double TNBandNoiseGam [MAX_TNBN]
- int TNBandNoiseC [MAX_TNBN]
- int nTNGroupNoise
- char TNGroupNoiseFlagID [MAX_TNGN][MAX_FLAG_LEN]
- char TNGroupNoiseFlagVal [MAX_TNGN][MAX_FLAG_LEN]
- double TNGroupNoiseAmp [MAX_TNGN]
- double TNGroupNoiseGam [MAX_TNGN]
- int TNGroupNoiseC [MAX_TNGN]
- int nDMEvents
- double TNDMEvStart [MAX_TNDMEv]
- double TNDMEvLength [MAX_TNDMEv]
- double TNDMEvAmp [MAX_TNDMEv]
- double TNDMEvGam [MAX_TNDMEv]
- int TNDMEvOff [MAX_TNDMEv]
- int TNDMEvLin [MAX_TNDMEv]
- int TNDMEvQuad [MAX_TNDMEv]
- int nTNShapeletEvents
- int TNShapeletEvN [MAX_TNDMEv]
- double TNShapeletEvPos [MAX_TNDMEv]
- double TNShapeletEvWidth [MAX_TNDMEv]
- double TNShapeletEvFScale [MAX_TNDMEv]
- char whiteNoiseModelFile [MAX_STRLEN]
- double rasim
- double decsim
- int simflag
- char fitFunc [MAX_FILELEN]
- int nconstraints
- enum constraint constraints [MAX_PARAMS]
- char auto_constraints
- FitInfo fitinfo

13.16.1 Detailed Description

contains the details for a single pulsar.

Includes an array of observations and parameters

13.16.2 Member Data Documentation

13.16.2.1 double pulsar::addTNGlobalEQ

13.16.2.2 char pulsar::auto_constraints

13.16.2.3 int pulsar::AverageDMResiduals

13.16.2.4 float pulsar::AverageEpochWidth

13.16.2.5 char pulsar::AverageFlag[MAX_FLAG_LEN]

13.16.2.6 int pulsar::AverageResiduals

13.16.2.7 char pulsar::binaryModel[100]

Binary model e.g. BT/ELL1/BT2P etc.

13.16.2.8 int pulsar::bootStrap

> 0 if calculating errors using bootstrap Monte-Carlo method

13.16.2.9 int pulsar::calcShapiro

= 1 Calculate Solar system Shapiro delay (otherwise -1)

13.16.2.10 double pulsar::cgw_angpol

13.16.2.11 double pulsar::cgw_cosinc

13.16.2.12 double pulsar::cgw_h0

13.16.2.13 double pulsar::cgw_mc

13.16.2.14 double pulsar::clk_offsE[MAX_TEL_CLK_OFFS]

13.16.2.15 double pulsar::clk_offsT[MAX_TEL_CLK_OFFS]

13.16.2.16 double pulsar::clk_offsV[MAX_TEL_CLK_OFFS]

13.16.2.17 int pulsar::clkOffsN

13.16.2.18 char pulsar::clock[16]

Clock standard to use as "UTC"

13.16.2.19 char pulsar::clockFromOverride[64]

Clock code to assume TOAs are measured against (e.g. UTC to turn off clock corrections, or TDB/TCG to turn off those + Einstein delay

13.16.2.20 enum constraint pulsar::constraints[MAX_PARAMS]

Which constraints are specified

13.16.2.21 int pulsar::correctTroposphere

whether or not do correct for tropospheric delay

13.16.2.22 double** pulsar::covar

13.16.2.23 char pulsar::decjStrPost[100]

String containing RAJ and DECJ (postfit)

13.16.2.24 char pulsar::decjStrPre[100]

String containing RAJ and DECJ (prefit)

13.16.2.25 double pulsar::decsim

13.16.2.26 char pulsar::deleteFileName[100]

File name containing deleted points

13.16.2.27 int pulsar::dilateFreq

whether or not to apply SS time dilation to RFs

13.16.2.28 double pulsar::dmoffsCM[MAX_IFUNC]

13.16.2.29 double pulsar::dmoffsCM_error[MAX_IFUNC]

13.16.2.30 double pulsar::dmoffsCM_mjd[MAX_IFUNC]

13.16.2.31 double pulsar::dmoffsCM_weight[MAX_IFUNC]

13.16.2.32 int pulsar::dmoffsCMnum

13.16.2.33 double pulsar::dmoffsDM[MAX_IFUNC]

13.16.2.34 double pulsar::dmoffsDM_error[MAX_IFUNC]

13.16.2.35 double pulsar::dmoffsDM_mjd[MAX_IFUNC]

13.16.2.36 double pulsar::dmoffsDM_weight[MAX_IFUNC]

13.16.2.37 int pulsar::dmoffsDMnum

13.16.2.38 double pulsar::dmOffset

Value to add to DM flags

13.16.2.39 int pulsar::eclCoord

= 1 for ecliptic coords otherwise celestial coords

13.16.2.40 char pulsar::eopc04_file[MAX_FILELEN]

13.16.2.41 char pulsar::ephemeris[MAX_FILELEN]

13.16.2.42 char pulsar::filterStr[MAX_STRLEN]

String describing filters

13.16.2.43 double pulsar::fitChisq

Chisq value from the fit

13.16.2.44 char pulsar::fitFunc[MAX_FILELEN]

13.16.2.45 FitInfo pulsar::fitinfo

13.16.2.46 int pulsar::fitJump[MAX_JUMPS]

= 1 if fit for jump

13.16.2.47 int pulsar::fitMode

= 0 not fitting with errors, = 1 fitting with errors (MODE 1)

13.16.2.48 int pulsar::fitNfree

Number of degrees of freedom in fit

13.16.2.49 int pulsar::fitParamGlobalI[MAX_FIT]

13.16.2.50 int pulsar::fitParamGlobalK[MAX_FIT]

13.16.2.51 int pulsar::fitParamI[MAX_FIT]

13.16.2.52 int pulsar::fitParamK[MAX_FIT]

13.16.2.53 int pulsar::fixedFormat

= 0 for separate .par and .tim files, $>$ 0 indicates number of lines to skip

13.16.2.54 char pulsar::fjumpID[16]

13.16.2.55 int pulsar::globalNfit

Total number of parameters in the fit

13.16.2.56 int pulsar::globalNoConstrain

Total number of points without constraints

13.16.2.57 double pulsar::gwb_decj

13.16.2.58 double pulsar::gwb_epoch

13.16.2.59 double pulsar::gwb_geom_c

13.16.2.60 double pulsar::gwb_geom_p

13.16.2.61 double pulsar::gwb_raj

13.16.2.62 double pulsar::gwb_width

13.16.2.63 double pulsar::gwecc_dec

13.16.2.64 double pulsar::gwecc_distance

13.16.2.65 double pulsar::gwecc_e

13.16.2.66 double pulsar::gwecc_epoch

13.16.2.67 double pulsar::gwecc_inc

13.16.2.68 double pulsar::gwecc_m1

13.16.2.69 double pulsar::gwecc_m2

13.16.2.70 double pulsar::gwecc_nodes_orientation

13.16.2.71 double pulsar::gwecc_orbital_period

13.16.2.72 double pulsar::gwecc_psrdist

13.16.2.73 int pulsar::gwecc_pulsarTermOn

13.16.2.74 double pulsar::gwecc_ra

13.16.2.75 double pulsar::gwecc_redshift

13.16.2.76 double pulsar::gwecc_theta_0

13.16.2.77 double pulsar::gwecc_theta_nodes

13.16.2.78 double pulsar::gwm_decj

13.16.2.79 double pulsar::gwm_dphase

13.16.2.80 double pulsar::gwm_epoch

13.16.2.81 double pulsar::gwm_phi

13.16.2.82 double pulsar::gwm_raj

13.16.2.83 double pulsar::gwsrc_across_i

13.16.2.84 double pulsar::gwsrc_across_i_e

13.16.2.85 double pulsar::gwsrc_across_r

13.16.2.86 double pulsar::gwsrc_across_r_e

13.16.2.87 double pulsar::gwsrc_aplus_i

13.16.2.88 double pulsar::gwsrc_aplus_i_e

13.16.2.89 double pulsar::gwsrc_aplus_r

13.16.2.90 double pulsar::gwsrc_aplus_r_e

13.16.2.91 double pulsar::gwsrc_dec

13.16.2.92 double pulsar::gwsrc_epoch

13.16.2.93 double pulsar::gwsrc_psrdist

13.16.2.94 double pulsar::gwsrc_ra

13.16.2.95 double pulsar::ifunc_weights[MAX_IFUNC]

13.16.2.96 double pulsar::ifuncE[MAX_IFUNC]

13.16.2.97 int pulsar::ifuncN

13.16.2.98 double pulsar::ifuncT[MAX_IFUNC]

13.16.2.99 double pulsar::ifuncV[MAX_IFUNC]

13.16.2.100 int pulsar::ipm

= 1 if use interplanetary medium DM correction, = 0 otherwise

13.16.2.101 int pulsar::jboFormat

= 1 => JBO arrival time format and file structure (not byte swapping) = 2 => JBO format with byte swapping

13.16.2.102 char pulsar::JPL_EPHEMERIS[MAX_FILELEN]

13.16.2.103 char pulsar::jumpStr[MAX_JUMPS][MAX_STRLEN]

String describing jump

13.16.2.104 double pulsar::jumpVal[MAX_JUMPS]

Value of jump

13.16.2.105 double pulsar::jumpValErr[MAX_JUMPS]

Error on jump

13.16.2.106 char pulsar::name[100]

13.16.2.107 int pulsar::nCompanion

Number of binary companions

13.16.2.108 int pulsar::nconstraints

Number of fit constraints specified

13.16.2.109 int pulsar::nDMEvents

13.16.2.110 int pulsar::ndmx

Number of DM steps

13.16.2.111 double pulsar::ne_sw

Electron density at 1AU due to the solar wind

13.16.2.112 int pulsar::nFit

Number of points in the fit

13.16.2.113 int pulsar::nGlobal

Number of global parameters in the fit

13.16.2.114 int pulsar::nits

Number of iterations for the fit

13.16.2.115 int pulsar::nJumps

Number of jumps

13.16.2.116 int pulsar::nobs

Number of observations in .tim file

13.16.2.117 int pulsar::noWarnings

= 1, do not display warning messages

13.16.2.118 int pulsar::nParam

Number of parameters in the fit

13.16.2.119 int pulsar::nPhaseJump

Number of phase jumps

13.16.2.120 int pulsar::nQuad

13.16.2.121 int pulsar::nStorePrecision

13.16.2.122 int pulsar::nT2efac

13.16.2.123 int pulsar::nT2equad

13.16.2.124 int pulsar::nTelDX

13.16.2.125 int pulsar::nTelDY

13.16.2.126 int pulsar::nTelDZ

13.16.2.127 int pulsar::nTNBandNoise

13.16.2.128 int pulsar::nTNECORR

13.16.2.129 int pulsar::nTNEF

13.16.2.130 int pulsar::nTNEQ

13.16.2.131 int pulsar::nTNGroupNoise

13.16.2.132 int pulsar::nTNShapeletEvents

13.16.2.133 int pulsar::nTNSQ

13.16.2.134 int pulsar::nToffset

13.16.2.135 int pulsar::nWhite

13.16.2.136 int pulsar::nWhite_dm

13.16.2.137 observation* pulsar::obsn

[MAX_OBSN_VAL];

13.16.2.138 double pulsar::offset

Offset, always fitted for

13.16.2.139 double pulsar::offset_e

Error in the offset

13.16.2.140 int pulsar::outputTMatrix

13.16.2.141 parameter pulsar::param[MAX_PARAMS]

13.16.2.142 char pulsar::passStr[MAX_STRLEN]

String describing filters

13.16.2.143 longdouble pulsar::phaseJump[MAX_JUMPS]

Time of phase jump

13.16.2.144 int pulsar::phaseJumpDir[MAX_JUMPS]

Size and direction of phase jump

13.16.2.145 int pulsar::phaseJumpID[MAX_JUMPS]

ID of closest point to the phase jump

13.16.2.146 int pulsar::planetShapiro

= 1 if included otherwise 0

13.16.2.147 double pulsar::posPulsar[3]

3-vector pointing at pulsar

13.16.2.148 double pulsar::quad_across_i[MAX_QUAD]

13.16.2.149 double pulsar::quad_across_i_e[MAX_QUAD]

13.16.2.150 double pulsar::quad_across_r[MAX_QUAD]

13.16.2.151 double pulsar::quad_across_r_e[MAX_QUAD]

13.16.2.152 double pulsar::quad_aplus_i[MAX_QUAD]

13.16.2.153 double pulsar::quad_aplus_i_e[MAX_QUAD]

13.16.2.154 double pulsar::quad_aplus_r[MAX_QUAD]

13.16.2.155 double pulsar::quad_aplus_r_e[MAX_QUAD]

13.16.2.156 double pulsar::quad_ifunc_c_DEC

13.16.2.157 double pulsar::quad_ifunc_c_RA

13.16.2.158 double pulsar::quad_ifunc_geom_c

13.16.2.159 double pulsar::quad_ifunc_geom_p

13.16.2.160 double pulsar::quad_ifunc_p_DEC

13.16.2.161 double pulsar::quad_ifunc_p_RA

13.16.2.162 double pulsar::quad_ifuncE_c[MAX_IFUNC]

13.16.2.163 double pulsar::quad_ifuncE_p[MAX_IFUNC]

13.16.2.164 int pulsar::quad_ifuncN_c

13.16.2.165 int pulsar::quad_ifuncN_p

13.16.2.166 double pulsar::quad_ifuncT_c[MAX_IFUNC]

13.16.2.167 double pulsar::quad_ifuncT_p[MAX_IFUNC]

13.16.2.168 double pulsar::quad_ifuncV_c[MAX_IFUNC]

13.16.2.169 double pulsar::quad_ifuncV_p[MAX_IFUNC]

13.16.2.170 double pulsar::quadDEC

13.16.2.171 double pulsar::quadEpoch

13.16.2.172 double pulsar::quadRA

13.16.2.173 char pulsar::rajStrPost[100]

13.16.2.174 char pulsar::rajStrPre[100]

13.16.2.175 double pulsar::rasim

13.16.2.176 int pulsar::rescaleErrChisq

= 1 to rescale errors based on the reduced chisq, = 0 not to do this

13.16.2.177 double pulsar::rmsPost

13.16.2.178 double pulsar::rmsPre

13.16.2.179 char pulsar::robust

13.16.2.180 int pulsar::setTelVelX

13.16.2.181 int pulsar::setTelVelY

13.16.2.182 int pulsar::setTelVelZ

13.16.2.183 int pulsar::setUnits

13.16.2.184 int pulsar::simflag

Which fit function are we using

13.16.2.185 char pulsar::sorted

ToAs sorted Path for the file containing the corrections between observatory clocks and UTC(NIST) - set in read↩ Parfile.C char OBSERVATORY_CLOCK_2_UTC_NIST[MAX_FILELEN];

13.16.2.186 storePrecision pulsar::storePrec[MAX_STOREPRECISION]

13.16.2.187 int pulsar::swm

= 0 for basic tempo2 solar wind model, = 1 for XPY Solar wind model For whitening

13.16.2.188 int pulsar::t2cMethod

How to transform from terrestrial to celestial coords

13.16.2.189 char pulsar::T2efacFlagID[MAX_T2EFAC][MAX_FLAG_LEN]

13.16.2.190 char pulsar::T2efacFlagVal[MAX_T2EFAC][MAX_FLAG_LEN]

13.16.2.191 double pulsar::T2efacVal[MAX_T2EFAC]

13.16.2.192 char pulsar::T2equadFlagID[MAX_T2EQUAD][MAX_FLAG_LEN]

13.16.2.193 char pulsar::T2equadFlagVal[MAX_T2EQUAD][MAX_FLAG_LEN]

13.16.2.194 double pulsar::T2equadVal[MAX_T2EQUAD]

13.16.2.195 double pulsar::T2globalEfac

13.16.2.196 double pulsar::telDX_e[MAX_TEL_DX]

13.16.2.197 double pulsar::telDX_t[MAX_TEL_DX]

13.16.2.198 double pulsar::telDX_v[MAX_TEL_DX]

13.16.2.199 double pulsar::telDX_vel[MAX_TEL_DX]

13.16.2.200 double pulsar::telDX_vel_e[MAX_TEL_DX]

13.16.2.201 double pulsar::telDY_e[MAX_TEL_DY]

13.16.2.202 double pulsar::telDY_t[MAX_TEL_DY]

13.16.2.203 double pulsar::telDY_v[MAX_TEL_DY]

13.16.2.204 double pulsar::telDY_vel[MAX_TEL_DY]

13.16.2.205 double pulsar::telDY_vel_e[MAX_TEL_DY]

13.16.2.206 double pulsar::telDZ_e[MAX_TEL_DZ]

13.16.2.207 double pulsar::telDZ_t[MAX_TEL_DZ]

13.16.2.208 double pulsar::telDZ_v[MAX_TEL_DZ]

13.16.2.209 double pulsar::telDZ_vel[MAX_TEL_DZ]

13.16.2.210 double pulsar::telDZ_vel_e[MAX_TEL_DZ]

13.16.2.211 int pulsar::tempo1

= 1 if tempo1 is emulated

13.16.2.212 int pulsar::timeEphemeris

Which code to use for Einstein delay

13.16.2.213 double pulsar::TNBandDMAmp

13.16.2.214 int pulsar::TNBandDMC

13.16.2.215 double pulsar::TNBandDMGam

13.16.2.216 double pulsar::TNBandNoiseAmp[MAX_TNBN]

13.16.2.217 int pulsar::TNBandNoiseC[MAX_TNBN]

13.16.2.218 double pulsar::TNBandNoiseGam[MAX_TNBN]

13.16.2.219 double pulsar::TNBandNoiseHF[MAX_TNBN]

13.16.2.220 double pulsar::TNBandNoiseLF[MAX_TNBN]

13.16.2.221 double pulsar::TNDMAmp

13.16.2.222 int pulsar::TNDMC

13.16.2.223 double pulsar::TNDMCoeffs[200]

13.16.2.224 double pulsar::TNDMEvAmp[MAX_TNDMEv]

13.16.2.225 double pulsar::TNDMEvGam[MAX_TNDMEv]

13.16.2.226 double pulsar::TNDMEvLength[MAX_TNDMEv]

13.16.2.227 int pulsar::TNDMEvLin[MAX_TNDMEv]

13.16.2.228 int pulsar::TNDMEvOff[MAX_TNDMEv]

13.16.2.229 int pulsar::TNDMEvQuad[MAX_TNDMEv]

13.16.2.230 double pulsar::TNDMEvStart[MAX_TNDMEv]

13.16.2.231 double pulsar::TNDMGam

13.16.2.232 char pulsar::TNECORRFlagID[MAX_TNECORR][MAX_FLAG_LEN]

13.16.2.233 char pulsar::TNECORRFlagVal[MAX_TNECORR][MAX_FLAG_LEN]

13.16.2.234 double pulsar::TNECORRVal[MAX_TNECORR]

13.16.2.235 char pulsar::TNEFFlagID[MAX_TNEF][MAX_FLAG_LEN]

13.16.2.236 char pulsar::TNEFFlagVal[MAX_TNEF][MAX_FLAG_LEN]

13.16.2.237 double pulsar::TNEFVal[MAX_TNEF]

13.16.2.238 char pulsar::TNEQFlagID[MAX_TNEQ][MAX_FLAG_LEN]

13.16.2.239 char pulsar::TNEQFlagVal[MAX_TNEQ][MAX_FLAG_LEN]

13.16.2.240 double pulsar::TNEQVal[MAX_TNEQ]

13.16.2.241 double pulsar::TNGlobalEF

13.16.2.242 double pulsar::TNGlobalEQ

13.16.2.243 double pulsar::TNGroupNoiseAmp[MAX_TNGN]

13.16.2.244 int pulsar::TNGroupNoiseC[MAX_TNGN]

13.16.2.245 char pulsar::TNGroupNoiseFlagID[MAX_TNGN][MAX_FLAG_LEN]

13.16.2.246 char pulsar::TNGroupNoiseFlagVal[MAX_TNGN][MAX_FLAG_LEN]

13.16.2.247 double pulsar::TNGroupNoiseGam[MAX_TNGN]

13.16.2.248 double pulsar::TNRedAmp

13.16.2.249 int pulsar::TNRedC

13.16.2.250 double pulsar::TNRedCoeffs[200]

13.16.2.251 double pulsar::TNRedCorner

13.16.2.252 double pulsar::TNRedFLow

13.16.2.253 double pulsar::TNRedGam

13.16.2.254 double pulsar::TNShapeletEvFScale[MAX_TNDMEv]

13.16.2.255 int pulsar::TNShapeletEvN[MAX_TNDMEv]

13.16.2.256 double pulsar::TNShapeletEvPos[MAX_TNDMEv]

13.16.2.257 double pulsar::TNShapeletEvWidth[MAX_TNDMEv]

13.16.2.258 char pulsar::TNSQFlagID[MAX_TNSQ][MAX_FLAG_LEN]

13.16.2.259 char pulsar::TNSQFlagVal[MAX_TNSQ][MAX_FLAG_LEN]

13.16.2.260 double pulsar::TNSQVal[MAX_TNSQ]

13.16.2.261 int pulsar::TNsubtractDM

13.16.2.262 int pulsar::TNsubtractRed

13.16.2.263 double∗∗ pulsar::ToAextraCovar

13.16.2.264 double pulsar::tOffset[MAX_TOFFSET]

Offsets in TOAs in seconds

13.16.2.265 double pulsar::tOffset_f1[MAX_TOFFSET]

13.16.2.266 double pulsar::tOffset_f2[MAX_TOFFSET]

Range for offset to be applied

13.16.2.267 double pulsar::tOffset_t1[MAX_TOFFSET]

13.16.2.268 double pulsar::tOffset_t2[MAX_TOFFSET]

13.16.2.269 char pulsar::tOffsetFlags[MAX_TOFFSET][1000]

13.16.2.270 char pulsar::tOffsetSite[MAX_TOFFSET][100]

13.16.2.271 char pulsar::tzrsite[100]

Site-code for polyco

13.16.2.272 int pulsar::units

TDB or SI units (tempo emulation mode uses TDB) see #define definition above for possible units

13.16.2.273 int pulsar::useCalceph

13.16.2.274 int pulsar::useTNOrth

13.16.2.275 double pulsar::velPulsar[3]

3-vector giving pulsar's velocity

13.16.2.276 double pulsar::wave_cos[MAX_WHITE]

13.16.2.277 double pulsar::wave_cos_dm[MAX_WHITE]

13.16.2.278 double pulsar::wave_cos_dm_err[MAX_WHITE]

13.16.2.279 double pulsar::wave_cos_err[MAX_WHITE]

13.16.2.280 double pulsar::wave_sine[MAX_WHITE]

13.16.2.281 double pulsar::wave_sine_dm[MAX_WHITE]

13.16.2.282 double pulsar::wave_sine_dm_err[MAX_WHITE]

13.16.2.283 double pulsar::wave_sine_err[MAX_WHITE]

13.16.2.284 double pulsar::waveScale

13.16.2.285 char pulsar::whiteNoiseModelFile[MAX_STRLEN]

The documentation for this struct was generated from the following file:

- tempo2.h

13.17 storePrecision Struct Reference

```
#include <tempo2.h>
```

Public Attributes

- longdouble minPrec
- char routine [100]
- char comment [MAX_STRLEN]

13.17.1 Member Data Documentation

13.17.1.1 char storePrecision::comment[MAX_STRLEN]

13.17.1.2 longdouble storePrecision::minPrec

13.17.1.3 char storePrecision::routine[100]

The documentation for this struct was generated from the following file:

- tempo2.h

13.18 T1Polyco Struct Reference

```
#include <tempo2pred.h>
```

Public Attributes

- char psrname [64]
- char date_string [10]
- char utc_string [13]
- long double mjd_mid
- double dm
- double doppler
- double log10rms
- long double reference_phase
- long double frequency_psr_0
- char sitename [5]
- int span
- int ncoeff
- double frequency_obs
- double binary_phase
- double binary_frequency
- long double coeff [32]

13.18.1 Member Data Documentation

13.18.1.1 double T1Polyco::binary_frequency

13.18.1.2 double T1Polyco::binary_phase

13.18.1.3 long double T1Polyco::coeff[32]

13.18.1.4 char T1Polyco::date_string[10]

13.18.1.5 double T1Polyco::dm

13.18.1.6 double T1Polyco::doppler

13.18.1.7 double T1Polyco::frequency_obs

13.18.1.8 long double T1Polyco::frequency_psr_0

13.18.1.9 double T1Polyco::log10rms

13.18.1.10 long double T1Polyco::mjd_mid

13.18.1.11 int T1Polyco::ncoeff

13.18.1.12 char T1Polyco::psrname[64]

13.18.1.13 long double T1Polyco::reference_phase

13.18.1.14 char T1Polyco::sitename[5]

13.18.1.15 int T1Polyco::span

13.18.1.16 char T1Polyco::utc_string[13]

The documentation for this struct was generated from the following file:

- tempo2pred.h

13.19 T1PolycoSet Struct Reference

```
#include <tempo2pred.h>
```

Collaboration diagram for T1PolycoSet:

Public Attributes

- T1Polyco ∗ segments
- int nsegments

13.19.1 Member Data Documentation

13.19.1.1 int T1PolycoSet::nsegments

13.19.1.2 T1Polyco∗ T1PolycoSet::segments

The documentation for this struct was generated from the following file:

- tempo2pred.h

13.20 T2Predictor Struct Reference

`#include <tempo2pred.h>`

Collaboration diagram for T2Predictor:

Public Attributes

- T2PredictorKind kind
- union {
  ChebyModelSet cheby
  T1PolycoSet t1
  } modelset

13.20.1 Member Data Documentation

13.20.1.1 ChebyModelSet T2Predictor::cheby

13.20.1.2 T2PredictorKind T2Predictor::kind

13.20.1.3 union { ... } T2Predictor::modelset

13.20.1.4 T1PolycoSet T2Predictor::t1

The documentation for this struct was generated from the following file:

- tempo2pred.h

13.21 TabulatedFunction Struct Reference

`#include <tabulatedfunction.h>`

Collaboration diagram for TabulatedFunction:

Public Attributes

- char fileName [256]
- char header_line [256]
- DynamicArray samples

13.21.1 Member Data Documentation

13.21.1.1 char TabulatedFunction::fileName[256]

13.21.1.2 char TabulatedFunction::header_line[256]

13.21.1.3 DynamicArray TabulatedFunction::samples

The documentation for this struct was generated from the following file:

- tabulatedfunction.h

13.22 TabulatedFunctionSample Struct Reference

```
#include <tabulatedfunction.h>
```

Public Attributes

- double x
- double y

13.22.1 Member Data Documentation

13.22.1.1 double TabulatedFunctionSample::x

13.22.1.2 double TabulatedFunctionSample::y

The documentation for this struct was generated from the following file:

- tabulatedfunction.h

Chapter 14

File Documentation

14.1 cholesky.h File Reference

Functions

- void cholesky_readFromCovarianceFunction (double ∗∗m, const char ∗fname, double ∗resx, double ∗resy, double ∗rese, int np, int nc)
- void cholesky_covarFunc2matrix (double ∗∗m, double ∗covarFunc, int ndays, double ∗resx, double ∗resy, double ∗rese, int np, int nc)
- void cholesky_powerlawModel (double ∗∗m, double modelAlpha, double modelFc, double modelA, double ∗resx, double ∗resy, double ∗rese, int np, int nc)
- void cholesky_powerlawModel_withBeta (double ∗∗m, double modelAlpha, double beta, double modelFc, double modelA, double ∗resx, double ∗resy, double ∗rese, int np, int nc)
- int cholesky_formUinv (double ∗∗uinv, double ∗∗m, int np)
- void cholesky_dmModel (double ∗∗m, double D, double d, double ref_freq, double ∗resx, double ∗resy, double ∗rese, int np, int nc)
- void cholesky_ecm (double ∗∗m, char ∗fileName, double ∗resx, double ∗resy, double ∗rese, int np, int nc)
- void cholesky_dmModelCovarParam (double ∗∗m, double alpha, double a, double b, double ∗resx, double ∗resy, double ∗rese, int np, int nc)

14.1.1 Function Documentation

14.1.1.1 void cholesky_covarFunc2matrix (double ∗∗ *m,* double ∗ *covarFunc,* int *ndays,* double ∗ *resx,* double ∗ *resy,* double ∗ *rese,* int *np,* int *nc*)

14.1.1.2 void cholesky_dmModel (double ∗∗ *m,* double *D,* double *d,* double *ref_freq,* double ∗ *resx,* double ∗ *resy,* double ∗ *rese,* int *np,* int *nc*)

14.1.1.3 void cholesky_dmModelCovarParam (double ∗∗ *m,* double *alpha,* double *a,* double *b,* double ∗ *resx,* double ∗ *resy,* double ∗ *rese,* int *np,* int *nc*)

14.1.1.4 void cholesky_ecm (double ∗∗ *m,* char ∗ *fileName,* double ∗ *resx,* double ∗ *resy,* double ∗ *rese,* int *np,* int *nc*)

14.1.1.5 int cholesky_formUinv (double ∗∗ *uinv,* double ∗∗ *m,* int *np*)

14.1.1.6 void cholesky_powerlawModel (double ∗∗ *m,* double *modelAlpha,* double *modelFc,* double *modelA,* double ∗ *resx,* double ∗ *resy,* double ∗ *rese,* int *np,* int *nc*)

14.1.1.7 void cholesky_powerlawModel_withBeta (double ∗∗ *m,* double *modelAlpha,* double *beta,* double *modelFc,* double *modelA,* double ∗ *resx,* double ∗ *resy,* double ∗ *rese,* int *np,* int *nc*)

14.1.1.8 void cholesky_readFromCovarianceFunction (double ∗∗ *m,* const char ∗ *fname,* double ∗ *resx,* double ∗ *resy,* double ∗ *rese,* int *np,* int *nc*)

14.2 choleskyRoutines.h File Reference

```
#include "tempo2.h"
```

Include dependency graph for choleskyRoutines.h:

14.3 config.h File Reference

This graph shows which files directly or indirectly include this file:

Macros

- #define F77_FUNC(name, NAME) name ## _
- #define F77_FUNC_(name, NAME) name ## _
- #define HAVE_BLAS 1
- #define HAVE_CFITSIO 1
- #define HAVE_DLERROR 1
- #define HAVE_DLFCN_H 1
- #define HAVE_FFTW3 1
- #define HAVE_INTTYPES_H 1
- #define HAVE_LAPACK 1
- #define HAVE_LIBDL 1
- #define HAVE_LIBDLLOADER 1
- #define HAVE_LIBM 1
- #define HAVE_MEMORY_H 1
- #define HAVE_PGPLOT 1
- #define HAVE_PTHREAD 1
- #define HAVE_STDINT_H 1
- #define HAVE_STDLIB_H 1
- #define HAVE_STRINGS_H 1
- #define HAVE_STRING_H 1
- #define HAVE_SYS_STAT_H 1
- #define HAVE_SYS_TYPES_H 1
- #define HAVE_UNISTD_H 1
- #define LT_OBJDIR ".libs/"
- #define PACKAGE "tempo2"
- #define PACKAGE_BUGREPORT "george.hobbs@csiro.au"
- #define PACKAGE_NAME "Tempo2"
- #define PACKAGE_STRING "Tempo2 2016.08.0"
- #define PACKAGE_TARNAME "tempo2"
- #define PACKAGE_URL "http://www.bitbucket.org/psrsoft/tempo2"
- #define PACKAGE_VERSION "2016.08.0"
- #define STDC_HEADERS 1
- #define TEMPO2_ARCH "darwin15"
- #define VERSION "2016.08.0"
- #define _DARWIN_USE_64_BIT_INODE 1

14.3.1 Macro Definition Documentation

14.3.1.1 #define _DARWIN_USE_64_BIT_INODE 1

14.3.1.2 #define F77_FUNC(*name,* *NAME*) name ## _

14.3.1.3 #define F77_FUNC_(*name,* *NAME*) name ## _

14.3.1.4 #define HAVE_BLAS 1

14.3.1.5 #define HAVE_CFITSIO 1

14.3.1.6 #define HAVE_DLERROR 1

14.3.1.7 #define HAVE_DLFCN_H 1

14.3.1.8 #define HAVE_FFTW3 1

14.3.1.9 #define HAVE_INTTYPES_H 1

14.3.1.10 #define HAVE_LAPACK 1

14.3.1.11 #define HAVE_LIBDL 1

14.3.1.12 #define HAVE_LIBDLLOADER 1

14.3.1.13 #define HAVE_LIBM 1

14.3.1.14 #define HAVE_MEMORY_H 1

14.3.1.15 #define HAVE_PGPLOT 1

14.3.1.16 #define HAVE_PTHREAD 1

14.3.1.17 #define HAVE_STDINT_H 1

14.3.1.18 #define HAVE_STDLIB_H 1

14.3.1.19 #define HAVE_STRING_H 1

14.3.1.20 #define HAVE_STRINGS_H 1

14.3.1.21 #define HAVE_SYS_STAT_H 1

14.3.1.22 #define HAVE_SYS_TYPES_H 1

14.3.1.23 #define HAVE_UNISTD_H 1

14.3.1.24 #define LT_OBJDIR ".libs/"

14.3.1.25 #define PACKAGE "tempo2"

14.3.1.26 #define PACKAGE_BUGREPORT "george.hobbs@csiro.au"

14.3.1.27 #define PACKAGE_NAME "Tempo2"

14.3.1.28 #define PACKAGE_STRING "Tempo2 2016.08.0"

14.3.1.29 #define PACKAGE_TARNAME "tempo2"

14.3.1.30 #define PACKAGE_URL "http://www.bitbucket.org/psrsoft/tempo2"

14.3.1.31 #define PACKAGE_VERSION "2016.08.0"

14.3.1.32 #define STDC_HEADERS 1

14.3.1.33 #define TEMPO2_ARCH "darwin15"

14.3.1.34 #define VERSION "2016.08.0"

14.4 constraints.h File Reference

```
#include <string.h>
#include "tempo2.h"
```

Include dependency graph for constraints.h:

Functions

- std::string get_constraint_name (enum constraint c)
- void computeConstraintWeights (pulsar ∗psr)
- double consFunc_dmmodel_mean (pulsar ∗psr, int ipsr, int i, int k, int order)
- double consFunc_dmmodel_dm1 (pulsar ∗psr, int ipsr, int i, int k, int order)
- double consFunc_dmmodel_cw (pulsar ∗psr, int ipsr, int i, int k, int order)
- double consFunc_dmmodel_cw_year (pulsar ∗psr, int ipsr, int i, int k, int order)
- double consFunc_ifunc (pulsar ∗psr, int ipsr, int i, int k, int order)
- double consFunc_ifunc_year (pulsar ∗psr, int ipsr, int i, int k, int order)
- double consFunc_tel_dx (pulsar ∗psr, int ipsr, int i, int k, int order)
- double consFunc_tel_dy (pulsar ∗psr, int ipsr, int i, int k, int order)
- double consFunc_tel_dz (pulsar ∗psr, int ipsr, int i, int k, int order)
- double consFunc_quad_ifunc_p (pulsar ∗psr, int ipsr, int i, int k, int order)
- double consFunc_quad_ifunc_c (pulsar ∗psr, int ipsr, int i, int k, int order)
- double consFunc_qifunc_p_year (pulsar ∗psr, int ipsr, int i, int k, int order)
- double consFunc_qifunc_c_year (pulsar ∗psr, int ipsr, int i, int k, int order)
- void autosetDMCM (pulsar ∗psr, double dmstep, double cmstep, double start, double end, bool fixCMgrid)
- void CONSTRAINTfuncs (pulsar ∗psr, int ipsr, int nparams, int iconstraint, double ∗OUT)
- double standardConstraintFunctions (pulsar ∗psr, int ipsr, int iconstraint, int iparam, int constraintk, int k)

14.4.1 Function Documentation

14.4.1.1 void autosetDMCM (pulsar ∗ *psr,* double *dmstep,* double *cmstep,* double *start,* double *end,* bool *fixCMgrid*)

14.4.1.2 void computeConstraintWeights (pulsar ∗ *psr*)

14.4.1.3 double consFunc_dmmodel_cw (pulsar ∗ *psr,* int *ipsr,* int *i,* int *k,* int *order*)

14.4.1.4 double consFunc_dmmodel_cw_year (pulsar ∗ *psr,* int *ipsr,* int *i,* int *k,* int *order*)

14.4.1.5 double consFunc_dmmodel_dm1 (pulsar ∗ *psr,* int *ipsr,* int *i,* int *k,* int *order*)

14.4.1.6 double consFunc_dmmodel_mean (**pulsar** ∗ *psr,* int *ipsr,* int *i,* int *k,* int *order*)

14.4.1.7 double consFunc_ifunc (**pulsar** ∗ *psr,* int *ipsr,* int *i,* int *k,* int *order*)

14.4.1.8 double consFunc_ifunc_year (**pulsar** ∗ *psr,* int *ipsr,* int *i,* int *k,* int *order*)

14.4.1.9 double consFunc_qifunc_c_year (**pulsar** ∗ *psr,* int *ipsr,* int *i,* int *k,* int *order*)

14.4.1.10 double consFunc_qifunc_p_year (**pulsar** ∗ *psr,* int *ipsr,* int *i,* int *k,* int *order*)

14.4.1.11 double consFunc_quad_ifunc_c (**pulsar** ∗ *psr,* int *ipsr,* int *i,* int *k,* int *order*)

14.4.1.12 double consFunc_quad_ifunc_p (**pulsar** ∗ *psr,* int *ipsr,* int *i,* int *k,* int *order*)

14.4.1.13 double consFunc_tel_dx (**pulsar** ∗ *psr,* int *ipsr,* int *i,* int *k,* int *order*)

14.4.1.14 double consFunc_tel_dy (**pulsar** ∗ *psr,* int *ipsr,* int *i,* int *k,* int *order*)

14.4.1.15 double consFunc_tel_dz (**pulsar** ∗ *psr,* int *ipsr,* int *i,* int *k,* int *order*)

14.4.1.16 void CONSTRAINTfuncs (**pulsar** ∗ *psr,* int *ipsr,* int *nparams,* int *iconstraint,* double ∗ *OUT*)

14.4.1.17 std::string get_constraint_name (enum **constraint** *c*)

14.4.1.18 double standardConstraintFunctions (**pulsar** ∗ *psr,* int *ipsr,* int *iconstraint,* int *iparam,* int *constraintk,* int *k*)

14.5 documentation/1_USER_GUIDE.md File Reference

14.6 documentation/2_developers.md File Reference

14.7 documentation/3_DEVELOPER_GUIDE.md File Reference

14.8 documentation/4_directories.md File Reference

14.9 documentation/5_plugins.md File Reference

14.10 dynarr.h File Reference

`#include <stdlib.h>`
Include dependency graph for dynarr.h: This graph shows which files directly or indirectly include this file:

Classes

- struct DynamicArray

Functions

- void DynamicArray_init (DynamicArray ∗, size_t elemSize)
- void DynamicArray_resize (DynamicArray ∗, size_t nelem)
- void ∗ DynamicArray_push_back (DynamicArray ∗, void ∗elem)
- void DynamicArray_free (DynamicArray ∗)

14.10.1 Function Documentation

14.10.1.1 void DynamicArray_free (**DynamicArray** ∗)

14.10.1.2 void DynamicArray_init (**DynamicArray** ∗ , size_t *elemSize*)

14.10.1.3 void∗ DynamicArray_push_back (**DynamicArray** ∗ , void ∗ *elem*)

14.10.1.4 void DynamicArray_resize (**DynamicArray** ∗ , size_t *nelem*)

14.11 enum_str.h File Reference

Variables

- const char ∗ label_str [ ]
- const char ∗ constraint_str [ ]

14.11.1 Variable Documentation

14.11.1.1 const char∗ constraint_str[ ]

14.11.1.2 const char∗ label_str[ ]

14.12 GWsim.h File Reference

```
#include "tempo2.h"
```

Include dependency graph for GWsim.h:

Classes

- struct gwSrc
- struct gwgeneralSrc
- struct gwgenSpec

Typedefs

- typedef struct gwSrc gwSrc
- typedef struct gwgeneralSrc gwgeneralSrc
- typedef struct gwgenSpec gwgenSpec

Functions

- double Fe (double ec)
- double dadt (double ec, double a, double m1, double m2)
- double dedt (double ec, double a, double m1, double m2)
- double dtdt (double ec, double t, double p)
- double Rs (double m1)
- longdouble eccRes (pulsar ∗psr, int i, int ∗coalesceFlag, double ∗prev_p, double ∗prev_e, double ∗prev_a, double ∗prev_epoch, double ∗prev_theta)
- longdouble eccResWithEnergy (pulsar ∗psr, int i, int ∗coalesceFlag, double ∗prev_p, double ∗prev_e, double ∗prev_a, double ∗prev_epoch, double ∗prev_theta, float ∗eOut)
- void setupGW (gwSrc ∗gw)

- void matrixMult (longdouble m1[3][3], longdouble m2[3][3], longdouble out[3][3])
- longdouble dotProduct (longdouble ∗m1, longdouble ∗m2)
- void GWbackground (gwSrc ∗gw, int numberGW, long ∗idum, longdouble flo, longdouble fhi, double gwAmp, double alpha, int loglin)
- longdouble calculateResidualGW (longdouble ∗kp, gwSrc ∗gw, longdouble time, longdouble dist)
- void setupPulsar_GWsim (longdouble ra_p, longdouble dec_p, longdouble ∗kp)
- int GWbackground_read (gwSrc ∗gw, FILE ∗file, int ireal)
- void GWbackground_write (gwSrc ∗gw, FILE ∗file, int ngw, int ireal)
- double psrangle (double centre_long, double centre_lat, double psr_long, double psr_lat)
- double sphharm (int l, int m, double x)
- double Findphi (double prob, double amp, double phase)
- void setupgeneralGW (gwgeneralSrc ∗gw)
- void GWgeneralbackground (gwgeneralSrc ∗gw, int ∗numberGW, long ∗idum, longdouble flo, longdouble fhi, gwgenSpec gwAmps, int loglin)
- void GWgeneralanisotropicbackground (gwgeneralSrc ∗gw, int ∗numberGW, long ∗idum, longdouble flo, longdouble fhi, gwgenSpec gwAmps, int loglin, double ∗∗∗harmlist, int ∗nharms)
- void GWanisotropicbackground (gwSrc ∗gw, int numberGW, long ∗idum, longdouble flo, longdouble fhi, double gwAmp, double alpha, int loglin, double ∗∗harmlist, int nharms)
- void GWdipolebackground (gwSrc ∗gw, int numberGW, long ∗idum, longdouble flo, longdouble fhi, double gwAmp, double alpha, int loglin, double ∗dipoleamps)
- longdouble calculateResidualgeneralGW (longdouble ∗kp, gwgeneralSrc ∗gw, longdouble time, longdouble dist)
- int GWgeneralbackground_read (gwgeneralSrc ∗gw, FILE ∗file, int ireal)
- void GWgeneralbackground_write (gwgeneralSrc ∗gw, FILE ∗file, int ngw, int ireal)

14.12.1 Typedef Documentation

14.12.1.1 typedef struct gwgeneralSrc gwgeneralSrc

14.12.1.2 typedef struct gwgenSpec gwgenSpec

14.12.1.3 typedef struct gwSrc gwSrc

14.12.2 Function Documentation

14.12.2.1 longdouble calculateResidualgeneralGW (longdouble ∗ *kp,* gwgeneralSrc ∗ *gw,* longdouble *time,* longdouble *dist*)

14.12.2.2 longdouble calculateResidualGW (longdouble ∗ *kp,* gwSrc ∗ *gw,* longdouble *time,* longdouble *dist*)

14.12.2.3 double dadt (double *ec,* double *a,* double *m1,* double *m2*)

14.12.2.4 double dedt (double *ec,* double *a,* double *m1,* double *m2*)

14.12.2.5 longdouble dotProduct (longdouble ∗ *m1,* longdouble ∗ *m2*)

14.12.2.6 double dtdt (double *ec,* double *t,* double *p*)

14.12.2.7 longdouble eccRes (pulsar ∗ *psr,* int *i,* int ∗ *coalesceFlag,* double ∗ *prev_p,* double ∗ *prev_e,* double ∗ *prev_a,* double ∗ *prev_epoch,* double ∗ *prev_theta*)

14.12.2.8 longdouble eccResWithEnergy (pulsar ∗ *psr,* int *i,* int ∗ *coalesceFlag,* double ∗ *prev_p,* double ∗ *prev_e,* double ∗ *prev_a,* double ∗ *prev_epoch,* double ∗ *prev_theta,* float ∗ *eOut*)

14.12.2.9 double Fe (double *ec*)

14.12.2.10 double Findphi (double *prob,* double *amp,* double *phase*)

14.12.2.11 void GWanisotropicbackground (gwSrc ∗ *gw,* int *numberGW,* long ∗ *idum,* longdouble *flo,* longdouble *fhi,* double *gwAmp,* double *alpha,* int *loglin,* double ∗∗ *harmlist,* int *nharms*)

14.12.2.12 void GWbackground (gwSrc ∗ *gw,* int *numberGW,* long ∗ *idum,* longdouble *flo,* longdouble *fhi,* double *gwAmp,* double *alpha,* int *loglin*)

14.12.2.13 int GWbackground_read (gwSrc ∗ *gw,* FILE ∗ *file,* int *ireal*)

14.12.2.14 void GWbackground_write (gwSrc ∗ *gw,* FILE ∗ *file,* int *ngw,* int *ireal*)

14.12.2.15 void GWdipolebackground (gwSrc ∗ *gw,* int *numberGW,* long ∗ *idum,* longdouble *flo,* longdouble *fhi,* double *gwAmp,* double *alpha,* int *loglin,* double ∗ *dipoleamps*)

14.12.2.16 void GWgeneralanisotropicbackground (gwgeneralSrc ∗ *gw,* int ∗ *numberGW,* long ∗ *idum,* longdouble *flo,* longdouble *fhi,* gwgenSpec *gwAmps,* int *loglin,* double ∗∗∗ *harmlist,* int ∗ *nharms*)

14.12.2.17 void GWgeneralbackground (gwgeneralSrc ∗ *gw,* int ∗ *numberGW,* long ∗ *idum,* longdouble *flo,* longdouble *fhi,* gwgenSpec *gwAmps,* int *loglin*)

14.12.2.18 int GWgeneralbackground_read (gwgeneralSrc ∗ *gw,* FILE ∗ *file,* int *ireal*)

14.12.2.19 void GWgeneralbackground_write (gwgeneralSrc ∗ *gw,* FILE ∗ *file,* int *ngw,* int *ireal*)

14.12.2.20 void matrixMult (longdouble *m1[3][3],* longdouble *m2[3][3],* longdouble *out[3][3]*)

14.12.2.21 double psrangle (double *centre_long,* double *centre_lat,* double *psr_long,* double *psr_lat*)

14.12.2.22 double Rs (double *m1*)

14.12.2.23 void setupgeneralGW (gwgeneralSrc ∗ *gw*)

14.12.2.24 void setupGW (gwSrc ∗ *gw*)

14.12.2.25 void setupPulsar_GWsim (longdouble *ra_p,* longdouble *dec_p,* longdouble ∗ *kp*)

14.12.2.26 double sphharm (int *l,* int *m,* double *x*)

14.13 ifteph.h File Reference

```
#include "tempo2.h"
```

Include dependency graph for ifteph.h:

Macros

- #define IFTE_JD0 2443144.5003725 /∗ Epoch of TCB, TCG and TT ∗/
- #define IFTE_MJD0 43144.0003725
- #define IFTE_TEPH0 -65.564518e-6
- #define IFTE_LC 1.48082686742e-8
- #define IFTE_KM1 1.55051979176e-8
- #define IFTE_K (((longdouble)1.0) + ((longdouble)IFTE_KM1)) /∗ needs quad precision ∗/

Functions

- void IFTE_init (const char ∗fname)
- void IFTE_get_DeltaT_DeltaTDot (double Teph0, double Teph1, double ∗DeltaT, double ∗DeltaTDot)
- double IFTE_DeltaT (double Teph0, double Teph1)
- double IFTE_DeltaTDot (double Teph0, double Teph1)
- void IFTE_close_file ()
- void IFTE_get_vE_vEDot (double Teph0, double Teph1, double ∗ve, double ∗vEDot)
- void IFTE_get_vE (double Teph0, double Teph1, double ∗vE)
- void IFTE_get_vEDot (double Teph0, double Teph1, double ∗vEDot)

14.13.1 Macro Definition Documentation

14.13.1.1 #define IFTE_JD0 2443144.5003725 /∗ Epoch of TCB, TCG and TT ∗/

14.13.1.2 #define IFTE_K (((longdouble)1.0) + ((longdouble)IFTE_KM1)) /∗ needs quad precision ∗/

14.13.1.3 #define IFTE_KM1 1.55051979176e-8

14.13.1.4 #define IFTE_LC 1.48082686742e-8

14.13.1.5 #define IFTE_MJD0 43144.0003725

14.13.1.6 #define IFTE_TEPH0 -65.564518e-6

14.13.2 Function Documentation

14.13.2.1 void IFTE_close_file ( )

14.13.2.2 double IFTE_DeltaT (double *Teph0,* double *Teph1*)

14.13.2.3 double IFTE_DeltaTDot (double *Teph0,* double *Teph1*)

14.13.2.4 void IFTE_get_DeltaT_DeltaTDot (double *Teph0,* double *Teph1,* double ∗ *DeltaT,* double ∗ *DeltaTDot*)

14.13.2.5 void IFTE_get_vE (double *Teph0,* double *Teph1,* double ∗ *vE*)

14.13.2.6 void IFTE_get_vE_vEDot (double *Teph0,* double *Teph1,* double ∗ *ve,* double ∗ *vEDot*)

14.13.2.7 void IFTE_get_vEDot (double *Teph0,* double *Teph1,* double ∗ *vEDot*)

14.13.2.8 void IFTE_init (const char ∗ *fname*)

14.14 jpl_int.h File Reference

Classes

- struct interpolation_info
- struct jpl_eph_data

Macros

- #define JPL_HEADER_SIZE (5 ∗ sizeof(double) + 41 ∗ sizeof(int32_t))
- #define MAX_CHEBY 18

14.14.1 Macro Definition Documentation

14.14.1.1 #define JPL_HEADER_SIZE (5 ∗ sizeof(double) + 41 ∗ sizeof(int32_t))

14.14.1.2 #define MAX_CHEBY 18

14.15 jpleph.h File Reference

Macros

- #define DLL_FUNC
- #define JPL_EPHEM_START_JD 0
- #define JPL_EPHEM_END_JD 8
- #define JPL_EPHEM_STEP 16
- #define JPL_EPHEM_N_CONSTANTS 24
- #define JPL_EPHEM_AU_IN_KM 28
- #define JPL_EPHEM_EARTH_MOON_RATIO 36
- #define JPL_EPHEM_IPT_ARRAY 44
- #define JPL_EPHEM_EPHEMERIS_VERSION 224
- #define JPL_EPHEM_KERNEL_SIZE 228
- #define JPL_EPHEM_KERNEL_RECORD_SIZE 232
- #define JPL_EPHEM_KERNEL_NCOEFF 236
- #define JPL_EPHEM_KERNEL_SWAP_BYTES 240
- #define JPL_EPH_OUTSIDE_RANGE (-1)
- #define JPL_EPH_READ_ERROR (-2)
- #define JPL_EPH_QUANTITY_NOT_IN_EPHEMERIS (-3)
- #define JPL_EPH_INVALID_INDEX (-5)
- #define JPL_EPH_FSEEK_ERROR (-6)
- #define JPL_INIT_NO_ERROR 0
- #define JPL_INIT_FILE_NOT_FOUND -1
- #define JPL_INIT_FSEEK_FAILED -2
- #define JPL_INIT_FREAD_FAILED -3
- #define JPL_INIT_FREAD2_FAILED -4
- #define JPL_INIT_FREAD5_FAILED -10
- #define JPL_INIT_FILE_CORRUPT -5
- #define JPL_INIT_MEMORY_FAILURE -6
- #define JPL_INIT_FREAD3_FAILED -7
- #define JPL_INIT_FREAD4_FAILED -8
- #define JPL_INIT_NOT_CALLED -9
- #define jpl_get_pvsun(ephem) ((double ∗)((char ∗)ephem + 248))

Functions

- void ∗DLL_FUNC jpl_init_ephemeris (const char ∗ephemeris_filename, char nam[ ][6], double ∗val)
- void DLL_FUNC jpl_close_ephemeris (void ∗ephem)
- int DLL_FUNC jpl_state (void ∗ephem, const double et, const int list[14], double pv[ ][6], double nut[4], const int bary)
- int DLL_FUNC jpl_pleph (void ∗ephem, const double et, const int ntarg, const int ncent, double rrd[ ], const int calc_velocity)
- double DLL_FUNC jpl_get_double (const void ∗ephem, const int value)
- long DLL_FUNC jpl_get_long (const void ∗ephem, const int value)
- int DLL_FUNC make_sub_ephem (void ∗ephem, const char ∗sub_filename, const double start_jd, const double end_jd)
- double DLL_FUNC jpl_get_constant (const int idx, void ∗ephem, char ∗constant_name)
- int DLL_FUNC jpl_init_error_code (void)

14.15.1 Macro Definition Documentation

14.15.1.1 #define DLL_FUNC

14.15.1.2 #define JPL_EPH_FSEEK_ERROR (-6)

14.15.1.3 #define JPL_EPH_INVALID_INDEX (-5)

14.15.1.4 #define JPL_EPH_OUTSIDE_RANGE (-1)

14.15.1.5 #define JPL_EPH_QUANTITY_NOT_IN_EPHEMERIS (-3)

14.15.1.6 #define JPL_EPH_READ_ERROR (-2)

14.15.1.7 #define JPL_EPHEM_AU_IN_KM 28

14.15.1.8 #define JPL_EPHEM_EARTH_MOON_RATIO 36

14.15.1.9 #define JPL_EPHEM_END_JD 8

14.15.1.10 #define JPL_EPHEM_EPHEMERIS_VERSION 224

14.15.1.11 #define JPL_EPHEM_IPT_ARRAY 44

14.15.1.12 #define JPL_EPHEM_KERNEL_NCOEFF 236

14.15.1.13 #define JPL_EPHEM_KERNEL_RECORD_SIZE 232

14.15.1.14 #define JPL_EPHEM_KERNEL_SIZE 228

14.15.1.15 #define JPL_EPHEM_KERNEL_SWAP_BYTES 240

14.15.1.16 #define JPL_EPHEM_N_CONSTANTS 24

14.15.1.17 #define JPL_EPHEM_START_JD 0

14.15.1.18 #define JPL_EPHEM_STEP 16

14.15.1.19 #define jpl_get_pvsun(*ephem*) ((double ∗)((char ∗)ephem + 248))

14.15.1.20 #define JPL_INIT_FILE_CORRUPT -5

14.15.1.21 #define JPL_INIT_FILE_NOT_FOUND -1

14.15.1.22 #define JPL_INIT_FREAD2_FAILED -4

14.15.1.23 #define JPL_INIT_FREAD3_FAILED -7

14.15.1.24 #define JPL_INIT_FREAD4_FAILED -8

14.15.1.25 #define JPL_INIT_FREAD5_FAILED -10

14.15.1.26 #define JPL_INIT_FREAD_FAILED -3

14.15.1.27 #define JPL_INIT_FSEEK_FAILED -2

14.15.1.28 #define JPL_INIT_MEMORY_FAILURE -6

14.15.1.29 #define JPL_INIT_NO_ERROR 0

14.15.1.30 #define JPL_INIT_NOT_CALLED -9

14.15.2 Function Documentation

14.15.2.1 void **DLL_FUNC** jpl_close_ephemeris (void ∗ *ephem*)

14.15.2.2 double **DLL_FUNC** jpl_get_constant (const int *idx,* void ∗ *ephem,* char ∗ *constant_name*)

14.15.2.3 double **DLL_FUNC** jpl_get_double (const void ∗ *ephem,* const int *value*)

14.15.2.4 long **DLL_FUNC** jpl_get_long (const void ∗ *ephem,* const int *value*)

14.15.2.5 void∗ **DLL_FUNC** jpl_init_ephemeris (const char ∗ *ephemeris_filename,* char *nam[ ][6],* double ∗ *val*)

14.15.2.6 int **DLL_FUNC** jpl_init_error_code (void)

14.15.2.7 int **DLL_FUNC** jpl_pleph (void ∗ *ephem,* const double *et,* const int *ntarg,* const int *ncent,* double *rrd[ ],* const int *calc_velocity*)

14.15.2.8 int **DLL_FUNC** jpl_state (void ∗ *ephem,* const double *et,* const int *list[14],* double *pv[ ][6],* double *nut[4],* const int *bary*)

14.15.2.9 int **DLL_FUNC** make_sub_ephem (void ∗ *ephem,* const char ∗ *sub_filename,* const double *start_jd,* const double *end_jd*)

14.16 read_fortran.h File Reference

```
#include <stdio.h>
#include <string.h>
```

Include dependency graph for read_fortran.h:

Functions

- int open_file (char ∗fname)
- void close_file ()
- void read_character (int len, char ∗str)
- char read_char ()
- int read_int ()
- float read_float ()
- double read_double ()
- int read_record_int ()

Variables

- FILE ∗ c_fileptr
- int swapByte

14.16.1 Function Documentation

14.16.1.1 void close_file ( )

14.16.1.2 int open_file (char ∗ *fname*)

14.16.1.3 char read_char ( )

14.16.1.4 void read_character (int *len,* char ∗ *str*)

14.16.1.5 double read_double ( )

14.16.1.6 float read_float ( )

14.16.1.7 int read_int ( )

14.16.1.8 int read_record_int ( )

14.16.2 Variable Documentation

14.16.2.1 FILE∗ c_fileptr

14.16.2.2 int swapByte

14.17 read_fortran2.h File Reference

```
#include <stdio.h>
#include <string.h>
```

Include dependency graph for read_fortran2.h:

Functions

- void open_file2 (char ∗fname, int ∗swap)
- void close_file2 ()
- void read_character2 (int len, char ∗str)
- int read_int2 ()
- float read_float2 ()
- double read_double2 ()
- int read_record_int2 ()

Variables

- FILE ∗ c_fileptr2
- int swapByte2

14.17.1 Function Documentation

14.17.1.1 void close_file2 ( )

14.17.1.2 void open_file2 (char ∗ *fname,* int ∗ *swap*)

14.17.1.3 void read_character2 (int *len,* char ∗ *str*)

14.17.1.4 double read_double2 ( )

14.17.1.5 float read_float2 ( )

14.17.1.6 int read_int2 ( )

14.17.1.7 int read_record_int2 ( )

14.17.2 Variable Documentation

14.17.2.1 FILE∗ c_fileptr2

14.17.2.2 int swapByte2

14.18 README.md File Reference

14.19 T2accel.h File Reference

```
#include "config.h"
```

Include dependency graph for T2accel.h:

Macros

- #define ACCEL_UINV
- #define ACCEL_LSQ
- #define ACCEL_MULTMATRIX

Functions

- int accel_uinv (double ∗_m, int n)
- double accel_lsq_qr (double ∗∗dm, double ∗data, double ∗oparm, int ndata, int nparam, double ∗∗Ocvm)
- void accel_multMatrixVec (double ∗m1, double ∗v, int ndata, int npol, double ∗out)
- void accel_multMatrix (double ∗m1, double ∗m2, int ndata, int ndata2, int npol, double ∗out)

Variables

- char useT2accel

14.19.1 Macro Definition Documentation

14.19.1.1 #define ACCEL_LSQ

14.19.1.2 #define ACCEL_MULTMATRIX

14.19.1.3 #define ACCEL_UINV

14.19.2 Function Documentation

14.19.2.1 double accel_lsq_qr (double ∗∗ *dm,* double ∗ *data,* double ∗ *oparm,* int *ndata,* int *nparam,* double ∗∗ *Ocvm*)

14.19.2.2 void accel_multMatrix (double ∗ *m1,* double ∗ *m2,* int *ndata,* int *ndata2,* int *npol,* double ∗ *out*)

14.19.2.3 void accel_multMatrixVec (double ∗ *m1,* double ∗ *v,* int *ndata,* int *npol,* double ∗ *out*)

14.19.2.4 int accel_uinv (double ∗ *_m,* int *n*)

14.19.3 Variable Documentation

14.19.3.1 char useT2accel

14.20 t2fit.h File Reference

```
#include <tempo2.h>
```

Include dependency graph for t2fit.h:

Functions

- void t2Fit (pulsar ∗psr, unsigned int npsr, const char ∗covarFuncFile)
- unsigned int t2Fit_getFitData (pulsar ∗psr, double ∗x, double ∗y, double ∗e, int ∗ip)
- void t2Fit_fillGlobalFitInfo (pulsar ∗psr, unsigned int npsr, FitInfo &OUT)
- void t2Fit_fillFitInfo (pulsar ∗psr, FitInfo &OUT, const FitInfo &globals)
- void t2Fit_buildDesignMatrix (pulsar ∗psr, int ipsr, double x, int ipos, double ∗afunc)
- void t2Fit_buildConstraintsMatrix (pulsar ∗psr, int ipsr, int iconstraint, double ∗afunc)
- void t2Fit_updateParameters (pulsar ∗psr, int ipsr, double ∗val, double ∗error)

14.20.1 Function Documentation

14.20.1.1 void t2Fit (pulsar ∗ *psr,* unsigned int *npsr,* const char ∗ *covarFuncFile*)

14.20.1.2 void t2Fit_buildConstraintsMatrix (pulsar ∗ *psr,* int *ipsr,* int *iconstraint,* double ∗ *afunc*)

14.20.1.3 void t2Fit_buildDesignMatrix (pulsar ∗ *psr,* int *ipsr,* double *x,* int *ipos,* double ∗ *afunc*)

14.20.1.4 void t2Fit_fillFitInfo (pulsar ∗ *psr,* FitInfo & *OUT,* const FitInfo & *globals*)

14.20.1.5 void t2Fit_fillGlobalFitInfo (pulsar ∗ *psr,* unsigned int *npsr,* FitInfo & *OUT*)

14.20.1.6 unsigned int t2Fit_getFitData (pulsar ∗ *psr,* double ∗ *x,* double ∗ *y,* double ∗ *e,* int ∗ *ip*)

14.20.1.7 void t2Fit_updateParameters (pulsar ∗ *psr,* int *ipsr,* double ∗ *val,* double ∗ *error*)

14.21 t2fit_dmmodel.h File Reference

```
#include "tempo2.h"
```

Include dependency graph for t2fit_dmmodel.h: This graph shows which files directly or indirectly include this file:

Functions

- double t2FitFunc_dmmodelDM (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_dmmodelDM (pulsar ∗psr, int ipsr, param_label label, int k, double val, double err)
- double t2FitFunc_dmmodelCM (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_dmmodelCM (pulsar ∗psr, int ipsr, param_label label, int k, double val, double err)

14.21.1 Function Documentation

14.21.1.1 double t2FitFunc_dmmodelCM (**pulsar** ∗ *psr,* int *ipsr,* double *x,* int *ipos,* **param_label** *label,* int *k*)

14.21.1.2 double t2FitFunc_dmmodelDM (**pulsar** ∗ *psr,* int *ipsr,* double *x,* int *ipos,* **param_label** *label,* int *k*)

14.21.1.3 void t2UpdateFunc_dmmodelCM (**pulsar** ∗ *psr,* int *ipsr,* **param_label** *label,* int *k,* double *val,* double *err*)

14.21.1.4 void t2UpdateFunc_dmmodelDM (**pulsar** ∗ *psr,* int *ipsr,* **param_label** *label,* int *k,* double *val,* double *err*)

14.22 t2fit_dmother.h File Reference

```
#include "tempo2.h"
```

Include dependency graph for t2fit_dmother.h: This graph shows which files directly or indirectly include this file:

Functions

- double t2FitFunc_dmx (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- double t2FitFunc_dmsinusoids (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- double t2FitFunc_fd (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- double t2FitFunc_fddc (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)

14.22.1 Function Documentation

14.22.1.1 double t2FitFunc_dmsinusoids (**pulsar** ∗ *psr,* int *ipsr,* double *x,* int *ipos,* **param_label** *label,* int *k*)

14.22.1.2 double t2FitFunc_dmx (**pulsar** ∗ *psr,* int *ipsr,* double *x,* int *ipos,* **param_label** *label,* int *k*)

14.22.1.3 double t2FitFunc_fd (**pulsar** ∗ *psr,* int *ipsr,* double *x,* int *ipos,* **param_label** *label,* int *k*)

14.22.1.4 double t2FitFunc_fddc (**pulsar** ∗ *psr,* int *ipsr,* double *x,* int *ipos,* **param_label** *label,* int *k*)

14.23 t2fit_fitwaves.h File Reference

This graph shows which files directly or indirectly include this file:

Functions

- double t2FitFunc_fitwaves (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_fitwaves (pulsar ∗psr, int ipsr, param_label label, int k, double val, double err)

14.23.1 Function Documentation

14.23.1.1 double t2FitFunc_fitwaves (**pulsar** ∗ *psr,* int *ipsr,* double *x,* int *ipos,* **param_label** *label,* int *k*)

14.23.1.2 void t2UpdateFunc_fitwaves (**pulsar** ∗ *psr,* int *ipsr,* **param_label** *label,* int *k,* double *val,* double *err*)

14.24 t2fit_glitch.h File Reference

This graph shows which files directly or indirectly include this file:

Functions

- double t2FitFunc_stdGlitch (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_stdGlitch (pulsar ∗psr, int ipsr, param_label label, int k, double val, double err)

14.24.1 Function Documentation

14.24.1.1 double t2FitFunc_stdGlitch (**pulsar** ∗ *psr,* int *ipsr,* double *x,* int *ipos,* **param_label** *label,* int *k*)

14.24.1.2 void t2UpdateFunc_stdGlitch (**pulsar** ∗ *psr,* int *ipsr,* **param_label** *label,* int *k,* double *val,* double *err*)

14.25 t2fit_ifunc.h File Reference

`#include "tempo2.h"`
Include dependency graph for t2fit_ifunc.h: This graph shows which files directly or indirectly include this file:

Functions

- double ifunc (const double ∗mjd, const double t, const int N, const int k)
- double sinfunc (const double ∗T, const double t, const int k)
- double t2FitFunc_sifunc (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- double t2FitFunc_ifunc (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_ifunc (pulsar ∗psr, int ipsr, param_label label, int k, double val, double err)

14.25.1 Function Documentation

14.25.1.1 double ifunc (const double ∗ *mjd,* const double *t,* const int *N,* const int *k*)

14.25.1.2 double sinfunc (const double ∗ *T,* const double *t,* const int *k*)

14.25.1.3 double t2FitFunc_ifunc (**pulsar** ∗ *psr,* int *ipsr,* double *x,* int *ipos,* **param_label** *label,* int *k*)

14.25.1.4 double t2FitFunc_sifunc (**pulsar** ∗ *psr,* int *ipsr,* double *x,* int *ipos,* **param_label** *label,* int *k*)

14.25.1.5 void t2UpdateFunc_ifunc (**pulsar** ∗ *psr,* int *ipsr,* **param_label** *label,* int *k,* double *val,* double *err*)

14.26 t2fit_position.h File Reference

`#include <tempo2.h>`
Include dependency graph for t2fit_position.h: This graph shows which files directly or indirectly include this file:

Functions

- double t2FitFunc_stdPosition (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_stdPosition (pulsar ∗psr, int ipsr, param_label label, int k, double val, double err)

14.26.1 Function Documentation

14.26.1.1 double t2FitFunc_stdPosition (**pulsar** ∗ *psr,* int *ipsr,* double *x,* int *ipos,* **param_label** *label,* int *k*)

14.26.1.2 void t2UpdateFunc_stdPosition (**pulsar** ∗ *psr,* int *ipsr,* **param_label** *label,* int *k,* double *val,* double *err*)

14.27 t2fit_stdFitFuncs.h File Reference

```
#include <tempo2.h>
#include "t2fit_position.h"
#include "t2fit_fitwaves.h"
#include "t2fit_glitch.h"
#include "t2fit_ifunc.h"
#include "t2fit_dmmodel.h"
#include "t2fit_dmother.h"
```

Include dependency graph for t2fit_stdFitFuncs.h:

Functions

- void t2UpdateFunc_simpleAdd (pulsar ∗psr, int ipsr, param_label label, int k, double val, double error)
- void t2UpdateFunc_simpleMinus (pulsar ∗psr, int ipsr, param_label label, int k, double val, double error)
- double t2FitFunc_zero (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_zero (pulsar ∗psr, int ipsr, param_label label, int k, double val, double err)
- double t2FitFunc_stdFreq (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_stdFreq (pulsar ∗psr, int ipsr, param_label label, int k, double val, double err)
- double t2FitFunc_binaryModels (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_binaryModels (pulsar ∗psr, int ipsr, param_label label, int k, double val, double err)
- double t2FitFunc_planet (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_planet (pulsar ∗psr, int ipsr, param_label label, int k, double val, double err)
- double t2FitFunc_stdDm (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- double t2FitFunc_stdGravWav (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_stdGravWav (pulsar ∗psr, int ipsr, param_label label, int k, double val, double err)
- double t2FitFunc_telPos (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_telPos (pulsar ∗psr, int ipsr, param_label label, int k, double val, double err)
- double t2FitFunc_ifunc (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_ifunc (pulsar ∗psr, int ipsr, param_label label, int k, double val, double err)
- double t2FitFunc_jump (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_jump (pulsar ∗psr, int ipsr, param_label label, int k, double val, double err)
- double t2FitFunc_notImplemented (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_notImplemented (pulsar ∗psr, int ipsr, param_label label, int k, double val, double err)

14.27.1 Function Documentation

14.27.1.1 double t2FitFunc_binaryModels (pulsar ∗ *psr*, int *ipsr*, double *x*, int *ipos*, param_label *label*, int *k*)

14.27.1.2 double t2FitFunc_ifunc (pulsar ∗ *psr*, int *ipsr*, double *x*, int *ipos*, param_label *label*, int *k*)

14.27.1.3 double t2FitFunc_jump (pulsar ∗ *psr*, int *ipsr*, double *x*, int *ipos*, param_label *label*, int *k*)

14.27.1.4 double t2FitFunc_notImplemented (pulsar ∗ *psr*, int *ipsr*, double *x*, int *ipos*, param_label *label*, int *k*)

14.27.1.5 double t2FitFunc_planet (pulsar ∗ *psr*, int *ipsr*, double *x*, int *ipos*, param_label *label*, int *k*)

14.27.1.6 double t2FitFunc_stdDm (pulsar ∗ *psr*, int *ipsr*, double *x*, int *ipos*, param_label *label*, int *k*)

14.27.1.7 double t2FitFunc_stdFreq (pulsar ∗ *psr*, int *ipsr*, double *x*, int *ipos*, param_label *label*, int *k*)

14.27.1.8 double t2FitFunc_stdGravWav (pulsar ∗ *psr*, int *ipsr*, double *x*, int *ipos*, param_label *label*, int *k*)

14.27.1.9 double t2FitFunc_telPos (pulsar ∗ *psr*, int *ipsr*, double *x*, int *ipos*, param_label *label*, int *k*)

14.27.1.10 double t2FitFunc_zero (**pulsar** ∗ *psr,* int *ipsr,* double *x,* int *ipos,* **param_label** *label,* int *k*)

14.27.1.11 void t2UpdateFunc_binaryModels (**pulsar** ∗ *psr,* int *ipsr,* **param_label** *label,* int *k,* double *val,* double *err*)

14.27.1.12 void t2UpdateFunc_ifunc (**pulsar** ∗ *psr,* int *ipsr,* **param_label** *label,* int *k,* double *val,* double *err*)

14.27.1.13 void t2UpdateFunc_jump (**pulsar** ∗ *psr,* int *ipsr,* **param_label** *label,* int *k,* double *val,* double *err*)

14.27.1.14 void t2UpdateFunc_notImplemented (**pulsar** ∗ *psr,* int *ipsr,* **param_label** *label,* int *k,* double *val,* double *err*)

14.27.1.15 void t2UpdateFunc_planet (**pulsar** ∗ *psr,* int *ipsr,* **param_label** *label,* int *k,* double *val,* double *err*)

14.27.1.16 void t2UpdateFunc_simpleAdd (**pulsar** ∗ *psr,* int *ipsr,* **param_label** *label,* int *k,* double *val,* double *error*)

14.27.1.17 void t2UpdateFunc_simpleMinus (**pulsar** ∗ *psr,* int *ipsr,* **param_label** *label,* int *k,* double *val,* double *error*)

14.27.1.18 void t2UpdateFunc_stdFreq (**pulsar** ∗ *psr,* int *ipsr,* **param_label** *label,* int *k,* double *val,* double *err*)

14.27.1.19 void t2UpdateFunc_stdGravWav (**pulsar** ∗ *psr,* int *ipsr,* **param_label** *label,* int *k,* double *val,* double *err*)

14.27.1.20 void t2UpdateFunc_telPos (**pulsar** ∗ *psr,* int *ipsr,* **param_label** *label,* int *k,* double *val,* double *err*)

14.27.1.21 void t2UpdateFunc_zero (**pulsar** ∗ *psr,* int *ipsr,* **param_label** *label,* int *k,* double *val,* double *err*)

14.28 T2toolkit.h File Reference

Set of routines that are commonly used in tempo2 and/or its plugins.

Functions

- void TKconvertFloat1 (double ∗x, float ∗ox, int n)
- void TKconvertFloat2 (double ∗x, double ∗y, float ∗ox, float ∗oy, int n)
- float TKfindMin_f (float ∗x, int n)
- float TKfindMedian_f (float ∗val, int count)
- double TKfindMedian_d (double ∗val, int count)
- float TKfindRMS_f (float ∗x, int n)
- double TKfindRMS_d (double ∗x, int n)
- float TKfindRMSweight_d (double ∗x, double ∗e, int n)
- float TKfindMax_f (float ∗x, int n)
- float TKmean_f (float ∗x, int n)
- double TKmean_d (double ∗x, int n)
- double TKvariance_d (double ∗x, int n)
- double TKrange_d (double ∗x, int n)
- float TKrange_f (float ∗x, int n)
- double TKfindMin_d (double ∗x, int n)
- double TKfindMax_d (double ∗x, int n)
- double TKsign_d (double a, double b)
- double TKretMax_d (double a, double b)
- double TKretMin_d (double a, double b)
- float TKretMax_f (float a, float b)
- float TKretMin_f (float a, float b)
- int TKretMin_i (int a, int b)
- void TKsort_f (float ∗val, int nobs)
- void TKsort_d (double ∗val, int nobs)

- void TKsort_2f (float ∗val, float ∗val2, int nobs)
- void TKsort_3d (double ∗val, double ∗val2, double ∗val3, int nobs)
- void TKzeromean_d (int n, double ∗y)
- double TKranDev (long ∗seed)
- double TKgaussDev (long ∗seed)
- long TKsetSeed ()
- void init_genrand (unsigned long s)
- unsigned long genrand_int32 (void)
- double genrand_real1 (void)

14.28.1 Detailed Description

Set of routines that are commonly used in tempo2 and/or its plugins.

These routines are mainly stand-alone functions and exist for float and double precision variables

G. Hobbs: v2, 31 Dec 2008. Complete rewrite of the routines

NOTES: Related toolkits include: TKspectrum.h: contains routines for spectral estimation TKfit.h: contains routines for fitting

14.28.2 Function Documentation

14.28.2.1 unsigned long genrand_int32 (void)

14.28.2.2 double genrand_real1 (void)

14.28.2.3 void init_genrand (unsigned long *s*)

14.28.2.4 void TKconvertFloat1 (double ∗ *x,* float ∗ *ox,* int *n*)

14.28.2.5 void TKconvertFloat2 (double ∗ *x,* double ∗ *y,* float ∗ *ox,* float ∗ *oy,* int *n*)

14.28.2.6 double TKfindMax_d (double ∗ *x,* int *n*)

14.28.2.7 float TKfindMax_f (float ∗ *x,* int *n*)

14.28.2.8 double TKfindMedian_d (double ∗ *val,* int *count*)

14.28.2.9 float TKfindMedian_f (float ∗ *val,* int *count*)

14.28.2.10 double TKfindMin_d (double ∗ *x,* int *n*)

14.28.2.11 float TKfindMin_f (float ∗ *x,* int *n*)

14.28.2.12 double TKfindRMS_d (double ∗ *x,* int *n*)

14.28.2.13 float TKfindRMS_f (float ∗ *x,* int *n*)

14.28.2.14 float TKfindRMSweight_d (double ∗ *x,* double ∗ *e,* int *n*)

14.28.2.15 double TKgaussDev (long ∗ *seed*)

14.28.2.16 double TKmean_d (double ∗ *x,* int *n*)

14.28.2.17 float TKmean_f (float ∗ *x,* int *n*)

14.28.2.18 double TKranDev (long * *seed*)

14.28.2.19 double TKrange_d (double * *x,* int *n*)

14.28.2.20 float TKrange_f (float * *x,* int *n*)

14.28.2.21 double TKretMax_d (double *a,* double *b*)

14.28.2.22 float TKretMax_f (float *a,* float *b*)

14.28.2.23 double TKretMin_d (double *a,* double *b*)

14.28.2.24 float TKretMin_f (float *a,* float *b*)

14.28.2.25 int TKretMin_i (int *a,* int *b*)

14.28.2.26 long TKsetSeed ( )

14.28.2.27 double TKsign_d (double *a,* double *b*)

14.28.2.28 void TKsort_2f (float * *val,* float * *val2,* int *nobs*)

14.28.2.29 void TKsort_3d (double * *val,* double * *val2,* double * *val3,* int *nobs*)

14.28.2.30 void TKsort_d (double * *val,* int *nobs*)

14.28.2.31 void TKsort_f (float * *val,* int *nobs*)

14.28.2.32 double TKvariance_d (double * *x,* int *n*)

14.28.2.33 void TKzeromean_d (int *n,* double * *y*)

14.29 tabulatedfunction.h File Reference

`#include "dynarr.h"`

Include dependency graph for tabulatedfunction.h:

Classes

- struct TabulatedFunctionSample
- struct TabulatedFunction

Functions

- void TabulatedFunction_load (TabulatedFunction *func, char *fileName)
- double TabulatedFunction_getValue (TabulatedFunction *func, double x)
- double TabulatedFunction_getStartX (TabulatedFunction *func)
- double TabulatedFunction_getEndX (TabulatedFunction *func)

14.29.1 Function Documentation

14.29.1.1 double TabulatedFunction_getEndX (TabulatedFunction * *func*)

14.29.1.2 double TabulatedFunction_getStartX (**TabulatedFunction** ∗ *func*)

14.29.1.3 double TabulatedFunction_getValue (**TabulatedFunction** ∗ *func,* double *x*)

14.29.1.4 void TabulatedFunction_load (**TabulatedFunction** ∗ *func,* char ∗ *fileName*)

14.30 tempo2.h File Reference

contains the main interface to libtempo2.

```
#include <stdio.h>
#include <time.h>
#include "TKlongdouble.h"
#include "TKlog.h"
```

Include dependency graph for tempo2.h: This graph shows which files directly or indirectly include this file:

Classes

- struct FitInfo

  contains details of the fit
- struct storePrecision
- struct parameter

  Holds the values for a parameter.
- struct clock_correction
- struct observation

  A struct containing the details of a single obesrvation.
- struct pulsar

  contains the details for a single pulsar.
- struct observatory

Macros

- #define TEMPO2_h_HASH "$Id: e7d4e3eed480ba2cff605f4ca7e53d2d6368dede $"
- #define TEMPO2_h_VER "2015.09.0"
- #define TEMPO2_h_MAJOR_VER 2015.09
- #define TEMPO2_h_MINOR_VER 0
- #define TSUN longdouble(4.925490947e-6)
- #define MAX_FREQ_DERIVATIVES 13
- #define MAX_DM_DERIVATIVES 10
- #define MAX_PSR_VAL 40
- #define MAX_COMPANIONS 4
- #define NE_SW_DEFAULT 4
- #define ECLIPTIC_OBLIQUITY_VAL 84381.4059
- #define MAX_COEFF 5000
- #define MAX_CLKCORR 5000
- #define MAX_LEAPSEC 100
- #define MAX_STRLEN 1000
- #define MAX_FILELEN 500
- #define MAX_STOREPRECISION 50
- #define MAX_OBSN_VAL 20000
- #define MAX_SITE 100
- #define MAX_PARAMS 2000
- #define MAX_JUMPS 2000

- #define MAX_WHITE 100
- #define MAX_IFUNC 1000
- #define MAX_TEL_CLK_OFFS 500
- #define MAX_TEL_DX 500
- #define MAX_TEL_DY 500
- #define MAX_TEL_DZ 500
- #define MAX_FIT 10000
- #define MAX_T2EFAC 100
- #define MAX_T2EQUAD 100
- #define MAX_TNEF 50
- #define MAX_TNEQ 50
- #define MAX_TNGN 50
- #define MAX_TNBN 50 /∗maximum number of TNBandNoise parameters allowd∗/
- #define MAX_TNECORR 50
- #define MAX_TNDMEv 10 /∗Maximum number of TNDMEvents allowed ∗/
- #define MAX_TNSQ 50
- #define MAX_BPJ_JUMPS 5
- #define MAX_TOFFSET 10
- #define MAX_QUAD 150
- #define MAX_DMX 512
- #define MAX_FLAGS 40
- #define MAX_FLAG_LEN 32
- #define MAX_CLK_CORR 30
- #define SECDAY 86400.0
- #define SECDAYl longdouble(86400.0)
- #define SPEED_LIGHT 299792458.0
- #define SOLAR_MASS 1.98892e30
- #define SOLAR_RADIUS 6.96e8
- #define BIG_G 6.673e-11
- #define GM 1.3271243999e20
- #define GM_C3 4.925490947e-6
- #define GMJ_C3 4.70255e-9
- #define GMS_C3 1.40797e-9
- #define GMV_C3 1.2061e-11
- #define GMU_C3 2.14539e-10
- #define GMN_C3 2.54488e-10
- #define OBLQ 23.4458333333333333
- #define AULTSC 499.00478364
- #define AU_DIST 1.49598e11
- #define DM_CONST 2.41e-4
- #define DM_CONST_SI 7.436e6
- #define PCM 3.08568025e16
- #define MASYR2RADS 1.53628185e-16
- #define MAX_MSG 50
- #define LEAPSECOND_FILE "/clock/leap.sec"
- #define UT1_FILE "/clock/ut1.dat"
- #define TDBTDT_FILE "/ephemeris/TDB.1950.2050"
- #define IFTEPH_FILE "/ephemeris/TIMEEPH_short.te405"
- #define OBSSYS_FILE "/observatory/newobsys.dat"
- #define SI_UNITS 1
- #define TDB_UNITS 2
- #define IF99_TIMEEPH 1
- #define FB90_TIMEEPH 2
- #define T2C_IAU2000B 1
- #define T2C_TEMPO 2
- #define HAVE_GWSIM_H

Typedefs

- typedef int param_label
- typedef int constraint_label
- typedef double(∗ paramDerivFunc) (struct pulsar ∗, int, double, int, param_label, int)

  a function used to get the derivative of a parameter w.r.t. data.
- typedef double(∗ constraintDerivFunc) (struct pulsar ∗, int, constraint_label, param_label, int, int)

  a function used to get the derivative of a parameter w.r.t. constraint.
- typedef void(∗ paramUpdateFunc) (struct pulsar ∗, int, param_label, int, double, double)

  a function used to update the parameters after a fit.
- typedef struct FitInfo FitInfo

  contains details of the fit
- typedef struct storePrecision storePrecision
- typedef struct parameter parameter

  Holds the values for a parameter.
- typedef struct observation observation

  A struct containing the details of a single obesrvation.
- typedef struct pulsar pulsar

  contains the details for a single pulsar.

Enumerations

- enum label {
  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_fb, param_t0, param_a1, param_om,
  param_pmrv, param_ecc, param_edot, param_e2dot,
  param_xpbdot, param_pbdot, param_a1dot, param_a2dot,
  param_omdot, param_om2dot, param_orbpx, param_tasc,
  param_eps1, param_eps2, param_m2, param_gamma,
  param_mtot, param_glep, param_glph, param_glf0,
  param_glf1, param_glf2, param_glf0d, param_gltd,
  param_start, param_finish, param_track, param_bp,
  param_bpp, param_tzrmjd, param_tzrfrq, param_fddc,
  param_fddi, param_fd, 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, param_wave_dm,
  param_waveepoch_dm, param_dshk, param_ephver, param_daop,
  param_iperharm, param_dmassplanet, param_dphaseplanet, param_waveepoch,
  param_ifunc, param_clk_offs, param_dmx, param_dmxr1,
  param_dmxr2, param_dmmodel, param_gwsingle, param_cgw,
  param_quad_om, param_h3, param_h4, param_nharm,
  param_stig, param_telx, param_tely, param_telz,
  param_telEpoch, param_quad_ifunc_p, param_quad_ifunc_c, param_tel_dx,
  param_tel_dy, param_tel_dz, param_tel_vx, param_tel_vy,
  param_tel_vz, param_tel_x0, param_tel_y0, param_tel_z0,
  param_gwm_amp, param_gwecc, param_gwb_amp, param_dm_sin1yr,
  param_dm_cos1yr, param_brake, param_stateSwitchT, param_df1,
  param_LAST, param_ZERO, param_JUMP }

  enumeration for the various parameters that appear in a .par file

- enum constraint {
constraint_dmmodel_mean, constraint_dmmodel_dm1, constraint_dmmodel_cw_0, constraint_dmmodel_cw_1,
constraint_dmmodel_cw_2, constraint_dmmodel_cw_3, constraint_ifunc_0, constraint_ifunc_1,
constraint_ifunc_2, constraint_tel_dx_0, constraint_tel_dx_1, constraint_tel_dx_2,
constraint_tel_dy_0, constraint_tel_dy_1, constraint_tel_dy_2, constraint_tel_dz_0,
constraint_tel_dz_1, constraint_tel_dz_2, constraint_quad_ifunc_p_0, constraint_quad_ifunc_p_1,
constraint_quad_ifunc_p_2, constraint_quad_ifunc_c_0, constraint_quad_ifunc_c_1, constraint_quad_ifunc_c_2,
constraint_dmmodel_cw_year_sin, constraint_dmmodel_cw_year_cos, constraint_dmmodel_cw_year_xsin, constraint_dmmodel_cw_year_xcos,
constraint_dmmodel_cw_year_sin2, constraint_dmmodel_cw_year_cos2, constraint_dmmodel_cw_px, constraint_ifunc_year_sin,
constraint_ifunc_year_cos, constraint_ifunc_year_xsin, constraint_ifunc_year_xcos, constraint_ifunc_year_sin2,
constraint_ifunc_year_cos2, constraint_qifunc_p_year_sin, constraint_qifunc_p_year_cos, constraint_qifunc_p_year_xsin,
constraint_qifunc_p_year_xcos, constraint_qifunc_p_year_sin2, constraint_qifunc_p_year_cos2, constraint_qifunc_c_year_sin,
constraint_qifunc_c_year_cos, constraint_qifunc_c_year_xsin, constraint_qifunc_c_year_xcos, constraint_qifunc_c_year_sin2,
constraint_qifunc_c_year_cos2, constraint_LAST }

  These represent the possible constraints to the fit that have been implemented.

Functions

- int id_residual (float xcurs, float ycurs)
- float setStart (float xcurs, float ycurs, int flag)
- int zoom_graphics (float xcurs2, float ycurs2, int flag)
- void getInputs (pulsar ∗psr, int argc, char ∗argv[ ], char timFile[ ][MAX_FILELEN], char parFile[ ][MAX_FILELEN], int ∗displayParams, int ∗npsr, int ∗nGlobal, int ∗outRes, int ∗writeModel, char ∗outputSO, int ∗polyco, char ∗polyco_args, char ∗polyco_file, int ∗newpar, int ∗onlypre, char ∗dcmFile, char ∗covarFuncFile, char ∗newparname)
- void polyco (pulsar ∗psr, int npsr, longdouble polyco_MJD1, longdouble polyco_MJD2, int nspan, int ncoeff, longdouble maxha, char ∗sitename, longdouble freq, longdouble coeff[MAX_COEFF], int trueDM, char ∗polyco_file)
- void readParfile (pulsar ∗psr, char parFile[ ][MAX_FILELEN], char timFile[ ][MAX_FILELEN], int npsr)
- void readParfileGlobal (pulsar ∗psr, int npsr, char tpar[MAX_STRLEN][MAX_FILELEN], char ttim[MAX_STRLEN][MAX_FILELEN])
- int readSimpleParfile (FILE ∗fin, pulsar ∗p)
- int setupParameterFileDefaults (pulsar ∗p)
- void displayParameters (int pos, char timeFile[ ][MAX_FILELEN], char parFile[ ][MAX_FILELEN], pulsar ∗psr, int npsr)
- void initialise (pulsar ∗psr, int noWarnings)
- void initialiseOne (pulsar ∗psr, int noWarnings, int fullSetup)
- void destroyOne (pulsar ∗psr)
- void recordPrecision (pulsar ∗psr, longdouble prec, const char ∗routine, const char ∗comment)
- void readTimfile (pulsar ∗psr, char timFile[ ][MAX_FILELEN], int npsr)
- void formBats (pulsar ∗psr, int npsr)
- void formBatsAll (pulsar ∗psr, int npsr)
- void updateBatsAll (pulsar ∗psr, int npsr)
- void formResiduals (pulsar ∗psr, int npsr, int removeMean)
- int bootstrap (pulsar ∗psr, int p, int npsr)
- void doFitAll (pulsar ∗psr, int npsr, const char ∗covarFuncFile) DEPRECATED
- void doFit (pulsar ∗psr, int npsr, int writeModel) DEPRECATED

- void doFitDCM (pulsar ∗psr, const char ∗dcmFile, const char ∗covarFuncFile, int npsr, int writeModel) DEPRECATED
- void doFitGlobal (pulsar ∗psr, int npsr, double ∗globalParameter, int nGlobal, int writeModel) DEPRECATED
- void getCholeskyMatrix (double ∗∗uinv, const char ∗fname, pulsar ∗psr, double ∗resx, double ∗resy, double ∗rese, int np, int nc, int ∗ip)
- double getParamDeriv (pulsar ∗psr, int ipos, double x, int i, int k) DEPRECATED
- void textOutput (pulsar ∗psr, int npsr, double globalParameter, int nGlobal, int outRes, int newpar, const char ∗fname)
- void shapiro_delay (pulsar ∗psr, int npsr, int p, int i, double delt, double dt_SSB)
- void dm_delays (pulsar ∗psr, int npsr, int p, int i, double delt, double dt_SSB)
- void calculate_bclt (pulsar ∗psr, int npsr)
- void secularMotion (pulsar ∗psr, int npsr)
- void autoConstraints (pulsar ∗psr, int ipsr, int npsr)
- void setPlugPath ()
- void sortToAs (pulsar ∗psr)
- void preProcess (pulsar ∗psr, int npsr, int argc, char ∗argv[ ])
- void preProcessSimple (pulsar ∗psr)
- void preProcessSimple1 (pulsar ∗psr, int tempo1, double thelast)
- void preProcessSimple2 (pulsar ∗psr, float startdmmjd, int ndm, float ∗dmvals, int trimonly)
- void preProcessSimple3 (pulsar ∗psr)
- void useSelectFile (char ∗fname, pulsar ∗psr, int npsr)
- void processSimultaneous (char ∗line, pulsar ∗psr, int npsr)
- void processFlag (char ∗line, pulsar ∗psr, int npsr)
- void logicFlag (char ∗line, pulsar ∗psr, int npsr)
- void toa2utc (pulsar ∗psr, int npsr)
- void utc2tai (pulsar ∗psr, int npsr)
- void tt2tb (pulsar ∗psr, int npsr)
- void tt2tb_calceph (pulsar ∗psr, int npsr)
- void tai2tt (pulsar ∗psr, int npsr)
- void tai2ut1 (pulsar ∗psr, int npsr)
- void vectorPulsar (pulsar ∗psr, int npsr)
- void readEphemeris (pulsar ∗psr, int npsr, int addEphemNoise)
- void readOneEphemeris (pulsar ∗psr, int npsr, int addEphemNoise, int obsNumber)
- void readEphemeris_calceph (pulsar ∗psr, int npsr)
- void get_obsCoord (pulsar ∗psr, int npsr)
- void get_OneobsCoord (pulsar ∗psr, int npsr, int obs)
- double calcRMS (pulsar ∗psr, int p)
- void allocateMemory (pulsar ∗psr, int realloc)
- void destroyMemory (pulsar ∗psr)
- void readJBO_bat (char ∗fname, pulsar ∗psr, int p)
- void readObsFile (double alat[MAX_SITE], double along[MAX_SITE], double elev[MAX_SITE], int icoord[MAX_SITE], char obsnam[MAX_SITE][100], char obscode[MAX_SITE][100], int ∗nobservatory, int obsnum[MAX_SITE])
- double dotproduct (double ∗v1, double ∗v2)
- void vectorsum (double ∗res, double ∗v1, double ∗v2)
- void vectorscale (double ∗v, double k)
- void writeTim (const char ∗timname, pulsar ∗psr, const char ∗fileFormat)
- int turn_hms (double turn, char ∗hms)
- int turn_dms (double turn, char ∗dms)
- double dms_turn (char ∗line)
- double hms_turn (char ∗line)
- double turn_deg (double turn)
- longdouble fortran_mod (longdouble a, longdouble p)
- int fortran_nint (double x)
- long fortran_nlong (longdouble x)

- void equ2ecl (double ∗x)
- void copyParam (parameter p1, parameter ∗p2)
- void copyPSR (pulsar ∗p, int p1, int p2)
- longdouble getParameterValue (pulsar ∗psr, int param, int arr)
- void simplePlot (pulsar ∗psr, double unitFlag)
- double solarWindModel (pulsar psr, int iobs)
- double MSSmodel (pulsar ∗psr, int p, int obs, int param)
- void updateMSS (pulsar ∗psr, double val, double err, int pos)
- double BTmodel (pulsar ∗psr, int p, int obs, int param)
- void updateBT (pulsar ∗psr, double val, double err, int pos)
- double BTJmodel (pulsar ∗psr, int p, int obs, int param, int arr)
- void updateBTJ (pulsar ∗psr, double val, double err, int pos, int arr)
- double BTXmodel (pulsar ∗psr, int p, int obs, int param, int arr)
- void updateBTX (pulsar ∗psr, double val, double err, int pos, int arr)
- double ELL1model (pulsar ∗psr, int p, int obs, int param)
- void updateELL1 (pulsar ∗psr, double val, double err, int pos)
- longdouble DDmodel (pulsar ∗psr, int p, int obs, int param)
- void updateDD (pulsar ∗psr, double val, double err, int pos)
- double T2model (pulsar ∗psr, int p, int obs, int param, int arr)
- void updateT2 (pulsar ∗psr, double val, double err, int pos, int arr)
- double T2_PTAmodel (pulsar ∗psr, int p, int obs, int param, int arr)
- void updateT2_PTA (pulsar ∗psr, double val, double err, int pos, int arr)
- double JVmodel (pulsar ∗psr, int p, int obs, int param, int arr)
- void updateJV (pulsar ∗psr, double val, double err, int pos, int arr)
- double DDKmodel (pulsar ∗psr, int p, int obs, int param)
- void updateDDK (pulsar ∗psr, double val, double err, int pos)
- double DDSmodel (pulsar ∗psr, int p, int obs, int param)
- void updateDDS (pulsar ∗psr, double val, double err, int pos)
- double DDGRmodel (pulsar ∗psr, int p, int obs, int param)
- void updateDDGR (pulsar ∗psr, double val, double err, int pos)
- double DDHmodel (pulsar ∗psr, int p, int obs, int param)
- void updateDDH (pulsar ∗psr, double val, double err, int pos)
- double ELL1Hmodel (pulsar ∗psr, int p, int obs, int param)
- void updateELL1H (pulsar ∗psr, double val, double err, int pos)
- void displayMsg (int type, const char ∗key, const char ∗searchStr, const char ∗variableStr, int noWarnings)
- void CVSdisplayVersion (const char ∗file, const char ∗func, const char ∗verNum)
- void transform_units (struct pulsar ∗psr, int from, int to)
- void FITfuncs (double x, double afunc[ ], int ma, pulsar ∗psr, int ipos, int ipsr)
- void updateParameters (pulsar ∗psr, int p, double ∗val, double ∗error)
- void defineClockCorrectionSequence (char ∗fileList, int dispWarnings)
- void getClockCorrections (observation ∗obs, const char ∗clockFrom, const char ∗clockTo, int warnings)
- double getCorrectionTT (observation ∗obs)
- double getCorrection (observation ∗obs, const char ∗clockFrom, const char ∗clockTo, int warnings)
- observatory ∗ getObservatory (char ∗code)
- void lookup_observatory_alias (char ∗incode, char ∗outcode)
- void get_obsCoord_IAU2000B (double observatory_trs[3], double zenith_trs[3], longdouble tt_mjd, longdouble utc_mjd, double observatory_crs[3], double zenith_crs[3], double observatory_velocity_crs[3])
- void get_EOP (double mjd, double ∗xp, double ∗yp, double ∗dut1, double ∗dut1dot, int dispWarnings, char ∗eopcFile)
- void compute_tropospheric_delays (pulsar ∗psr, int npsr)

Variables

- char TEMPO2_ENVIRON [ ]
- char TEMPO2_ERROR [ ]
- char NEWFIT
- int MAX_PSR
- int MAX_OBSN
- double ECLIPTIC_OBLIQUITY
- int forceGlobalFit
- int veryFast
- char tempo2MachineType [MAX_FILELEN]
- int displayCVSversion
- char dcmFile [MAX_FILELEN]
- char covarFuncFile [MAX_FILELEN]
- char tempo2_plug_path [32][MAX_STRLEN]
- int tempo2_plug_path_len

14.30.1 Detailed Description

contains the main interface to libtempo2.

Note

some parts of this to be moved to an internal interface

14.30.2 Macro Definition Documentation

14.30.2.1 #define AU_DIST 1.49598e11

1 AU in m

14.30.2.2 #define AULTSC 499.00478364

Number of light seconds in 1 AU

14.30.2.3 #define BIG_G 6.673e-11

Gravitational constant

14.30.2.4 #define DM_CONST 2.41e-4

14.30.2.5 #define DM_CONST_SI 7.436e6

Dispersion constant in SI units

14.30.2.6 #define ECLIPTIC_OBLIQUITY_VAL 84381.4059

mean obliquity of ecliptic in arcsec

14.30.2.7 #define FB90_TIMEEPH 2

Fairhead & Bretagnon time ephemeris

14.30.2.8 #define GM 1.3271243999e20

Gravitational constant $*$ mass sun

14.30.2.9 #define GM_C3 4.925490947e-6

GM_odot/c^3 (in seconds)

14.30.2.10 #define GMJ_C3 4.70255e-9

GM_jupiter/c^3 (in seconds)

14.30.2.11 #define GMN_C3 2.54488e-10

GM_neptune/c^3 (in seconds)

14.30.2.12 #define GMS_C3 1.40797e-9

GM_saturn/c^3 (in seconds)

14.30.2.13 #define GMU_C3 2.14539e-10

GM_uranus/c^3 (in seconds)

14.30.2.14 #define GMV_C3 1.2061e-11

GM_venus/c^3 (in seconds)

14.30.2.15 #define HAVE_GWSIM_H

14.30.2.16 #define IF99_TIMEEPH 1

Irwin & Fukushima time ephemeris

14.30.2.17 #define IFTEPH_FILE "/ephemeris/TIMEEPH_short.te405"

14.30.2.18 #define LEAPSECOND_FILE "/clock/leap.sec"

Path for the file containing dates when leap seconds should be added

14.30.2.19 #define MASYR2RADS 1.53628185e-16

Converts from mas/yr to rad/s

14.30.2.20 #define MAX_BPJ_JUMPS 5

Maximum number of jumps in binary params - for BPJ model

14.30.2.21 #define MAX_CLK_CORR 30

Maximum number of steps in the correction to TT

14.30.2.22 #define MAX_CLKCORR 5000

Maximum number of lines in time.dat file

14.30.2.23 #define MAX_COEFF 5000

Maximum number of coefficients in polyco

14.30.2.24 #define MAX_COMPANIONS 4

Maximum number of binary companions

14.30.2.25 #define MAX_DM_DERIVATIVES 10

DM0 -> DMn where n=10

14.30.2.26 #define MAX_DMX 512

Max number of DM steps allowed

14.30.2.27 #define MAX_FILELEN 500

Maximum filename length

14.30.2.28 #define MAX_FIT 10000

Maximum number of parameters to fit for

14.30.2.29 #define MAX_FLAG_LEN 32

Maximum number of characters in each flag

14.30.2.30 #define MAX_FLAGS 40

Maximum number of flags in .tim file/observation

14.30.2.31 #define MAX_FREQ_DERIVATIVES 13

F0 -> Fn where n=10

14.30.2.32 #define MAX_IFUNC 1000

Maximum number of parameters for interpolation function

14.30.2.33 #define MAX_JUMPS 2000

Maximum number of phase jumps

14.30.2.34 #define MAX_LEAPSEC 100

Maximum number of line in the leap second file

14.30.2.35 #define MAX_MSG 50

Maximum number of different warnings

14.30.2.36 #define MAX_OBSN_VAL 20000

Maximum number of TOAs

14.30.2.37 #define MAX_PARAMS 2000

Maximum number of parameters

14.30.2.38 #define MAX_PSR_VAL 40

Maximum number of pulsars

14.30.2.39 #define MAX_QUAD 150

Maximum number of frequency channels in quadrupolar function

14.30.2.40 #define MAX_SITE 100

Maximum number of observatory sites

14.30.2.41 #define MAX_STOREPRECISION 50

How many routines in TEMPO2 store precision information

14.30.2.42 #define MAX_STRLEN 1000

Maximum length for strings

14.30.2.43 #define MAX_T2EFAC 100

Maximum number of T2EFACs allowed

14.30.2.44 #define MAX_T2EQUAD 100

Maximum number of T2EQUADs allowed

14.30.2.45 #define MAX_TEL_CLK_OFFS 500

Maximum number of parameters for telescope clock offset

14.30.2.46 #define MAX_TEL_DX 500

Maximum number of parameters for interpolation function

14.30.2.47 #define MAX_TEL_DY 500

Maximum number of parameters for interpolation function

14.30.2.48 #define MAX_TEL_DZ 500

Maximum number of parameters for interpolation function

14.30.2.49 #define MAX_TNBN 50 /∗maximum number of TNBandNoise parameters allowd∗/

14.30.2.50 #define MAX_TNDMEv 10 /∗Maximum number of TNDMEvents allowed ∗/

14.30.2.51 #define MAX_TNECORR 50

Maximum number of TNECORRss allowed

14.30.2.52 #define MAX_TNEF 50

Maximum number of TNEFACs allowed

14.30.2.53 #define MAX_TNEQ 50

Maximum number of TNEQUADs allowed

14.30.2.54 #define MAX_TNGN 50

maximum number of TNGroupNoise parameters allowed

14.30.2.55 #define MAX_TNSQ 50

Maximum number of TNEQUADs allowed

14.30.2.56 #define MAX_TOFFSET 10

Number of time jumps allowed in .par file

14.30.2.57 #define MAX_WHITE 100

Maximum number of parameters for whitening

14.30.2.58 #define NE_SW_DEFAULT 4

Default value for electron density (cm-3) at 1AU due to solar wind

14.30.2.59 #define OBLQ 23.44583333333333333

Obliquity of the ecliptic

14.30.2.60 #define OBSSYS_FILE "/observatory/newobsys.dat"

Path for file containing Observatory data (obsys.dat)

14.30.2.61 #define PCM 3.08568025e16

one parsec in meters

14.30.2.62 #define SECDAY 86400.0

Number of seconds in 1 day

14.30.2.63 #define SECDAYl longdouble(86400.0)

Number of seconds in 1 day

14.30.2.64 #define SI_UNITS 1

New tempo2 mode

14.30.2.65 #define SOLAR_MASS 1.98892e30

Mass of Sun (kg)

14.30.2.66 #define SOLAR_RADIUS 6.96e8

Radius of the Sun (in meters)

14.30.2.67 #define SPEED_LIGHT 299792458.0

Speed of light (m/s)

14.30.2.68 #define T2C_IAU2000B 1

14.30.2.69 #define T2C_TEMPO 2

14.30.2.70 #define TDB_UNITS 2

original tempo mode

14.30.2.71 #define TDBTDT_FILE "/ephemeris/TDB.1950.2050"

Path for file containing TDB-TDT ephemeris

14.30.2.72 #define TEMPO2_h_HASH "$Id: e7d4e3eed480ba2cff605f4ca7e53d2d6368dede $"

14.30.2.73 #define TEMPO2_h_MAJOR_VER 2015.09

14.30.2.74 #define TEMPO2_h_MINOR_VER 0

14.30.2.75 #define TEMPO2_h_VER "2015.09.0"

14.30.2.76 #define TSUN longdouble(4.925490947e-6)

Solar constant for mass calculations.

14.30.2.77 #define UT1_FILE "/clock/ut1.dat"

Path for the file containing TAI-UT1

14.30.3 Typedef Documentation

14.30.3.1 typedef int constraint_label

for 'strong typing' - type for enum constraint

14.30.3.2 typedef double(* constraintDerivFunc) (struct pulsar *, int, constraint_label, param_label, int, int)

a function used to get the derivative of a parameter w.r.t. constraint.

Used to build the derivative matrix for the least squares solvers.

14.30.3.3 typedef struct FitInfo FitInfo

contains details of the fit

Holds references to the fit functions, as well as references linking the index in the derivative matrix to the actual parameter fit for.

14.30.3.4 typedef struct observation observation

A struct containing the details of a single obesrvation.

14.30.3.5 typedef int param_label

for 'strong typing' - type for enum label

14.30.3.6 typedef double(* paramDerivFunc) (struct pulsar *, int, double, int, param_label, int)

a function used to get the derivative of a parameter w.r.t. data.

Used to build the derivative matrix for the least squares solvers.

14.30.3.7 typedef struct parameter parameter

Holds the values for a parameter.

May include multiple values, for e.g. F0, F1, F2,...

Note

If this structure is modified - must update copyParam in tempo2Util.C

14.30.3.8 typedef void(∗ paramUpdateFunc) (struct pulsar ∗, int, param_label, int, double, double)

a function used to update the parameters after a fit.

14.30.3.9 typedef struct pulsar pulsar

contains the details for a single pulsar.

Includes an array of observations and parameters

14.30.3.10 typedef struct storePrecision storePrecision

14.30.4 Enumeration Type Documentation

14.30.4.1 enum constraint

These represent the possible constraints to the fit that have been implemented.

Enumerator

constraint_dmmodel_mean

constraint_dmmodel_dm1

constraint_dmmodel_cw_0

constraint_dmmodel_cw_1

constraint_dmmodel_cw_2

constraint_dmmodel_cw_3

constraint_ifunc_0

constraint_ifunc_1

constraint_ifunc_2

constraint_tel_dx_0

constraint_tel_dx_1

constraint_tel_dx_2

constraint_tel_dy_0

constraint_tel_dy_1

constraint_tel_dy_2

constraint_tel_dz_0

constraint_tel_dz_1

constraint_tel_dz_2

constraint_quad_ifunc_p_0

constraint_quad_ifunc_p_1

constraint_quad_ifunc_p_2

constraint_quad_ifunc_c_0

constraint_quad_ifunc_c_1

constraint_quad_ifunc_c_2

constraint_dmmodel_cw_year_sin

constraint_dmmodel_cw_year_cos

constraint_dmmodel_cw_year_xsin

constraint_dmmodel_cw_year_xcos

constraint_dmmodel_cw_year_sin2

constraint_dmmodel_cw_year_cos2

constraint_dmmodel_cw_px

constraint_ifunc_year_sin

constraint_ifunc_year_cos

constraint_ifunc_year_xsin

constraint_ifunc_year_xcos

constraint_ifunc_year_sin2

constraint_ifunc_year_cos2

constraint_qifunc_p_year_sin

constraint_qifunc_p_year_cos

constraint_qifunc_p_year_xsin

constraint_qifunc_p_year_xcos

constraint_qifunc_p_year_sin2

constraint_qifunc_p_year_cos2

constraint_qifunc_c_year_sin

constraint_qifunc_c_year_cos

constraint_qifunc_c_year_xsin

constraint_qifunc_c_year_xcos

constraint_qifunc_c_year_sin2

constraint_qifunc_c_year_cos2

constraint_LAST marker for the last constraint

14.30.4.2 enum label

enumeration for the various parameters that appear in a .par file

The last parameter is param_LAST, but there are enumerations after this for spectial fits. It is important not to change the order of the elements

Note

when adding a new parameter, initialise it in intialise.c after param_LAST.

Enumerator

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_fb

param_t0

param_a1

param_om

param_pmrv

param_ecc

param_edot

param_e2dot

param_xpbdot

param_pbdot

param_a1dot

param_a2dot

param_omdot

param_om2dot

param_orbpx

param_tasc

param_eps1

param_eps2

param_m2

param_gamma

param_mtot

param_glep

param_glph

param_glf0

param_glf1

param_glf2

param_glf0d

param_gltd

param_start

param_finish

param_track

param_bp

param_bpp

param_tzrmjd

param_tzrfrq

param_fddc

param_fddi

param_fd

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

param_wave_dm

param_waveepoch_dm

param_dshk

param_ephver

param_daop

param_iperharm

param_dmassplanet

param_dphaseplanet

param_waveepoch

param_ifunc

param_clk_offs

param_dmx

param_dmxr1

param_dmxr2

param_dmmodel

param_gwsingle

param_cgw

param_quad_om

param_h3

param_h4

param_nharm

param_stig

param_telx

param_tely

param_telz

param_telEpoch

param_quad_ifunc_p

param_quad_ifunc_c

param_tel_dx

param_tel_dy

param_tel_dz

param_tel_vx

param_tel_vy

param_tel_vz

param_tel_x0

param_tel_y0

param_tel_z0

param_gwm_amp

param_gwecc

param_gwb_amp

param_dm_sin1yr

param_dm_cos1yr

param_brake

param_stateSwitchT

param_df1

param_LAST Marker for the last param to be used in for loops

param_ZERO virtual parameter for DC offset

param_JUMP virtual parameter for jumps

14.30.5 Function Documentation

14.30.5.1 void allocateMemory (pulsar ∗ *psr,* int *realloc*)

14.30.5.2 void autoConstraints (pulsar ∗ *psr,* int *ipsr,* int *npsr*)

14.30.5.3 int bootstrap (pulsar ∗ *psr,* int *p,* int *npsr*)

14.30.5.4 double BTJmodel (pulsar ∗ *psr,* int *p,* int *obs,* int *param,* int *arr*)

14.30.5.5 double BTmodel (pulsar ∗ *psr,* int *p,* int *obs,* int *param*)

14.30.5.6 double BTXmodel (pulsar ∗ *psr,* int *p,* int *obs,* int *param,* int *arr*)

14.30.5.7 double calcRMS (pulsar ∗ *psr,* int *p*)

14.30.5.8 void calculate_bclt (pulsar ∗ *psr,* int *npsr*)

14.30.5.9 void compute_tropospheric_delays (pulsar ∗ *psr,* int *npsr*)

14.30.5.10 void copyParam (parameter *p1,* parameter ∗ *p2*)

14.30.5.11 void copyPSR (pulsar ∗ *p,* int *p1,* int *p2*)

14.30.5.12 void CVSdisplayVersion (const char ∗ *file,* const char ∗ *func,* const char ∗ *verNum*)

14.30.5.13 double DDGRmodel (pulsar ∗ *psr,* int *p,* int *obs,* int *param*)

14.30.5.14 double DDHmodel (pulsar ∗ *psr,* int *p,* int *obs,* int *param*)

14.30.5.15 double DDKmodel (pulsar ∗ *psr,* int *p,* int *obs,* int *param*)

14.30.5.16 longdouble DDmodel (pulsar ∗ *psr,* int *p,* int *obs,* int *param*)

14.30.5.17 double DDSmodel (pulsar ∗ *psr,* int *p,* int *obs,* int *param*)

14.30.5.18 void defineClockCorrectionSequence (char ∗ *fileList,* int *dispWarnings*)

14.30.5.19 void destroyMemory (pulsar ∗ *psr*)

14.30.5.20 void destroyOne (pulsar ∗ *psr*)

14.30.5.21 void displayMsg (int *type,* const char ∗ *key,* const char ∗ *searchStr,* const char ∗ *variableStr,* int *noWarnings*)

14.30.5.22 void displayParameters (int *pos,* char *timeFile[ ][MAX_FILELEN],* char *parFile[ ][MAX_FILELEN],* pulsar ∗ *psr,* int *npsr*)

14.30.5.23 void dm_delays (pulsar ∗ *psr,* int *npsr,* int *p,* int *i,* double *delt,* double *dt_SSB*)

14.30.5.24 double dms_turn (char ∗ *line*)

14.30.5.25 void doFit (pulsar ∗ *psr,* int *npsr,* int *writeModel*)

14.30.5.26 void doFitAll (pulsar ∗ *psr,* int *npsr,* const char ∗ *covarFuncFile*)

14.30.5.27 void doFitDCM (pulsar ∗ *psr,* const char ∗ *dcmFile,* const char ∗ *covarFuncFile,* int *npsr,* int *writeModel*)

14.30.5.28 void doFitGlobal (pulsar ∗ *psr,* int *npsr,* double ∗ *globalParameter,* int *nGlobal,* int *writeModel*)

14.30.5.29 double dotproduct (double ∗ *v1,* double ∗ *v2*)

14.30.5.30 double ELL1Hmodel (pulsar ∗ *psr,* int *p,* int *obs,* int *param*)

14.30.5.31 double ELL1model (pulsar ∗ *psr,* int *p,* int *obs,* int *param*)

14.30.5.32 void equ2ecl (double ∗ *x*)

14.30.5.33 void FITfuncs (double *x,* double *afunc[ ],* int *ma,* pulsar ∗ *psr,* int *ipos,* int *ipsr*)

14.30.5.34 void formBats (pulsar ∗ *psr,* int *npsr*)

14.30.5.35 void formBatsAll (pulsar ∗ *psr,* int *npsr*)

14.30.5.36 void formResiduals (pulsar ∗ *psr,* int *npsr,* int *removeMean*)

14.30.5.37 longdouble fortran_mod (longdouble *a,* longdouble *p*)

14.30.5.38 int fortran_nint (double *x*)

14.30.5.39 long fortran_nlong (longdouble *x*)

14.30.5.40 void get_EOP (double *mjd*, double ∗ *xp*, double ∗ *yp*, double ∗ *dut1*, double ∗ *dut1dot*, int *dispWarnings*, char ∗ *eopcFile*)

14.30.5.41 void get_obsCoord (pulsar ∗ *psr*, int *npsr*)

14.30.5.42 void get_obsCoord_IAU2000B (double *observatory_trs[3]*, double *zenith_trs[3]*, longdouble *tt_mjd*, longdouble *utc_mjd*, double *observatory_crs[3]*, double *zenith_crs[3]*, double *observatory_velocity_crs[3]*)

14.30.5.43 void get_OneobsCoord (pulsar ∗ *psr*, int *npsr*, int *obs*)

14.30.5.44 void getCholeskyMatrix (double ∗∗ *uinv*, const char ∗ *fname*, pulsar ∗ *psr*, double ∗ *resx*, double ∗ *resy*, double ∗ *rese*, int *np*, int *nc*, int ∗ *ip*)

14.30.5.45 void getClockCorrections (observation ∗ *obs*, const char ∗ *clockFrom*, const char ∗ *clockTo*, int *warnings*)

14.30.5.46 double getCorrection (observation ∗ *obs*, const char ∗ *clockFrom*, const char ∗ *clockTo*, int *warnings*)

14.30.5.47 double getCorrectionTT (observation ∗ *obs*)

14.30.5.48 void getInputs (pulsar ∗ *psr*, int *argc*, char ∗ *argv[ ]*, char *timFile[ ][MAX_FILELEN]*, char *parFile[ ][MAX_FILELEN]*, int ∗ *displayParams*, int ∗ *npsr*, int ∗ *nGlobal*, int ∗ *outRes*, int ∗ *writeModel*, char ∗ *outputSO*, int ∗ *polyco*, char ∗ *polyco_args*, char ∗ *polyco_file*, int ∗ *newpar*, int ∗ *onlypre*, char ∗ *dcmFile*, char ∗ *covarFuncFile*, char ∗ *newparname*)

14.30.5.49 observatory∗ getObservatory (char ∗ *code*)

14.30.5.50 double getParamDeriv (pulsar ∗ *psr*, int *ipos*, double *x*, int *i*, int *k*)

14.30.5.51 longdouble getParameterValue (pulsar ∗ *psr*, int *param*, int *arr*)

14.30.5.52 double hms_turn (char ∗ *line*)

14.30.5.53 int id_residual (float *xcurs*, float *ycurs*)

14.30.5.54 void initialise (pulsar ∗ *psr*, int *noWarnings*)

14.30.5.55 void initialiseOne (pulsar ∗ *psr*, int *noWarnings*, int *fullSetup*)

14.30.5.56 double JVmodel (pulsar ∗ *psr*, int *p*, int *obs*, int *param*, int *arr*)

14.30.5.57 void logicFlag (char ∗ *line*, pulsar ∗ *psr*, int *npsr*)

14.30.5.58 void lookup_observatory_alias (char ∗ *incode*, char ∗ *outcode*)

14.30.5.59 double MSSmodel (pulsar ∗ *psr*, int *p*, int *obs*, int *param*)

14.30.5.60 void polyco (pulsar ∗ *psr*, int *npsr*, longdouble *polyco_MJD1*, longdouble *polyco_MJD2*, int *nspan*, int *ncoeff*, longdouble *maxha*, char ∗ *sitename*, longdouble *freq*, longdouble *coeff[MAX_COEFF]*, int *trueDM*, char ∗ *polyco_file*)

14.30.5.61 void preProcess (pulsar ∗ *psr*, int *npsr*, int *argc*, char ∗ *argv[ ]*)

14.30.5.62 void preProcessSimple (pulsar ∗ *psr*)

14.30.5.63 void preProcessSimple1 (pulsar ∗ *psr*, int *tempo1*, double *thelast*)

14.30.5.64 void preProcessSimple2 (pulsar ∗ *psr,* float *startdmmjd,* int *ndm,* float ∗ *dmvals,* int *trimonly*)

14.30.5.65 void preProcessSimple3 (pulsar ∗ *psr*)

14.30.5.66 void processFlag (char ∗ *line,* pulsar ∗ *psr,* int *npsr*)

14.30.5.67 void processSimultaneous (char ∗ *line,* pulsar ∗ *psr,* int *npsr*)

14.30.5.68 void readEphemeris (pulsar ∗ *psr,* int *npsr,* int *addEphemNoise*)

14.30.5.69 void readEphemeris_calceph (pulsar ∗ *psr,* int *npsr*)

14.30.5.70 void readJBO_bat (char ∗ *fname,* pulsar ∗ *psr,* int *p*)

14.30.5.71 void readObsFile (double *alat[MAX_SITE],* double *along[MAX_SITE],* double *elev[MAX_SITE],* int *icoord[MAX_SITE],* char *obsnam[MAX_SITE][100],* char *obscode[MAX_SITE][100],* int ∗ *nobservatory,* int *obsnum[MAX_SITE]*)

14.30.5.72 void readOneEphemeris (pulsar ∗ *psr,* int *npsr,* int *addEphemNoise,* int *obsNumber*)

14.30.5.73 void readParfile (pulsar ∗ *psr,* char *parFile[ ][MAX_FILELEN],* char *timFile[ ][MAX_FILELEN],* int *npsr*)

14.30.5.74 void readParfileGlobal (pulsar ∗ *psr,* int *npsr,* char *tpar[MAX_STRLEN][MAX_FILELEN],* char *ttim[MAX_STRLEN][MAX_FILELEN]*)

14.30.5.75 int readSimpleParfile (FILE ∗ *fin,* pulsar ∗ *p*)

14.30.5.76 void readTimfile (pulsar ∗ *psr,* char *timFile[ ][MAX_FILELEN],* int *npsr*)

14.30.5.77 void recordPrecision (pulsar ∗ *psr,* longdouble *prec,* const char ∗ *routine,* const char ∗ *comment*)

14.30.5.78 void secularMotion (pulsar ∗ *psr,* int *npsr*)

14.30.5.79 void setPlugPath ( )

14.30.5.80 float setStart (float *xcurs,* float *ycurs,* int *flag*)

14.30.5.81 int setupParameterFileDefaults (pulsar ∗ *p*)

14.30.5.82 void shapiro_delay (pulsar ∗ *psr,* int *npsr,* int *p,* int *i,* double *delt,* double *dt_SSB*)

14.30.5.83 void simplePlot (pulsar ∗ *psr,* double *unitFlag*)

14.30.5.84 double solarWindModel (pulsar *psr,* int *iobs*)

14.30.5.85 void sortToAs (pulsar ∗ *psr*)

14.30.5.86 double T2_PTAmodel (pulsar ∗ *psr,* int *p,* int *obs,* int *param,* int *arr*)

14.30.5.87 double T2model (pulsar ∗ *psr,* int *p,* int *obs,* int *param,* int *arr*)

14.30.5.88 void tai2tt (pulsar ∗ *psr,* int *npsr*)

14.30.5.89 void tai2ut1 (pulsar ∗ *psr,* int *npsr*)

14.30.5.90 void textOutput (pulsar ∗ *psr,* int *npsr,* double *globalParameter,* int *nGlobal,* int *outRes,* int *newpar,* const char ∗ *fname*)

14.30.5.91 void toa2utc (pulsar ∗ *psr,* int *npsr*)

14.30.5.92 void transform_units (struct pulsar ∗ *psr,* int *from,* int *to*)

14.30.5.93 void tt2tb (pulsar ∗ *psr,* int *npsr*)

14.30.5.94 void tt2tb_calceph (pulsar ∗ *psr,* int *npsr*)

14.30.5.95 double turn_deg (double *turn*)

14.30.5.96 int turn_dms (double *turn,* char ∗ *dms*)

14.30.5.97 int turn_hms (double *turn,* char ∗ *hms*)

14.30.5.98 void updateBatsAll (pulsar ∗ *psr,* int *npsr*)

14.30.5.99 void updateBT (pulsar ∗ *psr,* double *val,* double *err,* int *pos*)

14.30.5.100 void updateBTJ (pulsar ∗ *psr,* double *val,* double *err,* int *pos,* int *arr*)

14.30.5.101 void updateBTX (pulsar ∗ *psr,* double *val,* double *err,* int *pos,* int *arr*)

14.30.5.102 void updateDD (pulsar ∗ *psr,* double *val,* double *err,* int *pos*)

14.30.5.103 void updateDDGR (pulsar ∗ *psr,* double *val,* double *err,* int *pos*)

14.30.5.104 void updateDDH (pulsar ∗ *psr,* double *val,* double *err,* int *pos*)

14.30.5.105 void updateDDK (pulsar ∗ *psr,* double *val,* double *err,* int *pos*)

14.30.5.106 void updateDDS (pulsar ∗ *psr,* double *val,* double *err,* int *pos*)

14.30.5.107 void updateELL1 (pulsar ∗ *psr,* double *val,* double *err,* int *pos*)

14.30.5.108 void updateELL1H (pulsar ∗ *psr,* double *val,* double *err,* int *pos*)

14.30.5.109 void updateJV (pulsar ∗ *psr,* double *val,* double *err,* int *pos,* int *arr*)

14.30.5.110 void updateMSS (pulsar ∗ *psr,* double *val,* double *err,* int *pos*)

14.30.5.111 void updateParameters (pulsar ∗ *psr,* int *p,* double ∗ *val,* double ∗ *error*)

14.30.5.112 void updateT2 (pulsar ∗ *psr,* double *val,* double *err,* int *pos,* int *arr*)

14.30.5.113 void updateT2_PTA (pulsar ∗ *psr,* double *val,* double *err,* int *pos,* int *arr*)

14.30.5.114 void useSelectFile (char ∗ *fname,* pulsar ∗ *psr,* int *npsr*)

14.30.5.115 void utc2tai (pulsar ∗ *psr,* int *npsr*)

14.30.5.116 void vectorPulsar (pulsar ∗ *psr,* int *npsr*)

14.30.5.117 void vectorscale (double ∗ *v,* double *k*)

14.30.5.118 void vectorsum (double ∗ *res*, double ∗ *v1*, double ∗ *v2*)

14.30.5.119 void writeTim (const char ∗ *timname*, pulsar ∗ *psr*, const char ∗ *fileFormat*)

14.30.5.120 int zoom_graphics (float *xcurs2*, float *ycurs2*, int *flag*)

14.30.6 Variable Documentation

14.30.6.1 char covarFuncFile[MAX_FILELEN]

14.30.6.2 char dcmFile[MAX_FILELEN]

14.30.6.3 int displayCVSversion

Display CVS version

14.30.6.4 double ECLIPTIC_OBLIQUITY

14.30.6.5 int forceGlobalFit

Global = 1 if we are forcing a global fit

14.30.6.6 int MAX_OBSN

size of the arrays of observations inside each pulsar

14.30.6.7 int MAX_PSR

size of the array of pulsars used in tempo2

14.30.6.8 char NEWFIT

global boolean used to enable new fit.

Warning

> this will be removed in future.

14.30.6.9 char TEMPO2_ENVIRON[ ]

TEMPO2 environment variable

14.30.6.10 char TEMPO2_ERROR[ ]

TEMPO2 error messages

14.30.6.11 char tempo2_plug_path[32][MAX_STRLEN]

paths to search for plugins

14.30.6.12 int tempo2_plug_path_len

14.30.6.13 char tempo2MachineType[MAX_FILELEN]

14.30.6.14 int veryFast

Global to run the code fast

14.31 tempo2pred.h File Reference

```
#include <stdio.h>
```

Include dependency graph for tempo2pred.h: This graph shows which files directly or indirectly include this file:

Classes

- struct Cheby2D
- struct ChebyModel
- struct ChebyModelSet
- struct T1Polyco
- struct T1PolycoSet
- struct T2Predictor

Enumerations

- enum T2PredictorKind { NonePredType, Cheby, T1 }

Functions

- void T2Predictor_Init (T2Predictor ∗t2p)
- void T2Predictor_Copy (T2Predictor ∗into_t2p, const T2Predictor ∗from_t2p)
- int T2Predictor_Insert (T2Predictor ∗into_t2p, const T2Predictor ∗from_t2p)
- void T2Predictor_Keep (T2Predictor ∗, unsigned nmjd, const long double ∗mjd)
- void T2Predictor_Destroy (T2Predictor ∗t2p)
- int T2Predictor_Read (T2Predictor ∗t2p, char ∗fname)
- int T2Predictor_FRead (T2Predictor ∗t2p, FILE ∗f)
- void T2Predictor_Write (const T2Predictor ∗t2p, char ∗fname)
- void T2Predictor_FWrite (const T2Predictor ∗t2p, FILE ∗f)
- char ∗ T2Predictor_GetPSRName (T2Predictor ∗t2p)
- char ∗ T2Predictor_GetSiteName (T2Predictor ∗t2p)
- long double T2Predictor_GetStartMJD (T2Predictor ∗t2p)
- long double T2Predictor_GetEndMJD (T2Predictor ∗t2p)
- long double T2Predictor_GetStartFreq (T2Predictor ∗t2p)
- long double T2Predictor_GetEndFreq (T2Predictor ∗t2p)
- T2PredictorKind T2Predictor_Kind (T2Predictor ∗t2p)
- long double T2Predictor_GetPhase (const T2Predictor ∗t2p, long double mjd, long double freq)
- long double T2Predictor_GetFrequency (const T2Predictor ∗t2p, long double mjd, long double freq)
- int T2Predictor_GetPlan (char ∗filename, long double mjd_start, long double mjd_end, long double step, long double freq, long double ∗phase0, int ∗nsegments, long double ∗pulse_frequencies)
- int T2Predictor_GetPlan_Ext (char ∗filename, long double mjd_start, long double mjd_end, long double step, long double freq, char ∗psrname, char ∗sitename, long double ∗phase0, int ∗nsegments, long double ∗pulse_frequencies)

Variables

- int ChebyModelSet_OutOfRange

14.31.1 Enumeration Type Documentation

14.31.1.1 enum **T2PredictorKind**

Enumerator

NonePredType

Cheby

T1

14.31.2 Function Documentation

14.31.2.1 void T2Predictor_Copy (**T2Predictor** ∗ *into_t2p,* const **T2Predictor** ∗ *from_t2p*)

14.31.2.2 void T2Predictor_Destroy (**T2Predictor** ∗ *t2p*)

14.31.2.3 int T2Predictor_FRead (**T2Predictor** ∗ *t2p,* FILE ∗ *f*)

14.31.2.4 void T2Predictor_FWrite (const **T2Predictor** ∗ *t2p,* FILE ∗ *f*)

14.31.2.5 long double T2Predictor_GetEndFreq (**T2Predictor** ∗ *t2p*)

14.31.2.6 long double T2Predictor_GetEndMJD (**T2Predictor** ∗ *t2p*)

14.31.2.7 long double T2Predictor_GetFrequency (const **T2Predictor** ∗ *t2p,* long double *mjd,* long double *freq*)

14.31.2.8 long double T2Predictor_GetPhase (const **T2Predictor** ∗ *t2p,* long double *mjd,* long double *freq*)

14.31.2.9 int T2Predictor_GetPlan (char ∗ *filename,* long double *mjd_start,* long double *mjd_end,* long double *step,* long double *freq,* long double ∗ *phase0,* int ∗ *nsegments,* long double ∗ *pulse_frequencies*)

14.31.2.10 int T2Predictor_GetPlan_Ext (char ∗ *filename,* long double *mjd_start,* long double *mjd_end,* long double *step,* long double *freq,* char ∗ *psrname,* char ∗ *sitename,* long double ∗ *phase0,* int ∗ *nsegments,* long double ∗ *pulse_frequencies*)

14.31.2.11 char∗ T2Predictor_GetPSRName (**T2Predictor** ∗ *t2p*)

14.31.2.12 char∗ T2Predictor_GetSiteName (**T2Predictor** ∗ *t2p*)

14.31.2.13 long double T2Predictor_GetStartFreq (**T2Predictor** ∗ *t2p*)

14.31.2.14 long double T2Predictor_GetStartMJD (**T2Predictor** ∗ *t2p*)

14.31.2.15 void T2Predictor_Init (**T2Predictor** ∗ *t2p*)

14.31.2.16 int T2Predictor_Insert (**T2Predictor** ∗ *into_t2p,* const **T2Predictor** ∗ *from_t2p*)

14.31.2.17 void T2Predictor_Keep (**T2Predictor** ∗ , unsigned *nmjd,* const long double ∗ *mjd*)

14.31.2.18 **T2PredictorKind** T2Predictor_Kind (**T2Predictor** ∗ *t2p*)

14.31.2.19 int T2Predictor_Read (**T2Predictor** ∗ *t2p,* char ∗ *fname*)

14.31.2.20 void T2Predictor_Write (const **T2Predictor** ∗ *t2p,* char ∗ *fname*)

14.31.3 Variable Documentation

14.31.3.1 int ChebyModelSet_OutOfRange

14.32 tempo2pred_int.h File Reference

```
#include "tempo2.h"
#include "tempo2pred.h"
```

Include dependency graph for tempo2pred_int.h:

Functions

- void ChebyModel_Construct (ChebyModel ∗cm, const pulsar ∗psr)
- void ChebyModel_Test (ChebyModel ∗cm, const pulsar ∗psr, int nmjd, int nfreq, long double ∗residualRMS, long double ∗residualMAV)
- void ChebyModelSet_Construct (ChebyModelSet ∗cms, const pulsar ∗psr, const char ∗sitename, long double mjd_start, long double mjd_end, long double segment_length, long double overlap, long double freq_start, long double freq_end, int nmjdcoeff, int nfreqcoeff)
- void ChebyModelSet_Test (ChebyModelSet ∗cms, const pulsar ∗psr, int nmjd, int nfreq, long double ∗residualRMS, long double ∗residualMAV)
- void Cheby2D_Construct (Cheby2D ∗cheby, void(∗func)(long double ∗x, long double ∗y, int nx, int ny, long double ∗z, void ∗info), void ∗info)
- void Cheby2D_Construct_x_Derivative (Cheby2D ∗dcheby, const Cheby2D ∗cheby)
- void Cheby2D_Test (Cheby2D ∗cheby, int nx_test, int ny_test, void(∗func)(long double ∗x, long double ∗y, int nx, int ny, long double ∗z, void ∗info), void ∗info, long double ∗residualRMS, long double ∗residualMAV)
- void ChebyModel_Init (ChebyModel ∗cmodel, int nmjdcoeff, int nfreqcoeff)
- void ChebyModel_Copy (ChebyModel ∗cm, ChebyModel ∗from)
- void ChebyModel_Destroy (ChebyModel ∗cm)
- long double ChebyModel_GetPhase (const ChebyModel ∗cm, long double mjd, long double freq)
- long double ChebyModel_GetFrequency (const ChebyModel ∗cm, long double mjd, long double freq)
- void ChebyModel_Write (const ChebyModel ∗cm, FILE ∗f)
- int ChebyModel_Read (ChebyModel ∗cm, FILE ∗f)
- ChebyModel ∗ ChebyModelSet_GetNearest (const ChebyModelSet ∗cms, long double mjd)
- long double ChebyModelSet_GetPhase (const ChebyModelSet ∗cms, long double mjd, long double freq)
- long double ChebyModelSet_GetFrequency (const ChebyModelSet ∗cms, long double mjd, long double freq)
- void ChebyModelSet_Write (const ChebyModelSet ∗cms, FILE ∗f)
- int ChebyModelSet_Read (ChebyModelSet ∗cms, FILE ∗f)
- void ChebyModelSet_Init (ChebyModelSet ∗cms)
- int ChebyModelSet_Insert (ChebyModelSet ∗cms, const ChebyModelSet ∗from)
- void ChebyModelSet_Keep (ChebyModelSet ∗cms, unsigned nmjd, const long double ∗mjd)
- void ChebyModelSet_Destroy (ChebyModelSet ∗cms)
- long double T1Polyco_GetPhase (const T1Polyco ∗t1p, long double mjd, long double freq)
- long double T1Polyco_GetFrequency (const T1Polyco ∗t1p, long double mjd, long double freq)
- void T1Polyco_Write (const T1Polyco ∗t1p, FILE ∗f)
- int T1Polyco_Read (T1Polyco ∗t1p, FILE ∗f)
- T1Polyco ∗ T1PolycoSet_GetNearest (long double mjd)
- long double T1PolycoSet_GetPhase (const T1PolycoSet ∗t1ps, long double mjd, long double freq)
- long double T1PolycoSet_GetFrequency (const T1PolycoSet ∗t1ps, long double mjd, long double freq)
- void T1PolycoSet_Write (const T1PolycoSet ∗t1ps, FILE ∗f)
- int T1PolycoSet_Read (T1PolycoSet ∗t1ps, FILE ∗f)
- void T1PolycoSet_Destroy (T1PolycoSet ∗t1ps)

14.32.1 Function Documentation

14.32.1.1 void Cheby2D_Construct (**Cheby2D** ∗ *cheby,* void(∗)(long double ∗x, long double ∗y, int nx, int ny, long double ∗z, void ∗info) *func,* void ∗ *info*)

14.32.1.2 void Cheby2D_Construct_x_Derivative (**Cheby2D** ∗ *dcheby,* const **Cheby2D** ∗ *cheby*)

14.32.1.3 void Cheby2D_Test (**Cheby2D** ∗ *cheby,* int *nx_test,* int *ny_test,* void(∗)(long double ∗x, long double ∗y, int nx, int ny, long double ∗z, void ∗info) *func,* void ∗ *info,* long double ∗ *residualRMS,* long double ∗ *residualMAV*)

14.32.1.4 void ChebyModel_Construct (**ChebyModel** ∗ *cm,* const **pulsar** ∗ *psr*)

14.32.1.5 void ChebyModel_Copy (**ChebyModel** ∗ *cm,* **ChebyModel** ∗ *from*)

14.32.1.6 void ChebyModel_Destroy (**ChebyModel** ∗ *cm*)

14.32.1.7 long double ChebyModel_GetFrequency (const **ChebyModel** ∗ *cm,* long double *mjd,* long double *freq*)

14.32.1.8 long double ChebyModel_GetPhase (const **ChebyModel** ∗ *cm,* long double *mjd,* long double *freq*)

14.32.1.9 void ChebyModel_Init (**ChebyModel** ∗ *cmodel,* int *nmjdcoeff,* int *nfreqcoeff*)

14.32.1.10 int ChebyModel_Read (**ChebyModel** ∗ *cm,* FILE ∗ *f*)

14.32.1.11 void ChebyModel_Test (**ChebyModel** ∗ *cm,* const **pulsar** ∗ *psr,* int *nmjd,* int *nfreq,* long double ∗ *residualRMS,* long double ∗ *residualMAV*)

14.32.1.12 void ChebyModel_Write (const **ChebyModel** ∗ *cm,* FILE ∗ *f*)

14.32.1.13 void ChebyModelSet_Construct (**ChebyModelSet** ∗ *cms,* const **pulsar** ∗ *psr,* const char ∗ *sitename,* long double *mjd_start,* long double *mjd_end,* long double *segment_length,* long double *overlap,* long double *freq_start,* long double *freq_end,* int *nmjdcoeff,* int *nfreqcoeff*)

14.32.1.14 void ChebyModelSet_Destroy (**ChebyModelSet** ∗ *cms*)

14.32.1.15 long double ChebyModelSet_GetFrequency (const **ChebyModelSet** ∗ *cms,* long double *mjd,* long double *freq*)

14.32.1.16 **ChebyModel**∗ ChebyModelSet_GetNearest (const **ChebyModelSet** ∗ *cms,* long double *mjd*)

14.32.1.17 long double ChebyModelSet_GetPhase (const **ChebyModelSet** ∗ *cms,* long double *mjd,* long double *freq*)

14.32.1.18 void ChebyModelSet_Init (**ChebyModelSet** ∗ *cms*)

14.32.1.19 int ChebyModelSet_Insert (**ChebyModelSet** ∗ *cms,* const **ChebyModelSet** ∗ *from*)

14.32.1.20 void ChebyModelSet_Keep (**ChebyModelSet** ∗ *cms,* unsigned *nmjd,* const long double ∗ *mjd*)

14.32.1.21 int ChebyModelSet_Read (**ChebyModelSet** ∗ *cms,* FILE ∗ *f*)

14.32.1.22 void ChebyModelSet_Test (**ChebyModelSet** ∗ *cms,* const **pulsar** ∗ *psr,* int *nmjd,* int *nfreq,* long double ∗ *residualRMS,* long double ∗ *residualMAV*)

14.32.1.23 void ChebyModelSet_Write (const **ChebyModelSet** ∗ *cms,* FILE ∗ *f*)

14.32.1.24 long double T1Polyco_GetFrequency (const **T1Polyco** ∗ *t1p,* long double *mjd,* long double *freq*)

14.32.1.25 long double T1Polyco_GetPhase (const **T1Polyco** ∗ *t1p,* long double *mjd,* long double *freq*)

14.32.1.26 int T1Polyco_Read (**T1Polyco** ∗ *t1p,* FILE ∗ *f*)

14.32.1.27 void T1Polyco_Write (const **T1Polyco** ∗ *t1p,* FILE ∗ *f*)

14.32.1.28 void T1PolycoSet_Destroy (**T1PolycoSet** ∗ *t1ps*)

14.32.1.29 long double T1PolycoSet_GetFrequency (const **T1PolycoSet** ∗ *t1ps,* long double *mjd,* long double *freq*)

14.32.1.30 **T1Polyco**∗ T1PolycoSet_GetNearest (long double *mjd*)

14.32.1.31 long double T1PolycoSet_GetPhase (const **T1PolycoSet** ∗ *t1ps,* long double *mjd,* long double *freq*)

14.32.1.32 int T1PolycoSet_Read (**T1PolycoSet** ∗ *t1ps,* FILE ∗ *f*)

14.32.1.33 void T1PolycoSet_Write (const **T1PolycoSet** ∗ *t1ps,* FILE ∗ *f*)

14.33 tempo2Util.h File Reference

Functions

- double turn_deg (double turn)
- double dms_turn (char ∗line)
- double hms_turn (char ∗line)

14.33.1 Function Documentation

14.33.1.1 double dms_turn (char ∗ *line*)

14.33.1.2 double hms_turn (char ∗ *line*)

14.33.1.3 double turn_deg (double *turn*)

14.34 TKcholesky.h File Reference

Functions

- void cholesky_readFromCovarianceFunction (double ∗∗m, const char ∗fname, double ∗resx, double ∗resy, double ∗rese, int np, int nc)
- void cholesky_covarFunc2matrix (double ∗∗m, double ∗covarFunc, int ndays, double ∗resx, double ∗resy, double ∗rese, int np, int nc)
- void cholesky_powerlawModel (double ∗∗m, double modelAlpha, double modelFc, double modelA, double ∗resx, double ∗resy, double ∗rese, int np, int nc)
- void cholesky_powerlawModel_withBeta (double ∗∗m, double modelAlpha, double beta, double modelFc, double modelA, double ∗resx, double ∗resy, double ∗rese, int np, int nc)
- int cholesky_formUinv (double ∗∗uinv, double ∗∗m, int np)
- void cholesky_dmModel (double ∗∗m, double D, double d, double ref_freq, double ∗resx, double ∗resy, double ∗rese, int np, int nc)
- void cholesky_ecm (double ∗∗m, char ∗fileName, double ∗resx, double ∗resy, double ∗rese, int np, int nc)
- void cholesky_dmModelCovarParam (double ∗∗m, double alpha, double a, double b, double ∗resx, double ∗resy, double ∗rese, int np, int nc)

14.34.1 Function Documentation

14.34.1.1 void cholesky_covarFunc2matrix (double ∗∗ *m,* double ∗ *covarFunc,* int *ndays,* double ∗ *resx,* double ∗ *resy,* double ∗ *rese,* int *np,* int *nc*)

14.34.1.2 void cholesky_dmModel (double ∗∗ *m,* double *D,* double *d,* double *ref_freq,* double ∗ *resx,* double ∗ *resy,* double ∗ *rese,* int *np,* int *nc*)

14.34.1.3 void cholesky_dmModelCovarParam (double ∗∗ *m,* double *alpha,* double *a,* double *b,* double ∗ *resx,* double ∗ *resy,* double ∗ *rese,* int *np,* int *nc*)

14.34.1.4 void cholesky_ecm (double ∗∗ *m,* char ∗ *fileName,* double ∗ *resx,* double ∗ *resy,* double ∗ *rese,* int *np,* int *nc*)

14.34.1.5 int cholesky_formUinv (double ∗∗ *uinv,* double ∗∗ *m,* int *np*)

14.34.1.6 void cholesky_powerlawModel (double ∗∗ *m,* double *modelAlpha,* double *modelFc,* double *modelA,* double ∗ *resx,* double ∗ *resy,* double ∗ *rese,* int *np,* int *nc*)

14.34.1.7 void cholesky_powerlawModel_withBeta (double ∗∗ *m,* double *modelAlpha,* double *beta,* double *modelFc,* double *modelA,* double ∗ *resx,* double ∗ *resy,* double ∗ *rese,* int *np,* int *nc*)

14.34.1.8 void cholesky_readFromCovarianceFunction (double ∗∗ *m,* const char ∗ *fname,* double ∗ *resx,* double ∗ *resy,* double ∗ *rese,* int *np,* int *nc*)

14.35 TKfit.h File Reference

```
#include "TKmatrix.h"
#include "TKlongdouble.h"
```

Include dependency graph for TKfit.h:

Functions

- double TKleastSquares (double ∗b, double ∗white_b, double ∗∗designMatrix, double ∗∗white_designMatrix, int n, int nf, double tol, char rescale_errors, double ∗outP, double ∗e, double ∗∗CVM)
- double TKrobustLeastSquares (double ∗b, double ∗white_b, double ∗∗designMatrix, double ∗∗white_←designMatrix, int n, int nf, double tol, char rescale_errors, double ∗outP, double ∗e, double ∗∗cvm, char robust)
- double TKconstrainedLeastSquares (double ∗b, double ∗white_b, double ∗∗designMatrix, double ∗∗white_←designMatrix, double ∗∗constraintsMatrix, int n, int nf, int nconstraints, double tol, char rescale_errors, double ∗outP, double ∗e, double ∗∗cvm)
- double TKrobustConstrainedLeastSquares (double ∗b, double ∗white_b, double ∗∗designMatrix, double ∗∗white_designMatrix, double ∗∗constraintsMatrix, int n, int nf, int nconstraints, double tol, char rescale←_errors, double ∗outP, double ∗e, double ∗∗cvm, char robust)
- void TKleastSquares_svd (double ∗x, double ∗y, double ∗sig, int n, double ∗p, double ∗e, int nf, double ∗∗cvm, double ∗chisq, void(∗fitFuncs)(double, double[ ], int), int weight)
- void TKleastSquares_svd_noErr (double ∗x, double ∗y, int n, double ∗p, int nf, void(∗fitFuncs)(double, double[ ], int))
- void TKremovePoly_f (float ∗px, float ∗py, int n, int m)
- void TKremovePoly_d (double ∗px, double ∗py, int n, int m)
- void TKfindPoly_d (double ∗px, double ∗py, int n, int m, double ∗p)
- void TKfitPoly (double x, double ∗v, int m)

14.35.1 Function Documentation

14.35.1.1 double TKconstrainedLeastSquares (double ∗ *b,* double ∗ *white_b,* double ∗∗ *designMatrix,* double ∗∗ *white_designMatrix,* double ∗∗ *constraintsMatrix,* int *n,* int *nf,* int *nconstraints,* double *tol,* char *rescale_errors,* double ∗ *outP,* double ∗ *e,* double ∗∗ *cvm*)

14.35.1.2 void TKfindPoly_d (double ∗ *px,* double ∗ *py,* int *n,* int *m,* double ∗ *p*)

14.35.1.3 void TKfitPoly (double *x,* double ∗ *v,* int *m*)

14.35.1.4 double TKleastSquares (double ∗ *b,* double ∗ *white_b,* double ∗∗ *designMatrix,* double ∗∗ *white_designMatrix,* int *n,* int *nf,* double *tol,* char *rescale_errors,* double ∗ *outP,* double ∗ *e,* double ∗∗ *CVM*)

14.35.1.5 void TKleastSquares_svd (double ∗ *x,* double ∗ *y,* double ∗ *sig,* int *n,* double ∗ *p,* double ∗ *e,* int *nf,* double ∗∗ *cvm,* double ∗ *chisq,* void(∗)(double, double[ ], int) *fitFuncs,* int *weight*)

14.35.1.6 void TKleastSquares_svd_noErr (double ∗ *x,* double ∗ *y,* int *n,* double ∗ *p,* int *nf,* void(∗)(double, double[ ], int) *fitFuncs*)

14.35.1.7 void TKremovePoly_d (double ∗ *px,* double ∗ *py,* int *n,* int *m*)

14.35.1.8 void TKremovePoly_f (float ∗ *px,* float ∗ *py,* int *n,* int *m*)

14.35.1.9 double TKrobustConstrainedLeastSquares (double ∗ *b,* double ∗ *white_b,* double ∗∗ *designMatrix,* double ∗∗ *white_designMatrix,* double ∗∗ *constraintsMatrix,* int *n,* int *nf,* int *nconstraints,* double *tol,* char *rescale_errors,* double ∗ *outP,* double ∗ *e,* double ∗∗ *cvm,* char *robust*)

14.35.1.10 double TKrobustLeastSquares (double ∗ *b,* double ∗ *white_b,* double ∗∗ *designMatrix,* double ∗∗ *white_designMatrix,* int *n,* int *nf,* double *tol,* char *rescale_errors,* double ∗ *outP,* double ∗ *e,* double ∗∗ *cvm,* char *robust*)

14.36 TKlog.h File Reference

```
#include <stdio.h>
#include <time.h>
```

Include dependency graph for TKlog.h: This graph shows which files directly or indirectly include this file:

Macros

- #define TK_MAX_ERRORS 16
- #define TK_MAX_ERROR_LEN 128
- #define LOG_OUTFILE stdout
- #define RESETCOLOR "\033[0m"
- #define WARNCOLOR RESETCOLOR "\033[0;35m"
- #define BOLDCOLOR RESETCOLOR "\033[1m"
- #define ERRORCOLOR RESETCOLOR "\033[1;31m"
- #define WHERESTR "[%s:%d] "
- #define WHEREARG __FILE__, __LINE__
- #define ENDL "\n"
- #define WHEREERR ERRORCOLOR "∗∗∗ERROR∗∗∗\n [%s:%d] " RESETCOLOR
- #define WHEREWARN BOLDCOLOR "[%s:%d] " WARNCOLOR "Warning: " RESETCOLOR
- #define ENDERR "\n∗∗∗!!!!!∗∗∗"
- #define WHERETCHK "[%s:%d] T=%.2f s: "
- #define _LOG(_fmt, ...) _TKchklog(LOG_OUTFILE,_fmt,##__VA_ARGS__)
- #define logmsg(_fmt, ...) _LOG(WHERESTR _fmt ENDL, WHEREARG,##__VA_ARGS__)
- #define logdbg(_fmt, ...) if(debugFlag)logmsg(_fmt,##__VA_ARGS__)

- #define logerr(_fmt, ...) do{TK_STORE_ERROR(_fmt,##__VA_ARGS__); _LOG(WHEREERR _fmt ENDERR ENDL, WHEREARG,##__VA_ARGS__);}while(0)
- #define logwarn(_fmt, ...) do{TK_STORE_WARNING(_fmt,##__VA_ARGS__); _LOG(WHEREWARN _fmt ENDL, WHEREARG,##__VA_ARGS__);}while(0)
- #define logtchk(_fmt, ...) if(tcheck)_LOG(WHERETCHK _fmt ENDL, WHEREARG,(clock()-timer_clk)/(float)CLOCKS_PER_SEC,##__VA_ARGS__)
- #define TK_STORE_ERROR(_fmt, ...) if(TK_errorCount < TK_MAX_ERRORS)snprintf(TK_errorlog[TK_errorCount],TK_MAX_ERROR_LEN, _fmt,##__VA_ARGS__); ++TK_errorCount
- #define TK_STORE_WARNING(_fmt, ...) if(TK_warnCount < TK_MAX_ERRORS)snprintf(TK_warnlog[TK_warnCount],TK_MAX_ERROR_LEN, _fmt,##__VA_ARGS__); ++TK_warnCount
- #define DEPRECATED

Functions

- int logerr_check ()
- void _TKchklog (FILE ∗, const char ∗,...)

Variables

- int debugFlag
- int writeResiduals
- int tcheck
- clock_t timer_clk
- unsigned TK_errorCount
- unsigned TK_warnCount
- char TK_errorlog [TK_MAX_ERRORS][TK_MAX_ERROR_LEN]
- char TK_warnlog [TK_MAX_ERRORS][TK_MAX_ERROR_LEN]

14.36.1 Macro Definition Documentation

14.36.1.1 #define _LOG(_fmt, ...) _TKchklog(LOG_OUTFILE,_fmt,##__VA_ARGS__)

14.36.1.2 #define BOLDCOLOR RESETCOLOR "\033[1m"

14.36.1.3 #define DEPRECATED

14.36.1.4 #define ENDERR "\n∗∗∗!!!!!∗∗∗"

14.36.1.5 #define ENDL "\n"

14.36.1.6 #define ERRORCOLOR RESETCOLOR "\033[1;31m"

14.36.1.7 #define LOG_OUTFILE stdout

14.36.1.8 #define logdbg(_fmt, ...) if(debugFlag)logmsg(_fmt,##__VA_ARGS__)

14.36.1.9 #define logerr(_fmt, ...) do{TK_STORE_ERROR(_fmt,##__VA_ARGS__); _LOG(WHEREERR _fmt ENDERR ENDL, WHEREARG,##__VA_ARGS__);}while(0)

14.36.1.10 #define logmsg(_fmt, ...) _LOG(WHERESTR _fmt ENDL, WHEREARG,##__VA_ARGS__)

14.36.1.11 #define logtchk(_fmt, ...) if(tcheck)_LOG(WHERETCHK _fmt ENDL, WHEREARG,(clock()-timer_clk)/(float)CLOCKS_PER_SEC,##__VA_ARGS__)

14.36.1.12 #define logwarn(*_fmt, ...*) do{TK_STORE_WARNING(_fmt,##__VA_ARGS__); _LOG(WHEREWARN _fmt ENDL, WHEREARG,##__VA_ARGS__);}while(0)

14.36.1.13 #define RESETCOLOR "\033[0m"

14.36.1.14 #define TK_MAX_ERROR_LEN 128

14.36.1.15 #define TK_MAX_ERRORS 16

14.36.1.16 #define TK_STORE_ERROR(*_fmt, ...*) if(TK_errorCount $<$ TK_MAX_ERRORS)snprintf(TK_errorlog[TK_errorCount],TK_MAX_ERROR_LEN, _fmt,##__VA_ARGS__); ++TK_errorCount

14.36.1.17 #define TK_STORE_WARNING(*_fmt, ...*) if(TK_warnCount $<$ TK_MAX_ERRORS)snprintf(TK_warnlog[TK_warnCount],TK_MAX_ERROR_LEN, _fmt,##__VA_ARGS__); ++TK_warnCount

14.36.1.18 #define WARNCOLOR RESETCOLOR "\033[0;35m"

14.36.1.19 #define WHEREARG __FILE__, __LINE__

14.36.1.20 #define WHEREERR ERRORCOLOR "∗∗∗ERROR∗∗∗\n [%s:%d] " RESETCOLOR

14.36.1.21 #define WHERESTR "[%s:%d] "

14.36.1.22 #define WHERETCHK "[%s:%d] T=%.2f s: "

14.36.1.23 #define WHEREWARN BOLDCOLOR "[%s:%d] " WARNCOLOR "Warning: " RESETCOLOR

14.36.2 Function Documentation

14.36.2.1 void _TKchklog (FILE ∗ , const char ∗ , *...*)

14.36.2.2 int logerr_check ( )

14.36.3 Variable Documentation

14.36.3.1 int debugFlag

14.36.3.2 int tcheck

14.36.3.3 clock_t timer_clk

14.36.3.4 unsigned TK_errorCount

14.36.3.5 char TK_errorlog[TK_MAX_ERRORS][TK_MAX_ERROR_LEN]

14.36.3.6 unsigned TK_warnCount

14.36.3.7 char TK_warnlog[TK_MAX_ERRORS][TK_MAX_ERROR_LEN]

14.36.3.8 int writeResiduals

14.37 TKlongdouble.float128.h File Reference

```
#include <math.h>
```

```
#include <quadmath.h>
```

Include dependency graph for TKlongdouble.float128.h:

Macros

- #define USE_BUILTIN_LONGDOUBLE
- #define LONGDOUBLE_IS_FLOAT128
- #define LONGDOUBLE_ONE 1.0Q
- #define longdouble(a) a##Q
- #define FMT_LD "Q"
- #define LD_PI M_PIq
- #define cosl cosq
- #define sinl sinq
- #define floorl floorq
- #define fabsl fabsq
- #define powl powq

Typedefs

- typedef __float128 longdouble

Functions

- longdouble parse_longdouble (const char ∗str)
- int ld_printf (const char ∗__format,...)
- int ld_fprintf (FILE ∗__stream, const char ∗__format,...)
- int ld_sprintf (char ∗__str, const char ∗__format,...)

14.37.1 Macro Definition Documentation

14.37.1.1 #define cosl cosq

14.37.1.2 #define fabsl fabsq

14.37.1.3 #define floorl floorq

14.37.1.4 #define FMT_LD "Q"

14.37.1.5 #define LD_PI M_PIq

14.37.1.6 #define longdouble(*a*) a##Q

14.37.1.7 #define LONGDOUBLE_IS_FLOAT128

14.37.1.8 #define LONGDOUBLE_ONE 1.0Q

14.37.1.9 #define powl powq

14.37.1.10 #define sinl sinq

14.37.1.11 #define USE_BUILTIN_LONGDOUBLE

14.37.2 Typedef Documentation

14.37.2.1 typedef __float128 longdouble

14.37.3 Function Documentation

14.37.3.1 int ld_fprintf (FILE ∗ *__stream,* const char ∗ *__format,* *...*)

14.37.3.2 int ld_printf (const char ∗ *__format,* *...*)

14.37.3.3 int ld_sprintf (char ∗ *__str,* const char ∗ *__format,* *...*)

14.37.3.4 longdouble parse_longdouble (const char ∗ *str*)

14.38 TKlongdouble.h File Reference

```
#include <math.h>
```

Include dependency graph for TKlongdouble.h: This graph shows which files directly or indirectly include this file:

Macros

- #define USE_BUILTIN_LONGDOUBLE
- #define longdouble(a) a##L
- #define LD_PI M_PI
- #define LONGDOUBLE_IS_IEEE754
- #define LONGDOUBLE_ONE 1.0L
- #define ld_printf printf
- #define ld_fprintf fprintf
- #define ld_sprintf sprintf

Typedefs

- typedef long double longdouble

Functions

- longdouble parse_longdouble (const char ∗str)

14.38.1 Macro Definition Documentation

14.38.1.1 #define ld_fprintf fprintf

14.38.1.2 #define LD_PI M_PI

14.38.1.3 #define ld_printf printf

14.38.1.4 #define ld_sprintf sprintf

14.38.1.5 #define longdouble(*a*) a##L

14.38.1.6 #define LONGDOUBLE_IS_IEEE754

14.38.1.7 #define LONGDOUBLE_ONE 1.0L

14.38.1.8 #define USE_BUILTIN_LONGDOUBLE

14.38.2 Typedef Documentation

14.38.2.1 typedef long double **longdouble**

14.38.3 Function Documentation

14.38.3.1 **longdouble** parse_longdouble (const char ∗ *str*)

14.39 TKlongdouble.ld.h File Reference

```
#include <math.h>
```

Include dependency graph for TKlongdouble.ld.h:

Macros

- #define USE_BUILTIN_LONGDOUBLE
- #define longdouble(a) a##L
- #define LD_PI M_PI
- #define LONGDOUBLE_IS_IEEE754
- #define LONGDOUBLE_ONE 1.0L
- #define ld_printf printf
- #define ld_fprintf fprintf
- #define ld_sprintf sprintf

Typedefs

- typedef long double longdouble

Functions

- longdouble parse_longdouble (const char ∗str)

14.39.1 Macro Definition Documentation

14.39.1.1 #define ld_fprintf fprintf

14.39.1.2 #define LD_PI M_PI

14.39.1.3 #define ld_printf printf

14.39.1.4 #define ld_sprintf sprintf

14.39.1.5 #define **longdouble**(*a*) a##L

14.39.1.6 #define LONGDOUBLE_IS_IEEE754

14.39.1.7 #define LONGDOUBLE_ONE 1.0L

14.39.1.8 #define USE_BUILTIN_LONGDOUBLE

14.39.2 Typedef Documentation

14.39.2.1 typedef long double **longdouble**

14.39.3 Function Documentation

14.39.3.1 **longdouble** parse_longdouble (const char ∗ *str*)

14.40 TKmatrix.h File Reference

This graph shows which files directly or indirectly include this file:

Functions

- void TKmultMatrix_sq (double ∗∗idcm, double ∗∗u, int ndata, int npol, double ∗∗uout)
- void TKmultMatrixVec_sq (double ∗∗idcm, double ∗b, int ndata, double ∗bout)
- void TKmultMatrix (double ∗∗idcm, double ∗∗u, int ndata, int ndata2, int npol, double ∗∗uout)
- void TKmultMatrixVec (double ∗∗idcm, double ∗b, int ndata, int ndata2, double ∗bout)
- double ∗∗ malloc_uinv (int n)
- double ∗∗ malloc_blas (int n, int m)
- void free_blas (double ∗∗matrix)
- void free_uinv (double ∗∗uinv)
- int get_blas_rows (double ∗∗uinv)
- int get_blas_cols (double ∗∗uinv)
- float ∗∗ malloc_2df (int rows, int cols)
- void free_2df (float ∗∗uinv)

14.40.1 Function Documentation

14.40.1.1 void free_2df (float ∗∗ *uinv*)

14.40.1.2 void free_blas (double ∗∗ *matrix*)

14.40.1.3 void free_uinv (double ∗∗ *uinv*)

14.40.1.4 int get_blas_cols (double ∗∗ *uinv*)

14.40.1.5 int get_blas_rows (double ∗∗ *uinv*)

14.40.1.6 float∗∗ malloc_2df (int *rows,* int *cols*)

14.40.1.7 double∗∗ malloc_blas (int *n,* int *m*)

14.40.1.8 double∗∗ malloc_uinv (int *n*)

14.40.1.9 void TKmultMatrix (double ∗∗ *idcm,* double ∗∗ *u,* int *ndata,* int *ndata2,* int *npol,* double ∗∗ *uout*)

14.40.1.10 void TKmultMatrix_sq (double ∗∗ *idcm,* double ∗∗ *u,* int *ndata,* int *npol,* double ∗∗ *uout*)

14.40.1.11 void TKmultMatrixVec (double ∗∗ *idcm,* double ∗ *b,* int *ndata,* int *ndata2,* double ∗ *bout*)

14.40.1.12 void TKmultMatrixVec_sq (double ∗∗ *idcm,* double ∗ *b,* int *ndata,* double ∗ *bout*)

14.41 TKspectrum.h File Reference

Classes

- struct complexVal

Macros

- #define ABS(x) ((x) < 0 ? -(x) : (x))
- #define MAX(x, y) ((x) > (y) ? (x) : (y))
- #define MIN(x, y) ((x) < (y) ? (x) : (y))

Typedefs

- typedef struct complexVal complexVal

Functions

- void getprtj (int n)
- void indexx8 (int n, double ∗arrin, int ∗indx)
- void getweights (int n, double ∗wt)
- void fit4 (int ∗nfit, double ∗p4, double ∗cov4, int ndostats, double ∗chidf, double ∗avewt)
- void mat20 (double sam[21][21], double a[21][21], int n, double ∗determ, int ∗nbad)
- void sineFunc (double x, double ∗v, int ma)
- void TKsortit (double ∗x, double ∗y, int n)
- void TKaveragePts (double ∗x, double ∗y, int n, int width, double ∗meanX, double ∗meanY, int ∗nMean)
- void TKcmonot (int n, double x[ ], double y[ ], double yd[ ][4])
- void TKspline_interpolate (int n, double ∗x, double ∗y, double yd[ ][4], double ∗interpX, double ∗interpY, int nInterp)
- void TKinterpolateSplineSmoothFixedXPts (double ∗inX, double ∗inY, int inN, double ∗interpX, double ∗interpY, int nInterp)
- void TKhann (double ∗x, double ∗y, int n, double ∗ox, double ∗oy, int ∗on, int width)
- void TKfirstDifference (double ∗x, double ∗y, int n)
- void TK_fitSine (double ∗x, double ∗y, double ∗e, int n, int wErr, double ∗outX, double ∗outY, int ∗outN)
- void TKlomb_d (double ∗x, double ∗y, int n, double ofac, double hifac, double ∗ox, double ∗oy, int ∗outN, double ∗var)
- int TK_fft (short int dir, long n, double ∗x, double ∗y)
- void TK_dft (double ∗x, double ∗y, int n, double ∗outX, double ∗outY, int ∗outN, double ∗outY_re, double ∗outY_im)
- void TK_weightLS (double ∗x, double ∗y, double ∗sig, int n, double ∗outX, double ∗outY, int ∗outN, double ∗outY_re, double ∗outY_im)
- void TK_fitSinusoids (double ∗x, double ∗y, double ∗sig, int n, double ∗outX, double ∗outY, int ∗outN)
- int calcSpectraErr (double ∗∗uinv, double ∗resx, double ∗resy, int nres, double ∗specX, double ∗specY, double ∗specE, int nfit)
- double TKspectrum (double ∗x, double ∗y, double ∗e, int n, int averageTime, int smoothWidth, int smooth←Type, int fitSpline, int preWhite, int specType, double ofac, double hifac, int specOut, double ∗outX, double ∗outY, int ∗nout, int calcWhite, int output, double ∗outY_re, double ∗outY_im)
- void TKboxcar (double ∗x, double ∗y, int n, double ∗ox, double ∗oy, int ∗on, int width)
- int calcSpectra (double ∗∗uinv, double ∗resx, double ∗resy, int nres, double ∗specX, double ∗specY, int nfit)

Variables

- bool verbose_calc_spectra

14.41.1 Macro Definition Documentation

14.41.1.1 #define ABS(*x*) ((x) $<$ 0 ? -(x) : (x))

14.41.1.2 #define MAX(*x*, *y*) ((x) $>$ (y) ? (x) : (y))

14.41.1.3 #define MIN(*x*, *y*) ((x) $<$ (y) ? (x) : (y))

14.41.2 Typedef Documentation

14.41.2.1 typedef struct complexVal complexVal

14.41.3 Function Documentation

14.41.3.1 int calcSpectra (double ∗∗ *uinv*, double ∗ *resx*, double ∗ *resy*, int *nres*, double ∗ *specX*, double ∗ *specY*, int *nfit*)

14.41.3.2 int calcSpectraErr (double ∗∗ *uinv*, double ∗ *resx*, double ∗ *resy*, int *nres*, double ∗ *specX*, double ∗ *specY*, double ∗ *specE*, int *nfit*)

14.41.3.3 void fit4 (int ∗ *nfit*, double ∗ *p4*, double ∗ *cov4*, int *ndostats*, double ∗ *chidf*, double ∗ *avewt*)

14.41.3.4 void getprtj (int *n*)

14.41.3.5 void getweights (int *n*, double ∗ *wt*)

14.41.3.6 void indexx8 (int *n*, double ∗ *arrin*, int ∗ *indx*)

14.41.3.7 void mat20 (double *sam[21][21]*, double *a[21][21]*, int *n*, double ∗ *determ*, int ∗ *nbad*)

14.41.3.8 void sineFunc (double *x*, double ∗ *v*, int *ma*)

14.41.3.9 void TK_dft (double ∗ *x*, double ∗ *y*, int *n*, double ∗ *outX*, double ∗ *outY*, int ∗ *outN*, double ∗ *outY_re*, double ∗ *outY_im*)

14.41.3.10 int TK_fft (short int *dir*, long *n*, double ∗ *x*, double ∗ *y*)

14.41.3.11 void TK_fitSine (double ∗ *x*, double ∗ *y*, double ∗ *e*, int *n*, int *wErr*, double ∗ *outX*, double ∗ *outY*, int ∗ *outN*)

14.41.3.12 void TK_fitSinusoids (double ∗ *x*, double ∗ *y*, double ∗ *sig*, int *n*, double ∗ *outX*, double ∗ *outY*, int ∗ *outN*)

14.41.3.13 void TK_weightLS (double ∗ *x*, double ∗ *y*, double ∗ *sig*, int *n*, double ∗ *outX*, double ∗ *outY*, int ∗ *outN*, double ∗ *outY_re*, double ∗ *outY_im*)

14.41.3.14 void TKaveragePts (double ∗ *x*, double ∗ *y*, int *n*, int *width*, double ∗ *meanX*, double ∗ *meanY*, int ∗ *nMean*)

14.41.3.15 void TKboxcar (double ∗ *x*, double ∗ *y*, int *n*, double ∗ *ox*, double ∗ *oy*, int ∗ *on*, int *width*)

14.41.3.16 void TKcmonot (int *n*, double *x[ ]*, double *y[ ]*, double *yd[ ][4]*)

14.41.3.17 void TKfirstDifference (double ∗ *x*, double ∗ *y*, int *n*)

14.41.3.18 void TKhann (double ∗ *x*, double ∗ *y*, int *n*, double ∗ *ox*, double ∗ *oy*, int ∗ *on*, int *width*)

14.41.3.19 void TKinterpolateSplineSmoothFixedXPts (double ∗ *inX*, double ∗ *inY*, int *inN*, double ∗ *interpX*, double ∗ *interpY*, int *nInterp*)

14.41.3.20 void TKlomb_d (double ∗ *x,* double ∗ *y,* int *n,* double *ofac,* double *hifac,* double ∗ *ox,* double ∗ *oy,* int ∗ *outN,* double ∗ *var*)

14.41.3.21 void TKsortit (double ∗ *x,* double ∗ *y,* int *n*)

14.41.3.22 double TKspectrum (double ∗ *x,* double ∗ *y,* double ∗ *e,* int *n,* int *averageTime,* int *smoothWidth,* int *smoothType,* int *fitSpline,* int *preWhite,* int *specType,* double *ofac,* double *hifac,* int *specOut,* double ∗ *outX,* double ∗ *outY,* int ∗ *nout,* int *calcWhite,* int *output,* double ∗ *outY_re,* double ∗ *outY_im*)

14.41.3.23 void TKspline_interpolate (int *n,* double ∗ *x,* double ∗ *y,* double *yd[ ][4],* double ∗ *interpX,* double ∗ *interpY,* int *nInterp*)

14.41.4 Variable Documentation

14.41.4.1 bool verbose_calc_spectra

14.42 TKsvd.h File Reference

Functions

- void TKsingularValueDecomposition_lsq (longdouble ∗∗designMatrix, int n, int nf, longdouble ∗∗v, longdouble ∗w, longdouble ∗∗u)
- void TKbacksubstitution_svd (longdouble ∗∗V, longdouble ∗w, longdouble ∗∗U, longdouble ∗b, longdouble ∗x, int n, int nf)
- longdouble TKpythag (longdouble a, longdouble b)
- void TKbidiagonal (longdouble ∗∗a, longdouble ∗anorm, int ndata, int nfit, longdouble ∗∗v, longdouble ∗w, longdouble ∗∗u, longdouble ∗rv1)

14.42.1 Function Documentation

14.42.1.1 void TKbacksubstitution_svd (longdouble ∗∗ *V,* longdouble ∗ *w,* longdouble ∗∗ *U,* longdouble ∗ *b,* longdouble ∗ *x,* int *n,* int *nf*)

14.42.1.2 void TKbidiagonal (longdouble ∗∗ *a,* longdouble ∗ *anorm,* int *ndata,* int *nfit,* longdouble ∗∗ *v,* longdouble ∗ *w,* longdouble ∗∗ *u,* longdouble ∗ *rv1*)

14.42.1.3 longdouble TKpythag (longdouble *a,* longdouble *b*)

14.42.1.4 void TKsingularValueDecomposition_lsq (longdouble ∗∗ *designMatrix,* int *n,* int *nf,* longdouble ∗∗ *v,* longdouble ∗ *w,* longdouble ∗∗ *u*)

Index