Astronomical Data Analysis Software and Systems XXII
ASP Conference Series, Vol. 475
Douglas N. Friedel, ed.

PAL: A Positional Astronomy Library

Tim Jenness and David S. Berry

Joint Astronomy Centre, 660 N. A'ohōkū Place, Hilo, HI, 96720, U.S.A.

Abstract. PAL is a new positional astronomy library written in C that attempts to retain the SLALIB API but is distributed with an open source GPL license. The library depends on the IAU SOFA library wherever a SOFA routine exists and uses the most recent nutation and precession models. Currently about 100 of the 200 SLALIB routines are available. Interfaces are also available from Perl and Python. PAL is freely available via github.

1. Introduction

The SLALIB library, written by Patrick Wallace (Wallace 1994), is a very popular positional astronomy library that is available in Fortran as part of the Starlink Software Collection (e.g. Jenness et al. 2009) with a GPL license. An updated version is also available directly from the author as a proprietary and extended C library with a non-commercial non-redistribution license. The latter clause makes it hard to ship software relying on the C library and impossible to include the code in a public source code repository.

To overcome the distribution issues and to make use of current precession and nutation models, we have written, with advice and input from Patrick Wallace, the PAL library. The PAL library is written in C and has a GPL license. In most cases the API is designed to be identical to SLALIB except for the use of a `pal` prefix instead of a `sla` prefix in function calls. Where appropriate, IAU SOFA routines (Hohenkerk 2010; Wallace 1996), of which Patrick Wallace was the primary author, are called. Where the SOFA library does not have equivalent functionality the Fortran has been ported to C from the GPL Fortran library included with the Starlink distribution.

We have not ported the full SLALIB functionality to PAL but are adding routines as we need them for applications. Approximately 100 functions have been ported.

2. Implementation

PAL is written in very portable C with SOFA as the only dependency. Perl and Python wrappers have also been written and they are distributed with their own copies of PAL and SOFA to make installation as easy as possible.

The example code below shows the deliberate similarities between the SLALIB and PAL API:

SLALIB Fortran:

```
GMST = SLA_GMST( UT1 )
CALL SLA_DE2H( HIN, DIN, DP, DA, DE )
CALL SLA_DMOON( DATE, PV )
CALL SLA_OBS( N, C, NAME, W, P, H )
```

SLALIB C:

```
gmst = slaGmst( ut1 );
slaDe2h( hin, din, dp, &da, &de );
slaDmoon( date, pv );
slaObs( 0, "JCMT", telname, &w, &p, &h );
```

PAL C:

```
gmst = palGmst( ut1 );
palDe2h( hin, din, dp, &da, &de );
palDmoon( date, pv );
status = palObs( 0, "JCMT", short, slen,
                 long, llen, &w, &p, &h );
```

PAL Python:

```
import palpy as pal
gmst = pal.gmst( ut1 )
(da, de) = pal.de2h( hin, din, dp )
pv = pal.dmoon( date )
obsdata = pal.obs()
mmt = obsdata["MMT"]
```

PAL Perl:

```
use Astro::PAL qw/ :all /;
$gmst = palGmst( $ut1 );
($da,$de) = palDe2h( $hin, $din, $dp );
@pv = palDmoon( $date );
($id, $name, $w, $p, $h) = palObs( 22 );
($id, $name, $w, $p, $h) = palObs( "JCMT" );
```

In general a simple renaming of the SLALIB function will be sufficient to switch to PAL with one exception. As can be seen above the `palObs` routine now expects to be given the lengths of supplied string buffers and does not reuse the short name buffer, allowing const strings to be supplied without triggering compiler warnings.

The Python and Perl interfaces return results where appropriate rather than modifying arguments. The Python interface uses numpy arrays (e.g. van der Walt et al. 2011)[1] for all vectors and matrices, and the C library is wrapped using Cython[2]. The Perl interface is wrapped with the standard perl/XS system (e.g. Jenness & Cozens 2002) and uses simple lists for vectors and

[1] `http://numpy.scipy.org`

[2] `http://cython.org`

matrices rather than adding a dependency on the Perl Data Language (Glazebrook & Economou 1997).

The Fortran test suite was ported to C to test the PAL library. There are minor changes due to differences in the more modern precession and nutation models implemented in SOFA.

The PAL library is now used within the Starlink AST library (Berry & Jenness 2012) and in all Starlink C applications that previously used SLALIB. It was shipped with the Starlink *kapuahi* release (Berry et al. 2013).

The PAL library also has an advantage over Fortran SLALIB in that it is inherently thread-safe.

3. Obtaining the Software

PAL is available from github[3] and is also distributed with Starlink[4]. The Python and Perl wrappers are also on github[5] and distributions can be obtained from PyPI[6] and CPAN[7].

References

Berry, D., et al. 2013, in ADASS XXII, edited by D. Friedel (San Francisco: ASP), vol. 475 of ASP Conf. Ser., 247

Berry, D. S., & Jenness, T. 2012, in Astronomical Data Analysis Software and Systems XXI, edited by P. Ballester, D. Egret, & N. P. F. Lorente (San Francisco: ASP), vol. 461 of Astronomical Society of the Pacific Conference Series, 825

Glazebrook, K., & Economou, F. 1997, The Perl Journal, 5, 5. Reprinted in Dr. Dobb's Journal September 1997

Hohenkerk, C. 2010, Scholarpedia, 5, 11404. doi:10.4249/scholarpedia.11404

Jenness, T., & Cozens, S. 2002, Extending and Embedding Perl (Manning Publications)

Jenness, T., et al. 2009, in Astronomical Data Analysis Software and Systems XVIII, edited by D. A. Bohlender, D. Durand, & P. Dowler (San Francisco: ASP), vol. 411 of Astronomical Society of the Pacific Conference Series, 418

van der Walt, S., Colbert, S. C., & Varoquaux, G. 2011, Computing in Science and Engineering, 13, 22. arXiv:1102.1523

Wallace, P. T. 1994, in Astronomical Data Analysis Software and Systems III, edited by D. R. Crabtree, R. J. Hanisch, & J. Barnes (San Francisco: ASP), vol. 61 of Astronomical Society of the Pacific Conference Series, 481

— 1996, in Astronomical Data Analysis Software and Systems V, edited by G. H. Jacoby, & J. Barnes (San Francisco: ASP), vol. 101 of Astronomical Society of the Pacific Conference Series, 207

---

[3] `https://github.com/Starlink/pal/downloads`

[4] `http://www.starlink.ac.uk`

[5] `https://github.com/Starlink/palpy` and `https://github.com/timj/perl-Astro-PAL`

[6] `http://pypi.python.org/pypi/palpy`

[7] `http://search.cpan.org/~tjenness/Astro-PAL/`