JULIA INTERFACES TO STANDARD EPHEMERIS PLATFORMS

A PREPRINT

Joseph D Carpinelli
joey@loopy.codes

2024-04-21

ABSTRACT

Solar system ephemerides are available for free to researchers, students, and professionals in-industry through open source tools, and REST APIs, and web interfaces. Users commonly parse this data programatically with dynamic programming languages, including Python and Julia. This document presents several Julia packages which can aid ephemeris users in sourcing and parsing data with replicatability. Rather than include solar system ephemeris files in source code distrbutions, ephemeris data sourcing can be accomplished directly in-code. Three packages which interface to the JPL SPICE ephemeris platform are presented: `SPICEApplications.jl`, `SPICEKernels.jl`, and `SPICEBodies.jl`. In addition, two packages which interface with the JPL HORIZONS ephemeris platform are presented: `HorizonsAPI.jl` and `HorizonsEphemeris.jl`. All packages are described in-detail in their common documentation site: `ephemeris.loopy.codes`.

1 Introduction

Students and professionals in astronomy, astrodynamics, astrophysics, and other related fields often download and parse solar system ephemeris data from two major providers: Generic JPL SPICE Kernels, and JPL Horizons. SPICE Kernels are typically read through the SPICE Toolkit, which is available in a variety of programming languages, include the C Programming Language with `CSPICE` (Acton 1996). The Julia packages `CSPICE_jll.jl` and `SPICE.jl` expose many `CSPICE` functions through Julia functions. Julia users can load and interact with SPICE kernels `SPICE.furnsh` and `SPICE.spkez`. Horizons data is available through a variety of methods, including email, command-line, graphical web interfaces, and a REST API.

This paper introduces several packages which allow users to download and process ephemeris data idomatically, all from within Julia. Through the use of these packages, users can share replicatable code which automatically fetches publicly available ephemeris data, as opposed to manually including ephemeris data files with their source code distribution.

2 Statement of Need

While ephemeris users have all of the tools they need to fetch and parse ephemeris data within Julia, they do not have the tools to do so *simply* or *idiomatically*. Section 2.1 and Section 2.2 present the research needs filled by each of the five packages introduced in this paper.

2.1 JPL HORIZONS

The two HORIZONS-related packages presented in this paper — `HorizonsAPI.jl` and `HorizonsEphemeris.jl` — are respectively the first Julia packages to precisely match the REST API with tab-completion through static keyword arguments, and the first to offer automatic response parsing into `NamedTuple` types. The `NamedTuple` output of `HorizonsEphemeris.ephemeris`, the top-level method for fetching Cartesian state vectors from the HORIZONS platform, allows for easy plotting, file-saving, and `DataFrame` construction. Both `HorizonsAPI.jl` and `HorizonsEphemeris.jl` offer users a simple, repeatable way to query and parse HORIZONS ephemeris data.

2.2 JPL SPICE

The three SPICE-related packages presented in this paper — `SPICEApplications.jl`, `SPICEKernels.jl`, and `SPICEBodies.jl` — provide idiomatic kernel fetching, inspection, and caching from within Julia. While SPICE

Toolkit executables were *bundled* in Julia through `CSPICE_jll`, they have not been previously *exposed* through Julia functions. `SPICEApplications.jl` wraps each executable with a Julia function, allowing users to easily call SPICE Toolkit executables within their Julia programs, just as they can with `CSPICE` routines wrapped in `SPICE.jl`.

Julia users interact with SPICE kernels by downloading publicly-available aGeneric Kernels, and parsing the data using `SPICE.jl`, or another ephemeris parsing source. This workflow requires that users know how to find the correct generic kernels for their chosen application, and that they know how to use CSPICE functions to retrieve their desired data. `SPICEKernels.jl` and `SPICEBodies.jl` offer idiomatic interfaces to ephemeris fetching and parsing parsing respectively. Continuous integration in the `SPICEKernels.jl` repository multiple times daily, and automatically exports all available generic kernels as variables in Julia. SPICE Toolkit executables (provided by `SPICEApplications.jl`) are used to retrieve a description of each kernel's contents, and place that description in the Julia variable's docstring. As a result, users can use tab-completion and Julia's built-in documentation tools to inspect kernel contents, and download the correct kernel for their application. Once each kernel is downloaded and loaded into the SPICE kernel pool with `SPICE.jl`, users can use `SPICEBodies.jl` to idiomatically fetch data at a provided instance in time.

3 Usage

For detailed usage examples, consult the common documentation site for all of the packages presented in this paper. The code examples below show how a user may retrieve data from the HORIZONS platform, inspect a SPICE kernel before downloading it, and retrieve Cartesian state data at an instance in time.

Source: Article Notebook

Listing 1 Querying JPL HORIZONS in Julia

```
using Dates, DataFrames
using HorizonsEphemeris

ephemeris("earth", now()) |> DataFrame
```

	t	cal	x	y	z		
	Float64	String31	Float64	Float64	Float64	Float64	
1	2.46042e6	A.D. 2024-Apr-21 16:06:17.0241	-0.862456	-0.488047	-0.211335	0.00876658	...

Source: Article Notebook

Listing 2 Inspecting Generic SPICE Kernels in Julia

```
using SPICEApplications, SPICEKernels

brief(de440s()); # alternatively, check the kernel variable's docstring: @doc(de440s)
```

```
BRIEF -- Version 4.1.0, September 17, 2021 -- Toolkit Version N0067


Summary for: /Users/joey/.julia/scratchspaces/8e9d28ce-e483-4ef7-bfd9-45b8fef6369c/kernels/de440s.bsp

Bodies: MERCURY BARYCENTER (1)  SATURN BARYCENTER (6)   MERCURY (199)
        VENUS BARYCENTER (2)    URANUS BARYCENTER (7)   VENUS (299)
        EARTH BARYCENTER (3)    NEPTUNE BARYCENTER (8)  MOON (301)
        MARS BARYCENTER (4)     PLUTO BARYCENTER (9)    EARTH (399)
        JUPITER BARYCENTER (5)  SUN (10)
        Start of Interval (ET)              End of Interval (ET)
        ------------------------------      ------------------------------
        1849 DEC 26 00:00:00.000            2150 JAN 22 00:00:00.000
```

Source: Article Notebook

Listing 3 Using SPICE Kernels in Julia

```
using Dates, SPICE
using SPICEKernels, SPICEBodies

return furnsh(
    de432s(),                   # position and velocity data for nearby planets
    latest_leapseconds_lsk(),   # timekeeping, parsing epochs
    gm_de440(),                 # mass parameters for major solar system bodies
    pck00011(),                 # physical properties of major solar system bodies
)

earth = KernelBody("earth")
x, y, z, x, y, z = earth(now())
```

```
6-element Vector{Float64}:
  -1.2902152989585207e8
  -7.301079489346015e7
  -3.161527373450055e7
  15.178959061081855
 -23.355316033584334
 -10.12327706840625
```

Source: Article Notebook

4 External Packages

The packages presented in this paper which interact with the SPICE Toolkit require users to use `SPICE.jl`, or another SPICE-compatible kernel loading tool. Support for other SPICE kernel management packages, such as `Ephemerides.jl`, may be added in the future.

In addition to the packages in this paper which interface with the JPL HORIZONS ephemeris platform, the `HORIZONS.JL` package offers simplified interfaces for constructing and sending queries to the JPL HORIZONS REST API.

5 References

Source: Article Notebook

Acton, C. H. 1996. "Ancillary Data Services of NASA's Navigation and Ancillary Information Facility." *Planetary and Space Science* 44 (1): 65–70.