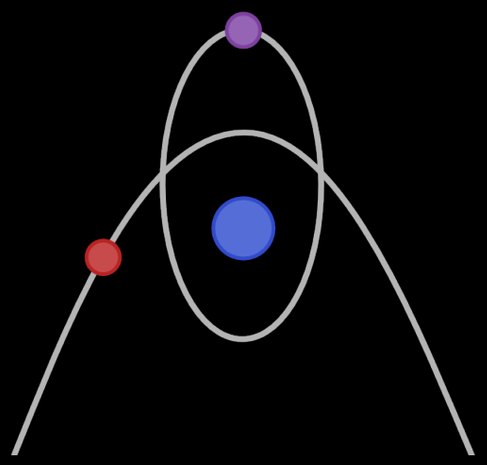
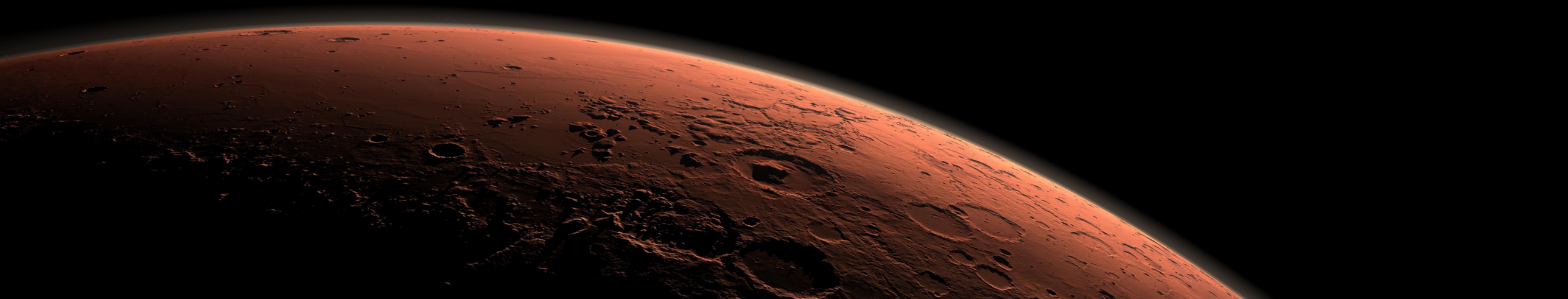


Astrodynamics.jl



AN OPEN-SOURCE FRAMEWORK FOR INTERACTIVE HIGH-PERFORMANCE MISSION ANALYSIS

HELGE EICHHORN – OSCW 23/11/2017

https://github.com/helgee/oscw-2017

What implicit assumptions are holding us back?

Reusable

New Shepard

pace Shuttle

Affordable

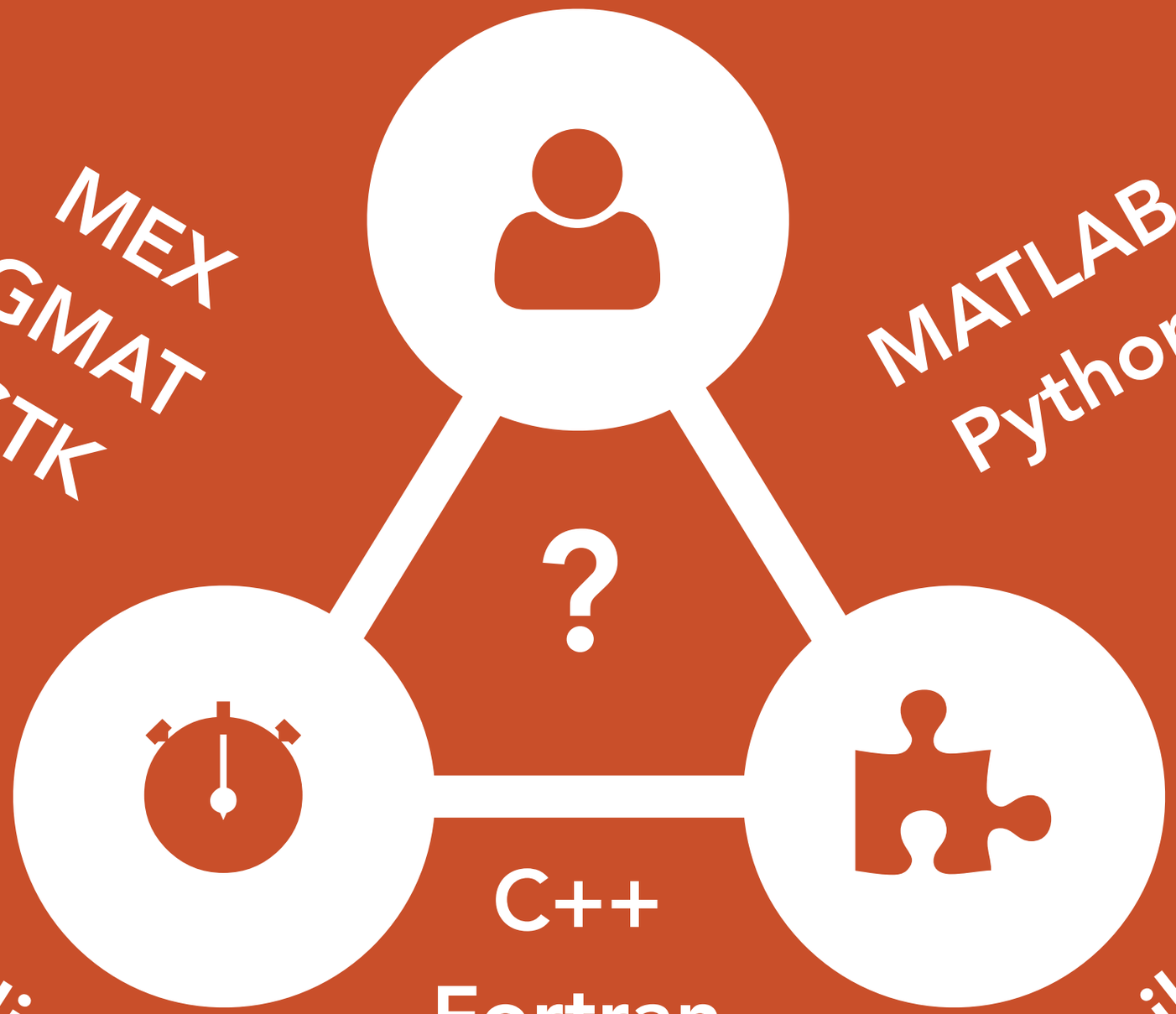
Orbital

LOL, NOPE.

Interactive

MEX
GMAT
STK

MATLAB
Python



High Performance

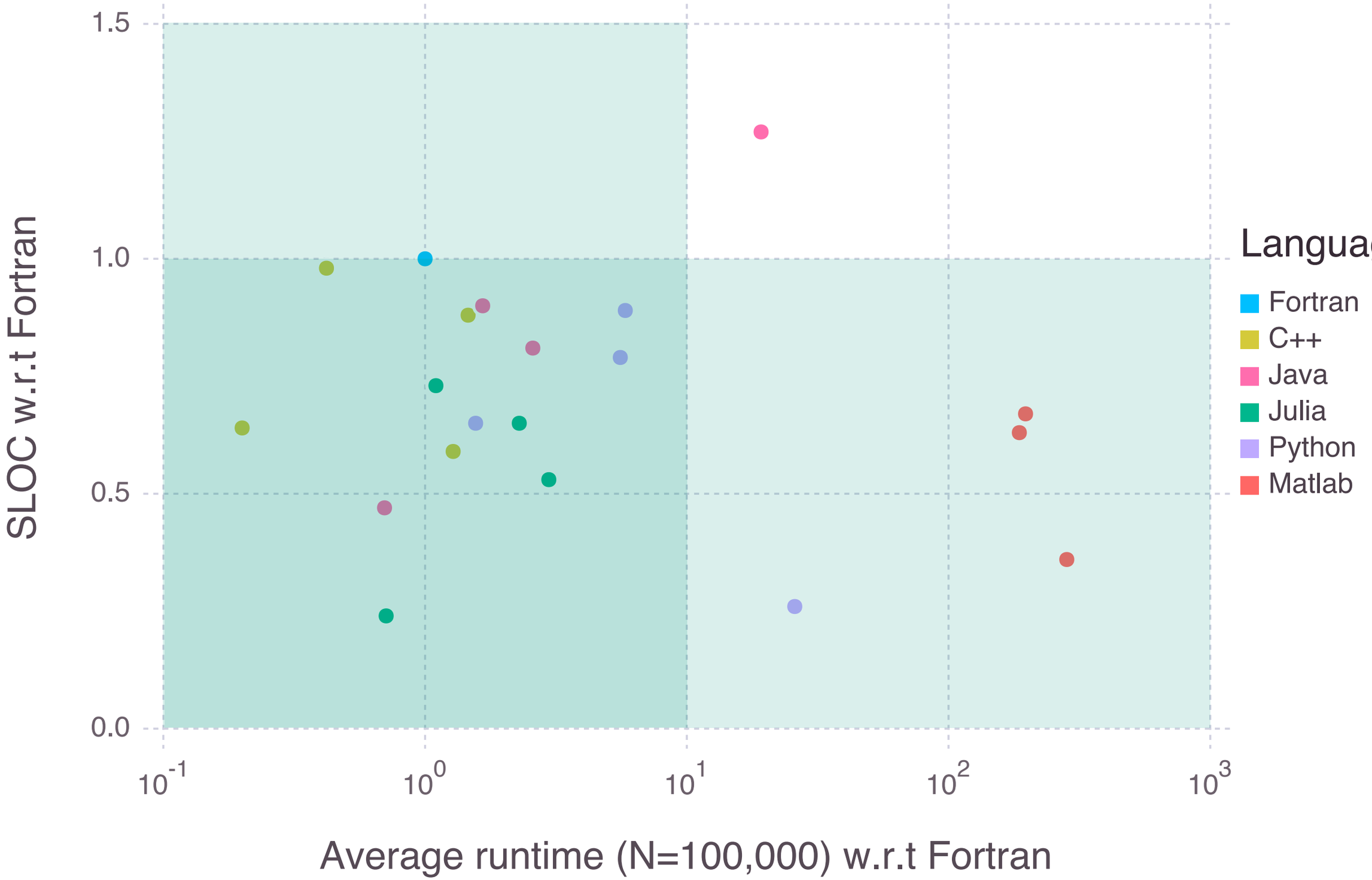
C++
Fortran
Java

Easily Extensible

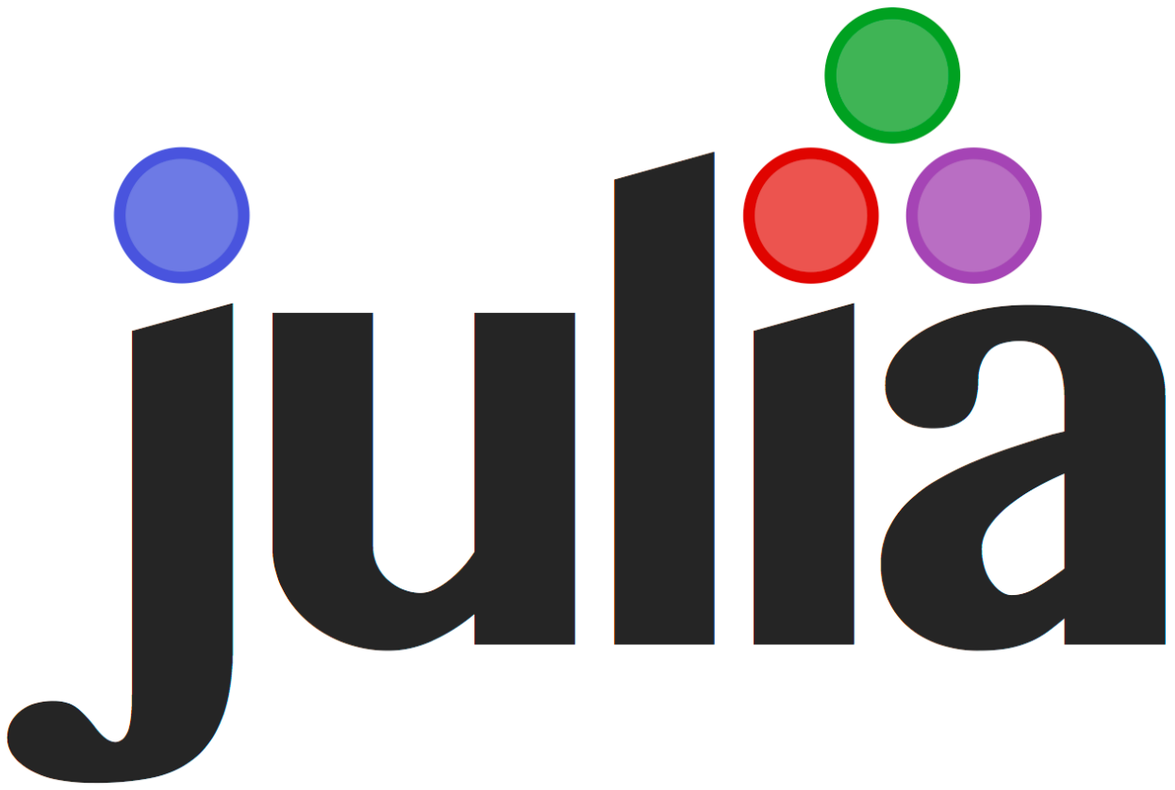
ICATT STUDY

1. Calculating the Keplerian orbital elements
2. Solving Kepler's equation
3. Solving Lambert's problem
4. Calling the DOP853 Fortran 77 code

Fortran, C++, Java, MATLAB, Python, **Julia**


1.5
1.0
0.5
0.0
SLOC w.r.t Fortran
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
Average runtime (N=100,000) w.r.t Fortran
Languages
Fortran
C++
Java
Julia
Python
Matlab

How I Learned to Stop Worrying and Love the JIT


julia

Astrodynamics.jl

https://github.com/JuliaAstrodynamics/Astrodynamics.jl

Example

High performance

Extensible at runtime

API for humans

Well documented

Make STK obsolete

How I can I help you get your mission of the ground?

THANK YOU VERY MUCH!