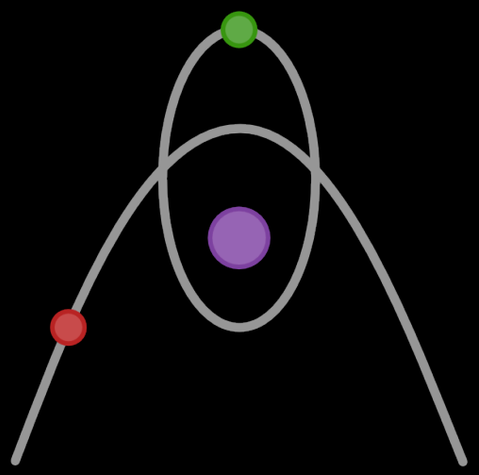
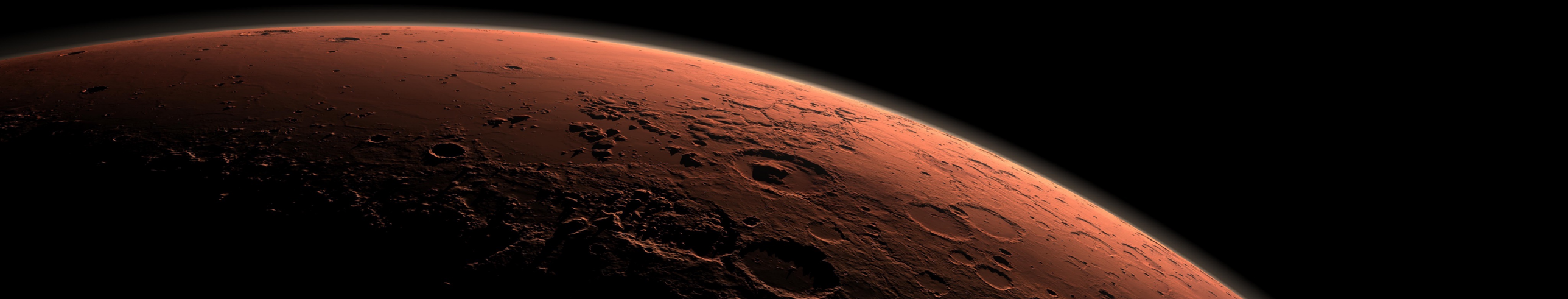


# Astrodynamics.jl



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

HELGE EICHHORN – OSCW 23/11/2017

# What implicit assumptions are holding us back?

# Reusable

Affordable

Orbital

Reusable

Vega

Affordable

Orbital

Reusable

Space Shuttle

Affordable

Orbital

Reusable

New Shepard

Affordable

Orbital

Reusable


Affordable

Orbital

# Interactive

Fast

Easily Extensible

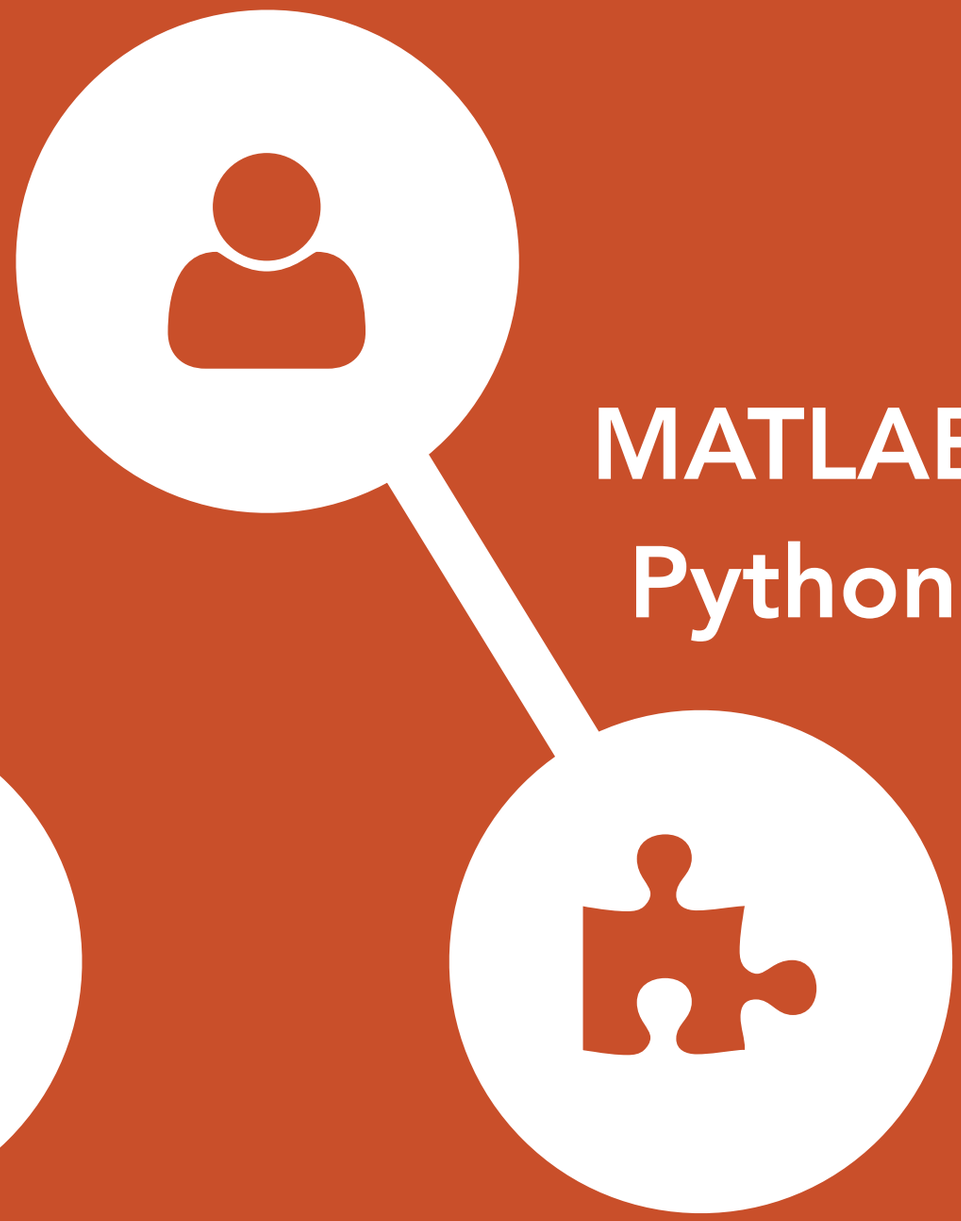
# Interactive

C++
Fortran
Java

Fast

Easily
Extensible

Interactive



MATLAB
Python

Fast

Easily
Extensible

Interactive

MEX
GMAT
STK

Fast

Easily
Extensible

# Interactive

?

Fast

Easily
Extensible

# ICATT 2016 STUDY

Eichhorn, H., Cano, J.L., McLean, F. et al. CEAS Space J (2017). https://doi.org/10.1007/s12567-017-0170-8.

# ICATT 2016 STUDY

1. Calculating the Keplerian orbital elements

Eichhorn, H., Cano, J.L., McLean, F. et al. CEAS Space J (2017). https://doi.org/10.1007/s12567-017-0170-8.

# ICATT 2016 STUDY

1. Calculating the Keplerian orbital elements
2. Solving Kepler's equation

Eichhorn, H., Cano, J.L., McLean, F. et al. CEAS Space J (2017). https://doi.org/10.1007/s12567-017-0170-8.

# ICATT 2016 STUDY

1. Calculating the Keplerian orbital elements
2. Solving Kepler's equation
3. Solving Lambert's problem

Eichhorn, H., Cano, J.L., McLean, F. et al. CEAS Space J (2017). https://doi.org/10.1007/s12567-017-0170-8.

# ICATT 2016 STUDY

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

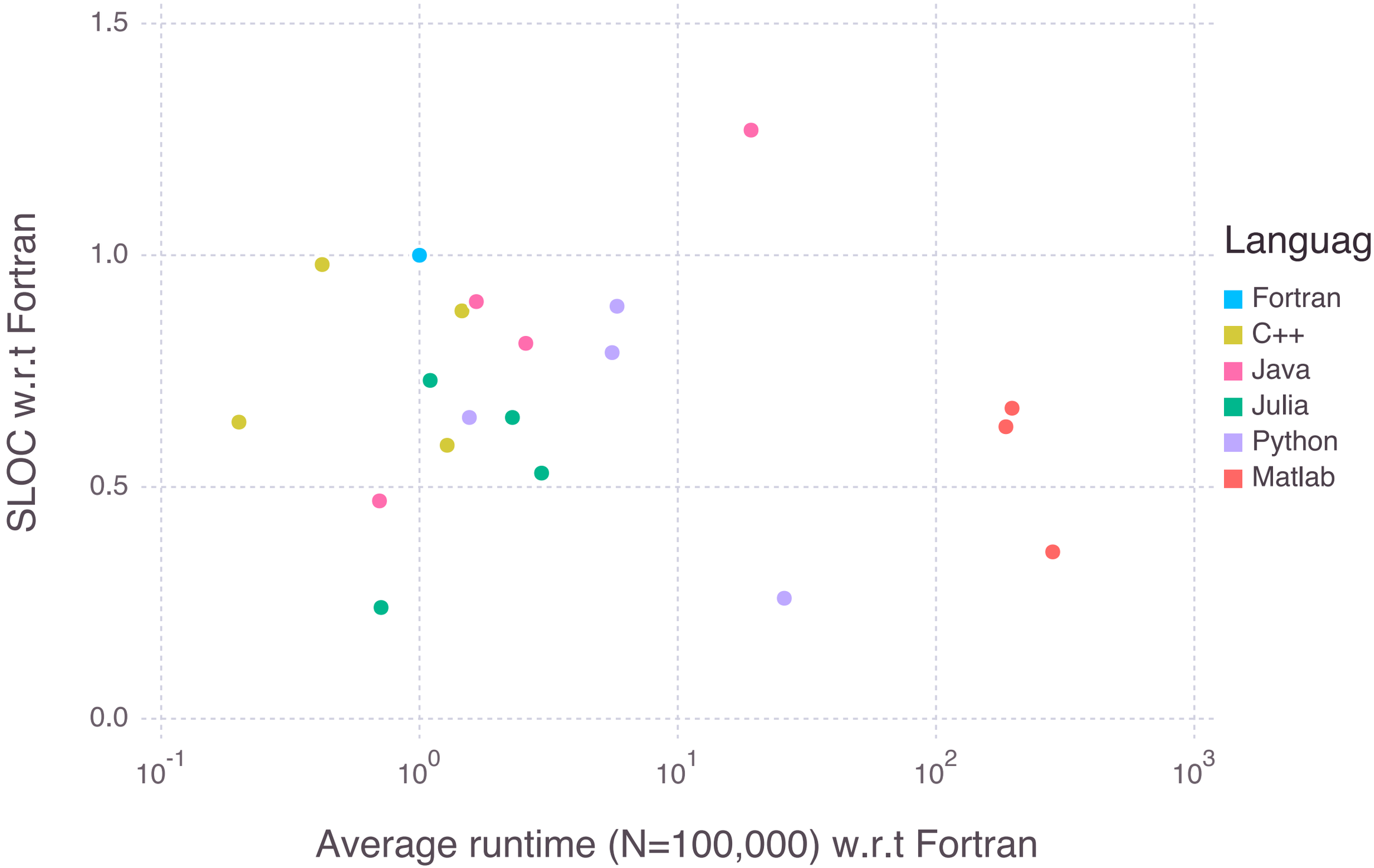
Eichhorn, H., Cano, J.L., McLean, F. et al. CEAS Space J (2017). https://doi.org/10.1007/s12567-017-0170-8.

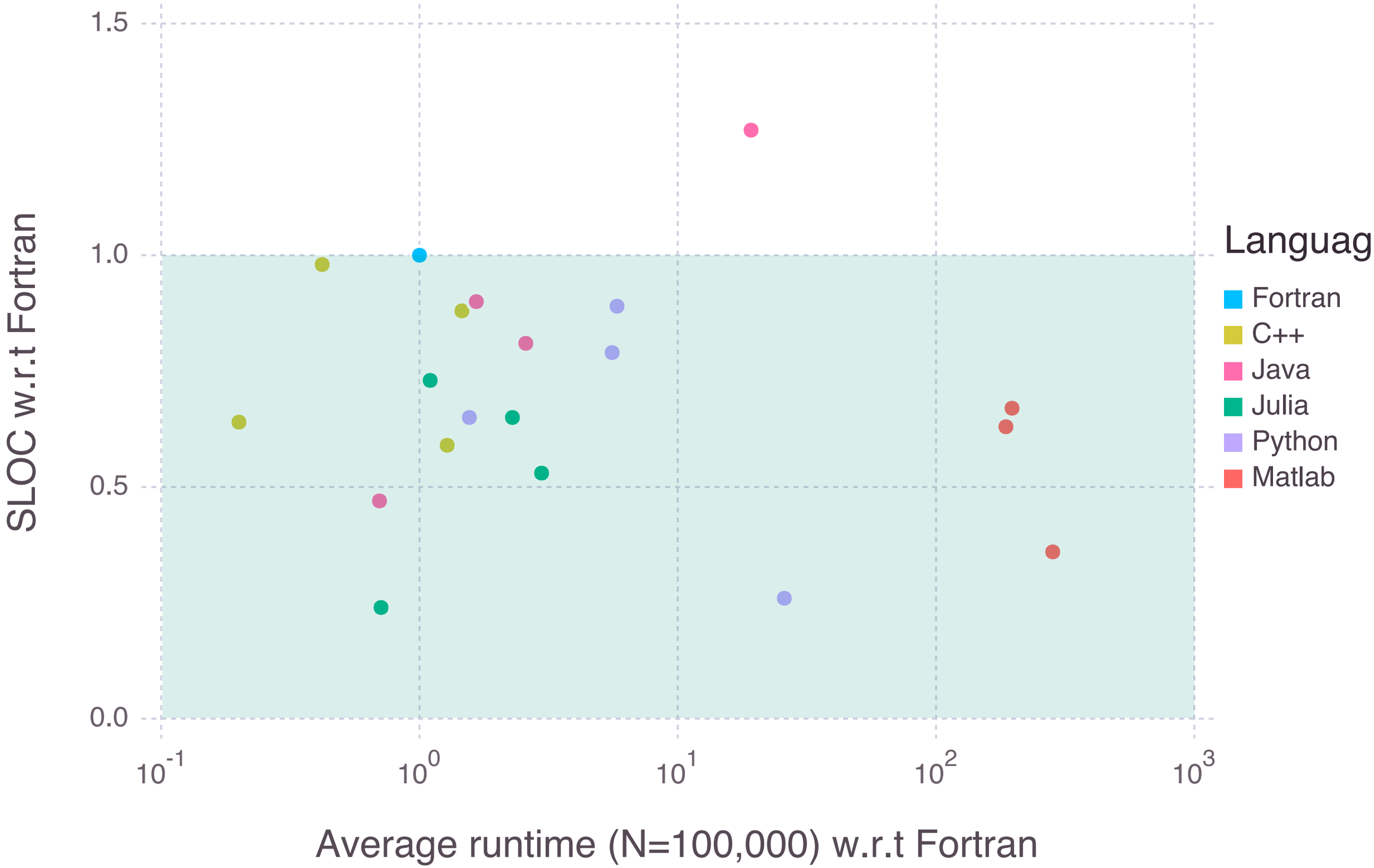
# ICATT 2016 STUDY

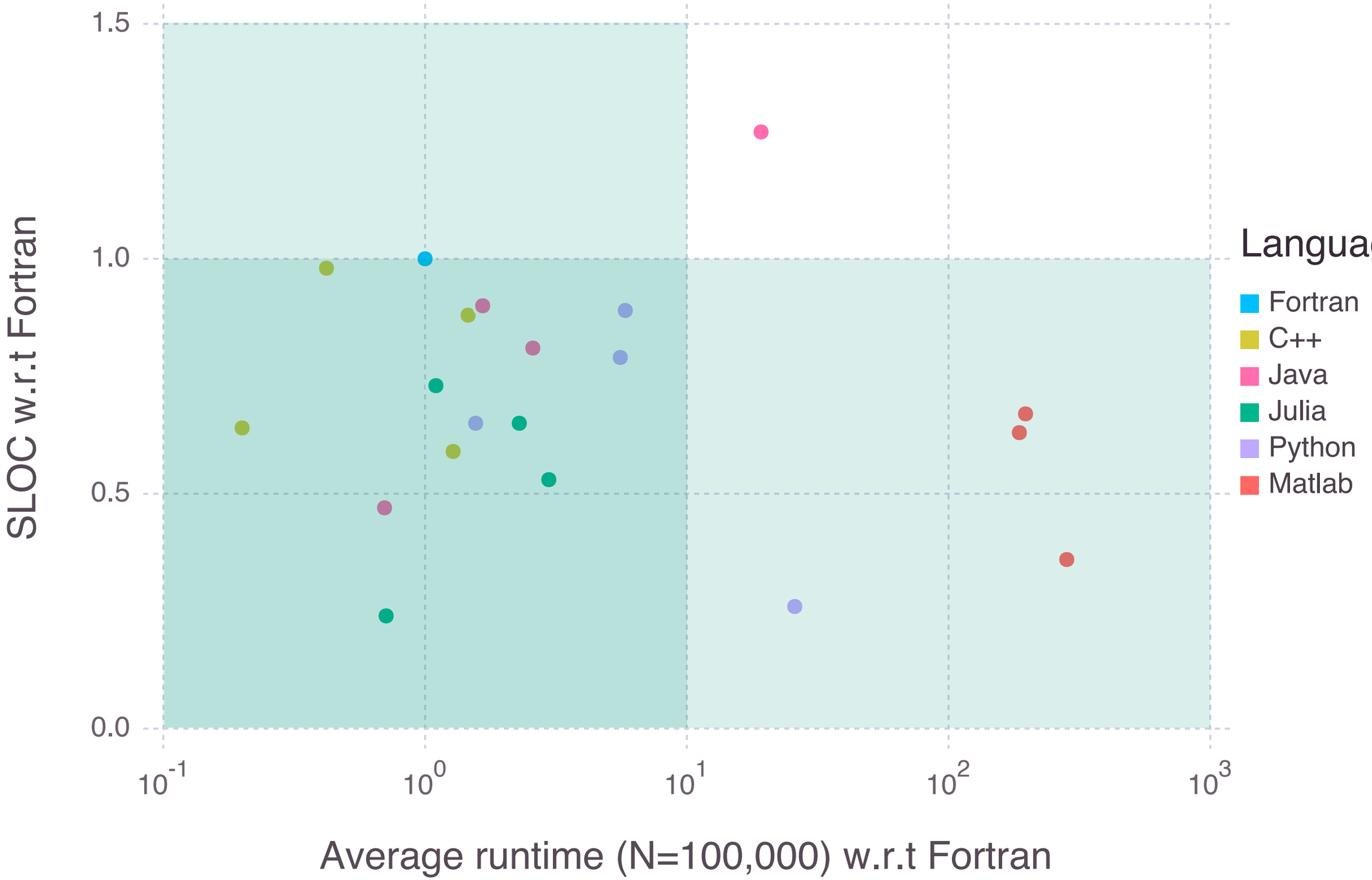
1. Calculating the Keplerian orbital elements
2. Solving Kepler's equation
3. Solving Lambert's problem
4. Calling the DOP853 Fortran 77 code

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

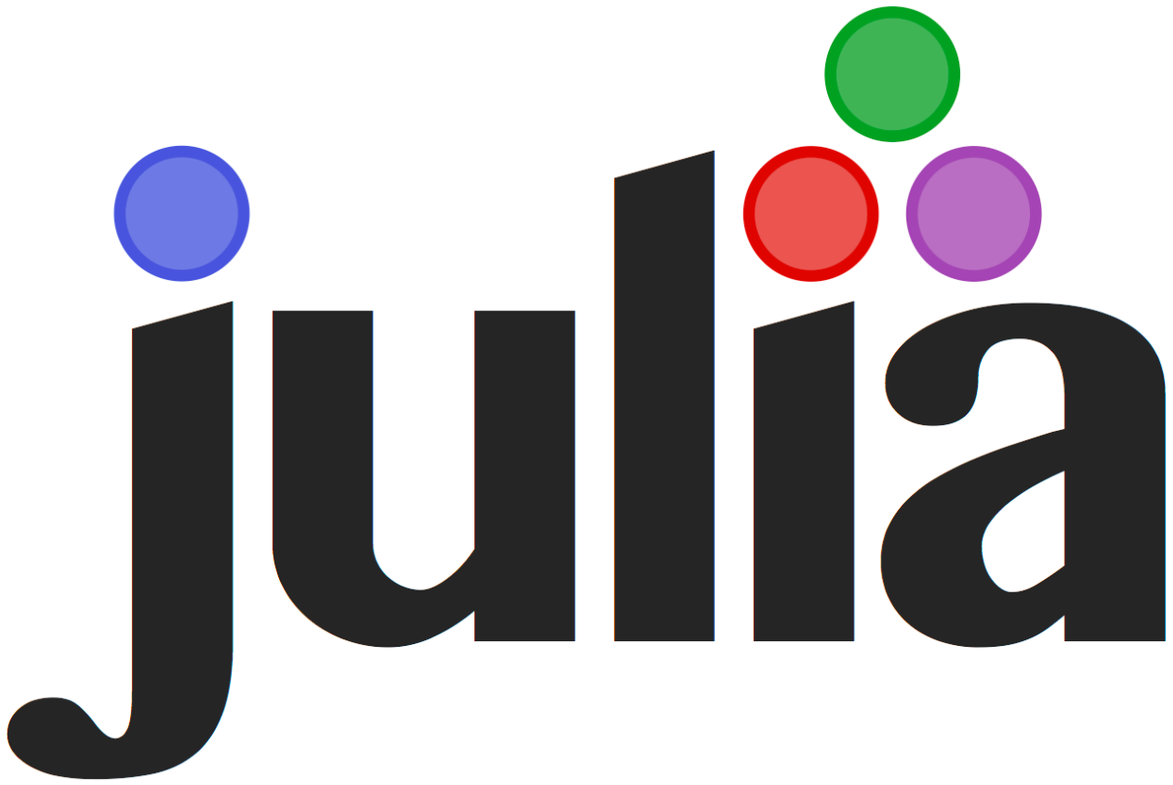
Eichhorn, H., Cano, J.L., McLean, F. et al. CEAS Space J (2017). https://doi.org/10.1007/s12567-017-0170-8.


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


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


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

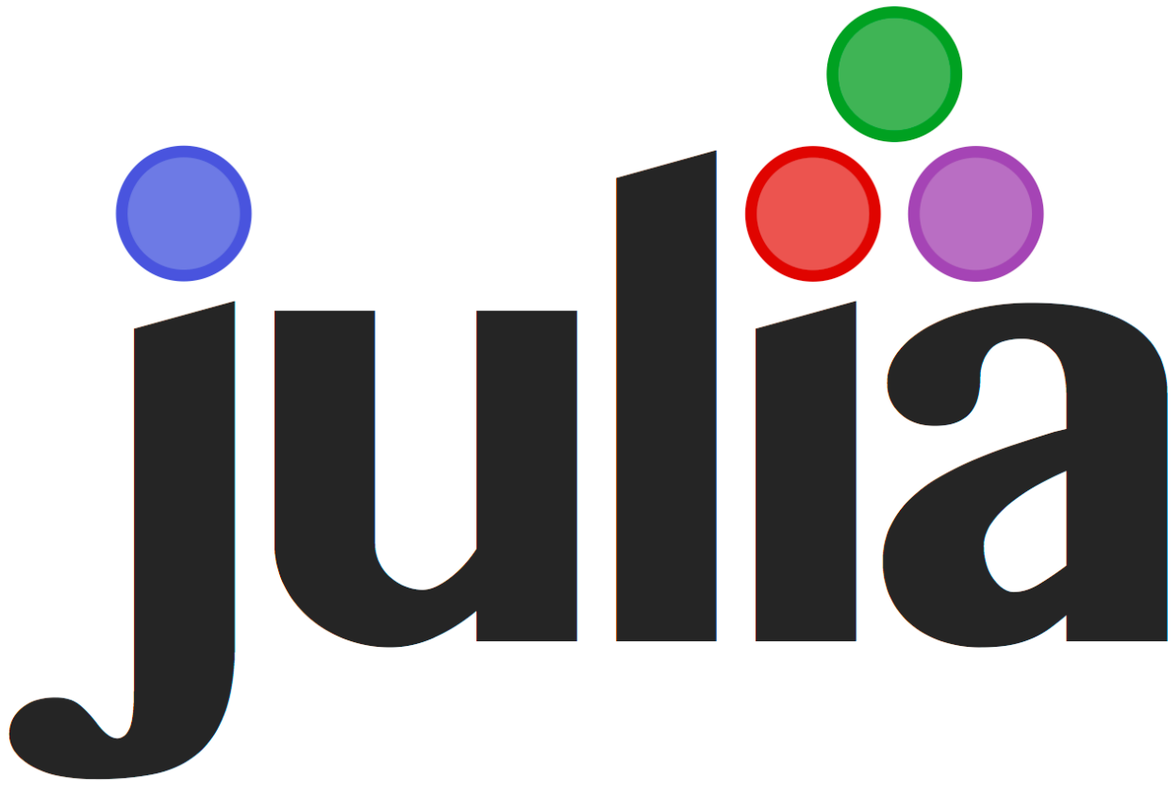



High-level, high-performance dynamic programming language

Initially developed at MIT now at Julia Computing, Inc.

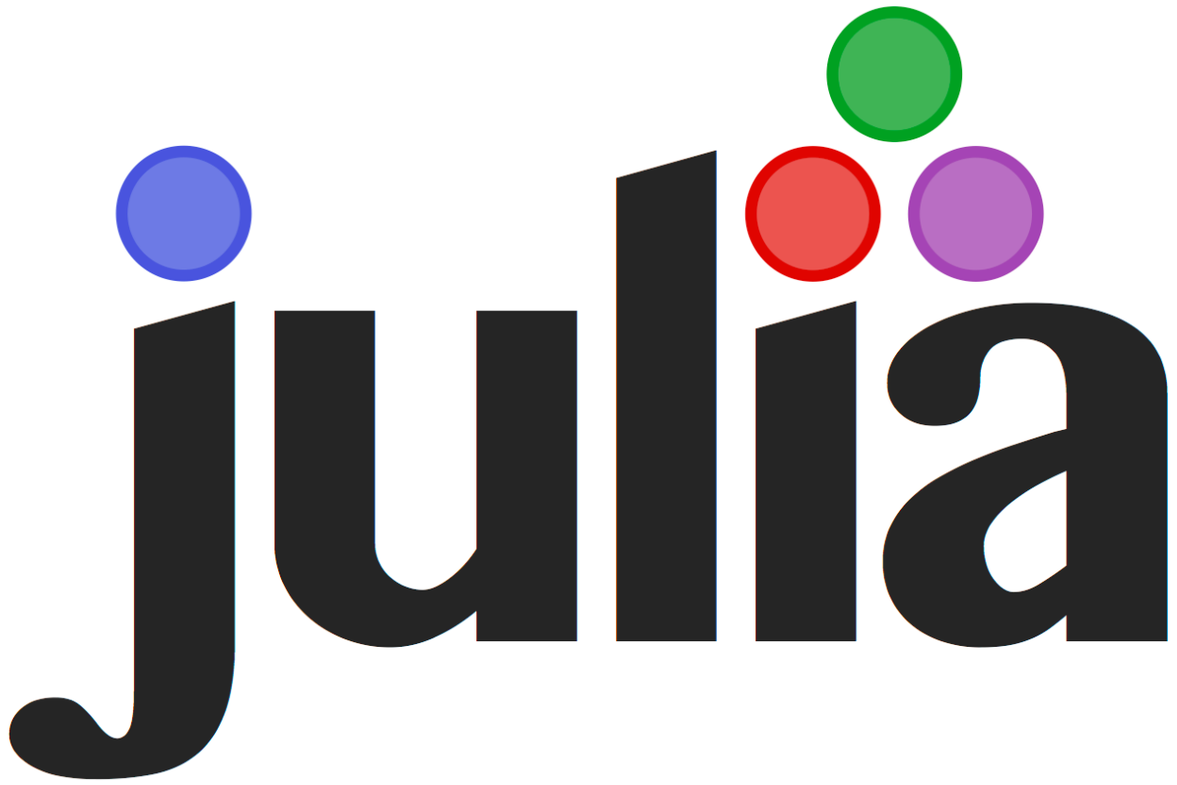
Comprehensive standard library and rapidly growing ecosystem

600+ individual contributors




High-level, high-performance dynamic programming language
Initially developed at MIT now at Julia Computing, Inc.
Comprehensive standard library and rapidly growing ecosystem
600+ individual contributors

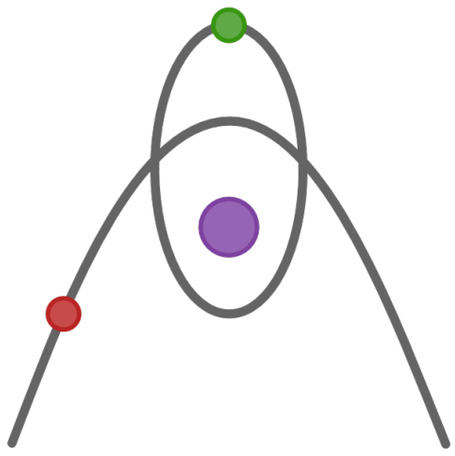
Looks like MATLAB with a heavy dose of Python but runs like Fortran




High-level, high-performance dynamic programming language
Initially developed at MIT now at Julia Computing, Inc.
Comprehensive standard library and rapidly growing ecosystem
600+ individual contributors

Looks like MATLAB with a heavy dose of Python but runs like Fortran

Current version: 0.6 → next will be 1.0 (Early 2018)



# Astrodynamics.jl

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

MPLv2-licensed framework for
astrodynamics applications

Proof-of-concept for my PhD thesis

Current Status: MVP

## Completed Features:

Time scale and reference frame conversions

High-performance ephemerides

Semi-analytical and numerical propagation with event detection

## Work in progress:

I/O (CCSDS, SPICE)

Trajectory optimization

Integrating third-party propagators (Orekit, GMAT)

**Validated with Orekit, GMAT, and SPICE**

# High performance

High performance

Extensible at runtime

High performance

Extensible at runtime

API for humans

High performance

Extensible at runtime

API for humans

Well documented

High performance

Extensible at runtime

API for humans

Well documented

Make STK obsolete

High performance

Extensible at runtime

API for humans

Well documented

Make STK obsolete

High performance

Extensible at runtime

API for humans

Well documented

Make STK obsolete

High performance

Extensible at runtime

API for humans

Well documented

Make STK obsolete

High performance

Extensible at runtime

API for humans

Well documented

Make STK obsolete

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?

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

THANK YOU VERY MUCH!

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