OSCW '17
Open Source Cubesat Workshop

Astrodynamics.jl: An Open-Source Framework for Interactive High-Performance Mission Analysis

Helge Eichhorn

Contact Author: Helge Eichhorn
Affiliation: Telespazio VEGA Deutschland GmbH
Address: Europaplatz 5, 64293 Darmstadt, Germany
Email: helge.eichhorn@telespazio-vega.de

Abstract

With Cubesat missions reaching beyond Low-Earth Orbit, the complexity of these missions increases and subsequently they require more complex mission analysis and flight dynamics software tools. These tools need to be easy to configure and extend to be able to handle complex space missions. They also need to offer high numerical performance and parallel computation capabilities to able to perform numerical propagations and optimizations in reasonable time.

While commercial off-the-shelf (COTS) astrodynamics solutions are able to handle a wide variety of mission scenarios, they force the user into vendor-specific workflows. Users need to learn proprietary scripting languages or work with constrained APIs. However, the major disadvantage of COTS mission analysis tools is that they are prohibitively expensive for many Cubesat projects.

Existing open source solutions like NASA's General Mission Analysis Tool (GMAT) are developed in statically compiled languages to fulfil the numerical performance requirements, which makes extending them difficult.

In the end, this often leads to a two-tiered approach to astrodynamics software development. The software is first prototyped in a dynamic language such as Matlab or Python and re-implemented in a high-performance language such as Fortran, C++ or Java. This in turn leads to duplication of effort and possible translation mistakes. Another approach is to port only computation-heavy parts to a compiled language and interface it with the dynamic parts of the software, e.g. Matlab MEX extensions. The problem here is that it greatly increases the complexity of the original software. This has been called the "Two Language Problem".

OSCW '17

Open Source Cubesat Workshop

Astrodynamics.jl

Astrodynamics.jl offers a fresh approach to computational astrodynamics and mission analysis applications. It is an open framework published under the Mozilla Public License Version 2 (MPLv2). The framework is written in the Julia programming language, which was originally developed at MIT and was recently spun out into the Julia Computing Inc. start-up. Julia is a dynamic language with syntax that takes inspiration from Matlab and Python. At the same time, it offers numerical performance that is competitive with C and Fortran through Just-In-Time (JIT) compilation. The main goal of Julia's developers has been to solve the aforementioned “Two Language Problem”.

Apart from the performance benefits and the expressive syntax, Astrodynamics.jl also leverages Julia's powerful user-extensible type system, its ability to seamlessly interface with other programming languages (including C/C++, Fortran, Java, and Python), built-in parallel computations, and its rapidly growing ecosystem of scientific software libraries, e.g. best-in-class ODE solvers.
Users can easily extend the framework with new force models or numerical solvers, even at runtime, without sacrificing performance.

If possible, I will give a live demo of the software and explain how Cubesat projects can leverage Astrodynamics.jl to get their mission off the ground.