IAF SPACE SYSTEMS SYMPOSIUM (D1)
Space Systems Engineering - Methods, Processes and Tools (2) (4B)

Author: Mr. Michele Ceresoli
Politecnico di Milano, Italy

Dr. Andrea Pasquale
Politecnico di Milano, Italy
Prof. Michèle Lavagna
Politecnico di Milano, Italy

MULTIVERSE: A NOVEL HIGH PERFORMANCE ASTRODYNAMIC TOOL IN JULIA

Abstract

Nowadays, the astrodynamic community is challenging itself to tackle problems of increasing complexity to foster the development of advanced algorithms and simulation models to enable the design and optimization of complex trajectories in the solar system. However, the non-analytical nature of these chaotic systems and the large solution space require algorithms and tools of ever-growing complexity. To tackle this problem, most tools are designed exploiting a two-language approach, which uses a dynamic and interactive language (e.g., MATLAB or Python) for the front end and a statically typed language (e.g., C++ or Fortran) on the back end for the performance-critical algorithms. Despite leading to better performances than a mere dynamic language, this design pattern introduces further complexities, mainly related to the extensibility and maintainability of the software.

A recent different approach to solve these issues has been addressed by Julia, a high-level dynamic language for high-performance numerical computing that exploits Just-In-Time (JIT) LLVM compilation to approach the speed of compiled C code. Designed around the multiple-dispatch paradigm, Julia's capabilities to automatically derive highly-optimized code for a wide range of applications provide a streamlined solution to reduce the time and effort required to develop and maintain astrodynamics tools.

This paper presents Multiverse.jl, a novel astrodynamic framework developed by PoliMi ASTRA team members to enable efficient and parallelizable trajectory design. Drafted as a full-featured Julia package, its core library offers ready-to-use and validated high-fidelity environment representations, high-precision time conversions, and key astronomy algorithms. By extensively using meta-programming and multiple-dispatch, Multiverse allows to automatically generate a simulation environment tailored to the user's needs with limitless combinations of ready-made and custom models. Among its features, the package also provides native and thread-safe access to binary ephemeris kernels, enabling Automatic Differentiation (AD) on all its components.

Starting from the package design principles, layout, and functionalities, the paper presents first extensive benchmarks against state-of-the-art libraries. Then, a comparison of the different libraries in terms of usability, maintainability, and extensibility is presented to highlight the relative advantages and disadvantages for early-stage designs. Finally, an example involving the orbit generation, optimization, and automatized transition to high-fidelity models is presented to demonstrate the practical advantages of the package both in terms of computational and development time efficiency.