

# THE JULIA ECOSYSTEM FOR ASTRONOMY, ASTROPHYSICS, AND GEOSPATIAL DATA SCIENCE IN 2025

Mike Kretlow<sup>1,2</sup>

<sup>1</sup>Deutsches Zentrum für Astrophysik (DZA), Görlitz, Germany.  
<sup>2</sup>Instituto de Astrofísica de Andalucía (IAA), Granada, Spain.

ADASS 2025  
Görlitz, Germany

## Abstract

The Julia ecosystem has grown rapidly in recent years. This poster offers a concise, personal overview of Julia’s ecosystem in 2025 for astronomy, astrophysics, and geospatial data science. It highlights key packages and tools (preferred but not exclusively in native Julia), without claiming to be complete.

## Introduction

Julia is a modern, high-level, dynamically typed programming language designed for high-performance numerical and scientific computing [1]. It combines the interactivity and ease of use of languages like Python or MATLAB with execution speeds approaching those of C/C++ and Fortran, enabled by its LLVM-based just-in-time (JIT) compilation and powerful multiple dispatch paradigm. A major strength of Julia lies in its interactive computing capabilities. It integrates seamlessly with Jupyter notebooks and other interactive environments (such as Pluto.jl and BonitoBook.jl), enabling efficient exploratory data analysis, algorithm development, visualization, and educational use. Julia ships with a powerful REPL (Read–Eval–Print–Loop) and a fully integrated environment system, built directly into its package manager (Pkg.jl).

Julia supports built-in multithreading and distributed computing, and GPU acceleration can be added via packages for high-performance computing.

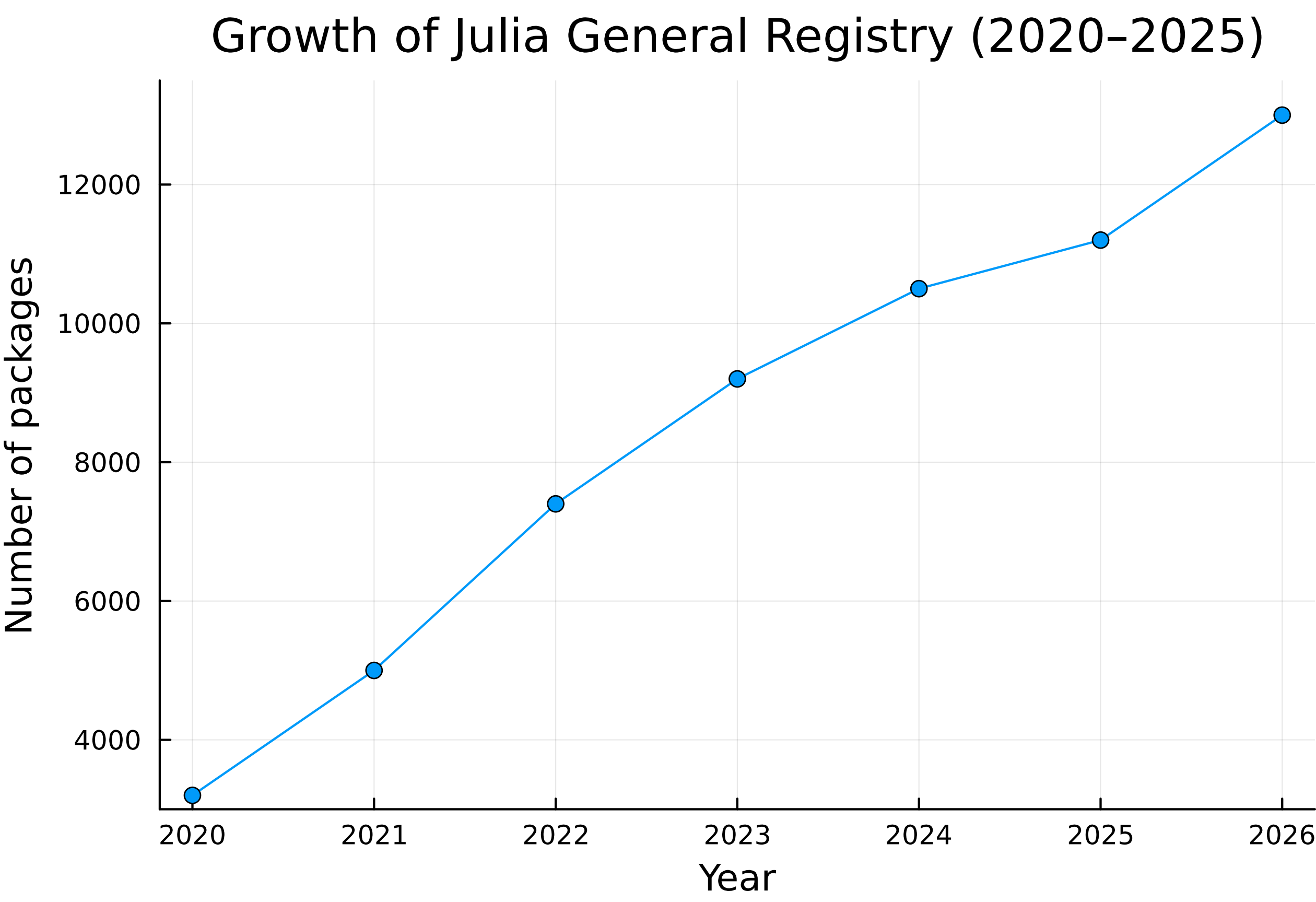
## Installation

The recommended way to install Julia is via juliaup (run "curl -fsSL https://install.julialang.org | sh" in your terminal), which installs the latest stable version, manages updates, and allows multiple versions. Alternatively, you can download binaries from the Julia website or compile from the GitHub repository (see "Resources").

## Resources

Website: [julialang.org](https://julialang.org), GitHub repo: [github.com/JuliaLang/julia](https://github.com/JuliaLang/julia), Packages: [juliapackages.com](https://juliapackages.com), [juliahub.com](https://juliahub.com), Community: [discourse.julialang.org](https://discourse.julialang.org), [julialang.slack.com](https://julialang.slack.com), [julialang.zulipchat.com](https://julialang.zulipchat.com),

## Growing Ecosystem



## References

[1] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B. Shah. Julia: A Fresh Approach to Numerical Computing. *SIAM Review*, 59(1):65–98, January 2017. ISSN 0036-1445. doi: 10.1137/141000671.

[2] Begnoni, Andrea and Anselmi, Stefano and Pieroni, Mauro and Renzi, Alessandro and Ricciardone, Angelo. Detectability and Parameter Estimation for Einstein Telescope Configurations with GWJulia. 6 2025. doi: 10.48550/arXiv.2506.21530.

[3] Jeffrey Regier, Kiran Pamnany, Keno Fischer, Andreas Noack, Maximilian Lam, and et al. Cataloging the visible universe through bayesian inference at petascale. 2018. doi: 10.48550/arXiv.1801.10277.

[4] Jonas Eschle, Tamás Gál, Mosè Giordano, Philippe Gras, Benedikt Hegner, and et al. Potential of the julia programming language for high energy physics computing. *Computing and Software for Big Science*, 7(1), October 2023. ISSN 2510-2044. doi: 10.1007/s41781-023-00104-x. URL <http://dx.doi.org/10.1007/s41781-023-00104-x>.

## Key Packages (Selection)

**Mathematics:** BoxLeastSquares.jl, DifferentialEquations.jl, Distributions.jl, Enzyme.jl, ForwardDiff.jl, JuMP.jl, LinearAlgebra.jl, Measurements.jl, NonlinearSolve.jl, Optim.jl, Symbolics.jl, Statistics.jl, Turing.jl, ...

**File I/O:** CSV.jl, JLD2.jl, JSON.jl, HDF5.jl, Parquet.jl, FileIO.jl, FortranFiles.jl, ...

**Tabular data:** DataFrames.jl, Tables.jl, CSV.jl, ...

**Databases:** DBInterface.jl, JuliaDB.jl, DuckDB.jl, SQLite.jl, ...

**GPU/HPC:** CUDA.jl, AMDGPU.jl, Metal.jl, Dagger.jl, ...

**Interactivity:** IJulia.jl, Pluto.jl, BonitoBook.jl

**Visualization:** Plots.jl (different backends), Makie.jl, ...

**(Sci) ML:** MLJ.jl, Lux.jl, ModelingToolkit.jl, ...

**Compiler:** JuliaC.jl (with Julia v1.12+ and --trim much smaller binaries)

## Astronomy & Astrophysics

**Constants & Units:** PhysicalConstants.jl, Unitful.jl, UnitfulAstro.jl, AstroAngles.jl

**Time & Coordinates:** AstroTime.jl, Astrometry.jl, ERFA.jl, EarthOrientation.jl, SkyCoords.jl, WCS.jl

**Astrodynamics:** GeneralAstrodynamics.jl, SatelliteToolbox.jl

**Ephemerides:** CALCEPH.jl, HORIZONS.jl, SPICE.jl, SolarPosition.jl

**Images:** CCDReduction.jl, LACosmic.jl, AstroImages.jl, FITSIO.jl, CFITSIO.jl, SAOImageDS9.jl

**Astrometry:** Astrometry.jl, WCS.jl, Astroalign.jl

**Photometry:** Photometry.jl, PhotometricFilters.jl, PSFModels.jl

**Time Series Analysis:** TimeSeries.jl, LombScargle.jl

**Spectroscopy:** Spectra.jl, SpectralFitting.jl

**Cosmology:** Cosmology.jl, DustExtinction.jl

**Exoplanets:** Transits.jl, Orbits.jl

**GR/GW:** GWJulia.jl/GWInference.jl [2]

**VO:** VirtualObservatory.jl, VOTables.jl

**Utilities:** AstroLib.jl, SAOImageDS9.jl, XPA.jl

**Resources:** [juliaastro.org](https://juliaastro.org) (JuliaAstro)

## Geospatial

**Data & Modeling:** GeoStats.jl, GDAL.jl, ArchGDAL.jl, NetCDF.jl, NCDatasets.jl, GeoDataFrames.jl, GeoJSON.jl, Shapefile.jl, ...

**Coordinates:** CoordinateTransformations.jl, Geodesy.jl, Proj.jl, ...

**Maps/Plotting:** GMT.jl, GeoMakie.jl, OpenStreetMap.jl, ...

**Resources:** [juliageo.org](https://juliageo.org) (JuliaGeo)

## Large Scale Projects

Julia has shown excellent performance in large-scale scientific computing. The Celeste.jl project processed 178 TB of astronomical data on 8192 nodes of the Cori supercomputer, achieving 1.54 PFLOP/s [3]. A CERN study found that Julia fulfills nearly all requirements for high-energy physics (HEP) applications, offering ease of development with performance matching or exceeding C++ and far surpassing Python in event-loop throughput [4].

Contact information:  
Mike Kretlow, DZA-ZTE  
[mike.kretlow@dzaastro.de](mailto:mike.kretlow@dzaastro.de)