

On modelling optimization problems via Julia JuMP

Prof. Mayron César de O. Moreira

Department of Computer Science

Universidade Federal de Lavras (UFLA)

May 15 - May 17, 2019

Reggio Emilia, UNIMORE, Italy

Outlines

- Lecture 1: an overview of Julia Language
 - Simple examples, Handling input and output data;
 - JuMP: commands, examples, modeling classical problems: knapsack, transportation, TSP, exercises;
- Lecture 2: Assembly Line Balancing Problems (ALBPs)
 - SALBP-1 and SALBP-2;
 - ALWABP;
 - Exercises.
- Lecture 3: Vehicle Routing Problem (VRP) and Production Planning
 - Modeling a variant of VRP;
 - Modeling a production planning problem;
 - Exercise.

Outlines

- Lecture 4: Problems on Graphs
 - Maximum Dispersion Problem (MaxDP);
 - Promoting Eulerian Districts for Arc Routing;
 - Exercise.
- Lecture 5: Final test

Universidade Federal de Lavras (UFLA)

- TOP 10 in the Brazilian Ranking of Courses (2018);
- Approximately 800 professors and 16,000 students;
- 35 undergraduate courses, and 43 graduate programs;
- Prize “**Blue University**”: Reference on *Water Sustainability*;
- The incentive to entrepreneurship: innovation hubs and, technological park (*LavrasTec*).


Universidade Federal de Lavras (UFLA)



Universidade Federal de Lavras (UFLA)


Follow us:

www.instagram.com/uflabr

 | Instagram

Busca

Entrar Cadastre-se



uflabr


Seguir


1.312 publicações 26,3mil seguidores 154 seguindo


Universidade Federal de Lavras

Perfil oficial da Universidade Federal de Lavras. Use as hashtags #OlharesUFLA e #UFLA, e marque @uflabr nas fotos tiradas no câmpus!

ufla.br


Mamute

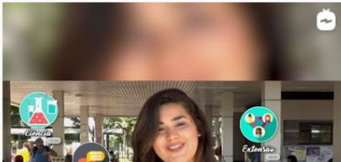

Divulgação

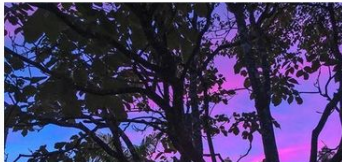

❤️


PUBLICAÇÕES

IGTV

MARCA DO







Department of Computer Science

- **40 professors (Ph.D.)** and 8 administrative technicians;
- 2 **undergraduate** courses: **Information Systems** and **Computer Science**;
- **MSc. Computer Science** (13 advisors): Software Engineering, Computational Intelligence, **Operations Research**, and Computer Networks;
- Teaching and Research Infrastructure.

Department of Computer Science



L-LOOP: Lavras Laboratory of Optimization

- 4 professors;
- 10 undergraduate students and 5 MSc. students;
- 4 projects funded by CNPq and FAPEMIG ;
- Expertise: *Theory and Practice of Operations Research.*



Lavras
Lab of Optimization

L-LOOP: since 2016

A short bio: Moreira, M. C. O.

- Professor at UFLA since 2015;
- Ph.D. at Universidade de São Paulo (USP);
 - Research stages: UPV (Valencia, Spain), UPC (Barcelona, Spain) and CIRRELT (Montreal, Canada);
- *Funded by:* CNPq, FAPEMIG and UFLA.

Collaborators:



“We want a language that’s **open source**, (...) We want the **speed of C** with the **dynamism of Ruby**. (...) familiar **mathematical notation** like Matlab. (...) **general programming** as Python, as easy for **statistics** as R, as **natural** for string processing as Perl, (...). We want it **interactive** and we want it **compiled**.”

Why Julia?



Jeff Bezanson
JeffBezanson



Viral B. Shah
ViralBShah



Alan
Edelman

Professor of Applied Mathematics
Computer Science and AI Laboratories
Applied Computing Group Leader

» Stefan Karpinski «



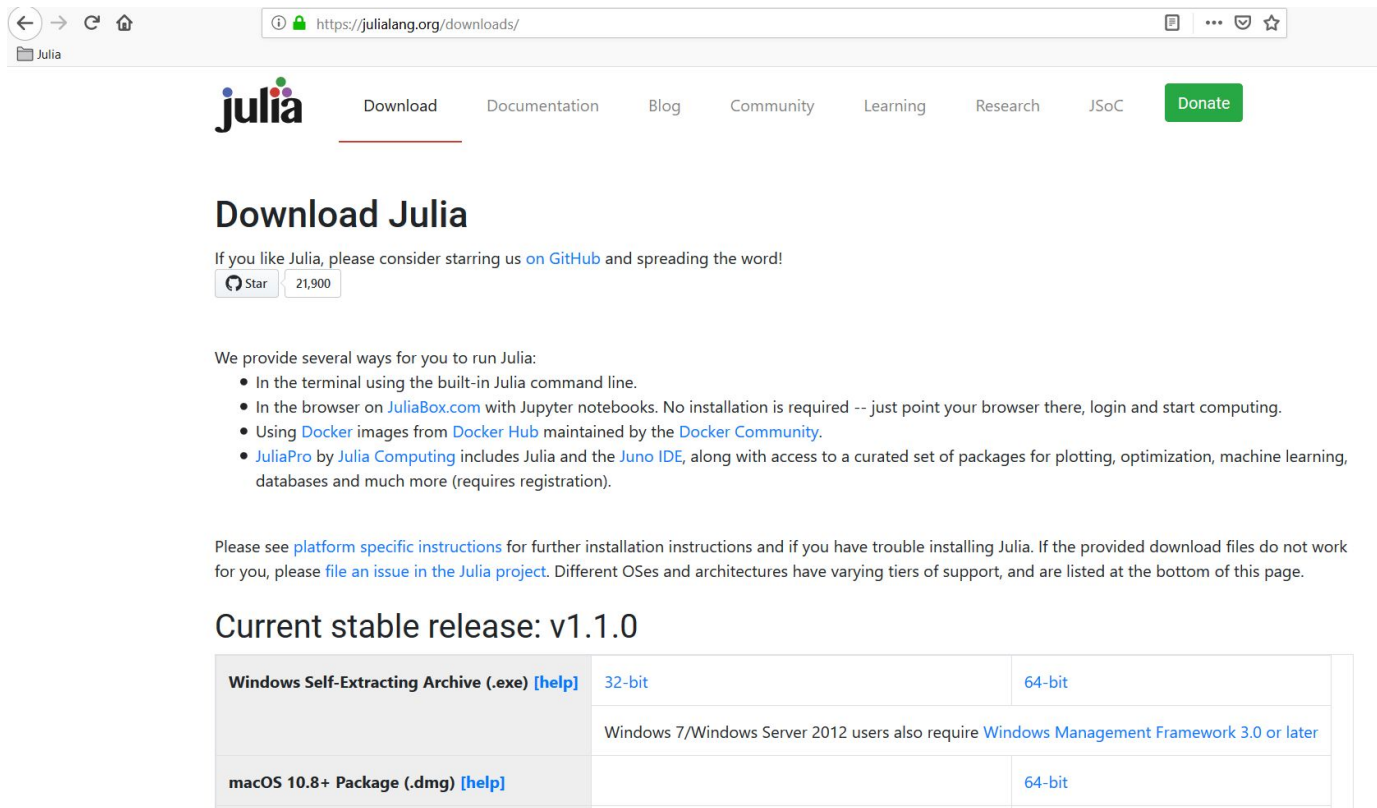
Why Julia?

- **High Performance** language;
- Rich language of **descriptive datatypes**;
- **Free** and **open source**;
- Just-in-time compilation;
- Julia has: **garbage collector, efficient libraries** for **floating-point calculations**, linear algebra, random number generation, and **regular expression matching**.

Why Julia?

Download and install (Windows, MacOS and Linux)

Remember to set up path variable.



The screenshot shows the JuliaLang website's download page. The browser address bar displays 'https://julialang.org/downloads/'. The page features a navigation bar with links for 'Download', 'Documentation', 'Blog', 'Community', 'Learning', 'Research', 'JSoC', and a green 'Donate' button. The main heading is 'Download Julia'. Below it, a message encourages starring the project on GitHub, accompanied by a GitHub star button showing 21,900 stars. A list of installation methods is provided: terminal command line, JuliaBox.com, Docker images, and JuliaPro. A note mentions that JuliaPro includes the Juno IDE and access to a curated set of packages. Further down, instructions refer to platform-specific guides and a link to file issues. At the bottom, the current stable release is v1.1.0, followed by a table of download links for Windows and macOS.

Download Julia

If you like Julia, please consider starring us [on GitHub](#) and spreading the word!

Star 21,900

We provide several ways for you to run Julia:

- In the terminal using the built-in Julia command line.
- In the browser on [JuliaBox.com](#) with Jupyter notebooks. No installation is required -- just point your browser there, login and start computing.
- Using [Docker](#) images from [Docker Hub](#) maintained by the [Docker Community](#).
- [JuliaPro](#) by [Julia Computing](#) includes Julia and the [Juno IDE](#), along with access to a curated set of packages for plotting, optimization, machine learning, databases and much more (requires registration).

Please see [platform specific instructions](#) for further installation instructions and if you have trouble installing Julia. If the provided download files do not work for you, please [file an issue in the Julia project](#). Different OSes and architectures have varying tiers of support, and are listed at the bottom of this page.

Current stable release: v1.1.0

Windows Self-Extracting Archive (.exe) [help]	32-bit	64-bit
	Windows 7/Windows Server 2012 users also require Windows Management Framework 3.0 or later	
macOS 10.8+ Package (.dmg) [help]		64-bit

Why Julia?

- Currently: Version 1.1.0;
- Other Resources
 - *Learning* (tutorials, texts and books):
 - <https://julialang.org/learning/>
 - *Coursera*: www.coursera.org/learn/julia-programming
 - *Official channel*: <https://julialang.org/community/>
 - *Documentation*: <https://docs.julialang.org/en/v1/>
- *Learn by examples* (version 0.6.2):
<https://juliabyexample.helpmanual.io/>
- IDE: [Juno](#).

Why Julia?

The image displays the Juno IDE interface, which is a flexible IDE for the 21st century. The interface is divided into several panels:

- Project Panel:** Located on the left, it shows a list of projects including ASTInterpreter2, atom-ink, atom-julia-client, atom-indent-detective, DiffEqUncertainty, DiffEqDevTools, Gadfly, Atom, junowebiste, and PlotThemes.
- Code Editor:** The central area shows a Julia script named "untitled". The script defines a function `profile_test(n)` that generates random data, performs FFT, and sorts it. Below the script, the REPL shows the execution of `profile_test(1)` and `@profile profile_test(10)`, followed by a series of REPL commands like `dct(A [, dims])`, `dct!(A [, dims])`, `do`, `div(x, y)`, `dec`, `done`, `diff(A::AbstractVector)`, `dump(x; maxdepth=8)`, and `detach(command)`.
- Workspace Panel:** Located on the right, it shows the current workspace with variables `a` (Float64[5]), `ans` (3.75...), and `profile_test` (profile_test).
- Plots Panel:** At the bottom right, it displays a line plot with three data series (y1, y2, y3) over a range of values. The y-axis ranges from 0.00 to 1.00.

Why Julia?

- *Some ecosystems*
 - **Bioinformatics and Computational Biology in Julia**
 - **BioJulia** - <https://github.com/BioJulia>
 - **Optimization packages**
 - **JuliaOpt** - <https://www.juliaopt.org/>
 - Image Processing
 - **JuliaImages** - <https://juliaimages.org/latest/>
 - Quantum Systems
 - **QuantumBFS** - <https://github.com/QuantumBFS>
 - **QuantumOptics** - <https://github.com/qojulia/QuantumOptics.jl>
 - Complex Systems
 - **JuliaDynamics** - <https://github.com/JuliaDynamics>

Why Julia?

- *Some ecosystems*
 - Quantitative economic modeling
 - QuantEcon - <https://github.com/QuantEcon>
 - Astronomy and Astrophysics
 - Julia Astro - <http://juliaastro.github.io/>
 - Ecology
 - EcoJulia - <https://github.com/EcoJulia>
 - **Machine Learning and Artificial Intelligence**
 - **Flux** - <https://juliacomputing.com/domains/ml-and-ai.html>
 - Statistics
 - JuliaStats - <https://github.com/JuliaStats>

Why Julia?


JuliaBox: run
in your
browser!

The screenshot shows the JuliaBox website interface. At the top, there's a navigation bar with the Julia logo and the text "Run Julia in your Browser". Below this, there are four login buttons: "Log in with Email", "Log in with GitHub", "Log in with Google", and "Log in with LinkedIn". The main content area features eight icons representing various features: Jupyter Notebook Interface, 75,000+ users served since 2015, Free registration, free to use, Perfect for classes, students, professors and new Julia users, Includes 275+ carefully curated popular Julia packages, Multi-node deployment capability, Parallel computing capability, and Buy added memory, storage, nodes and enterprise support. At the bottom, there's a table titled "Subscription Models (Login to purchase)" with three columns: Free, Silver, and Gold. The table lists the price for each model: Free (\$0/month), Silver (\$14/month), and Gold (\$70/month).

Subscription Models (Login to purchase)	Free	Silver	Gold
Price	\$0/month	\$14/month	\$70/month

Why Julia?

Julia Conference!



A screenshot of the JuliaCon 2019 website. The browser address bar shows <https://juliacon.org/2019/>. A yellow banner at the top states "Call for Proposals now closed | Early Bird Ticket Sale ends 2019-05-10 11:59 EDT". A green banner below it says "Request a Mentor for your talk by May 26!". The main header features the "juliacon 2019" logo and a link to "PREVIOUS EDITIONS / 2018 / 2017 / 2016 / 2015 / 2014". The central text reads "JuliaCon is coming to Baltimore" in large purple and blue letters, followed by "Monday 22nd to Friday 26th of July, 2019 at the University of Maryland Baltimore (UMB), Baltimore, MD, USA" in blue. The bottom of the page is decorated with a row of stylized black and red icons representing various Baltimore landmarks, including skyscrapers, a church, a lighthouse, and the University of Maryland building.

← → ↻ 🏠 <https://juliacon.org/2019/> 📄 ⋮ 📧 ☆ ⚡ 📖 📱

Julia

Call for Proposals now closed | Early Bird Ticket Sale ends 2019-05-10 11:59 EDT

Request a Mentor for your talk by May 26!

juliacon 2019

PREVIOUS EDITIONS / 2018 / 2017 / 2016 / 2015 / 2014

JuliaCon is coming to Baltimore

Monday 22nd to Friday 26th of July, 2019
at the University of Maryland Baltimore (UMB),
Baltimore, MD, USA

Why JuMP?

README.md



powered by NumFOCUS

JuMP is a domain-specific modeling language for [mathematical optimization](#) embedded in [Julia](#). It currently supports a number of open-source and commercial solvers ([Artelys Knitro](#), [BARON](#), [Bonmin](#), [Cbc](#), [Clp](#), [Couenne](#), [CPLEX](#), [ECOS](#), [FICO Xpress](#), [GLPK](#), [Gurobi](#), [Ipopt](#), [MOSEK](#), [NLOpt](#), [SCS](#)) for a variety of problem classes, including [linear programming](#), [\(mixed\) integer programming](#), [second-order conic programming](#), [semidefinite programming](#), and [nonlinear programming](#).

Why JuMP?

- Syntax close to mathematical notation!
- *“easy to specify and solve optimization problems without expert knowledge”* (<https://github.com/JuliaOpt/JuMP.jl>);
- **Online examples:**
 - <https://github.com/JuliaOpt/JuMP.jl/tree/master/examples>
 - <https://github.com/JuliaOpt/juliaopt-notebooks/blob/master/notebooks/Shuvomoy%20-%20Column%20generation.ipynb>

Let's practice!

We will use: Julia 1.1.0, JuMP 0.19.0, Cbc (solver), and Jupyter.

Go to:

[https://github.com/mayronmoreira/
ufla-unimore-julia-jump](https://github.com/mayronmoreira/ufla-unimore-julia-jump)

Other References

- Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM review*, 59(1), 65-98.
- Bezanson, J., Karpinski, S., Shah, V. B., & Edelman, A. (2012). Julia: A fast dynamic language for technical computing. *arXiv preprint arXiv:1209.5145*.
- Changhyun Kwon (2019). [Julia Programming for Operations Research](#), 2nd Edition.
- Lubin, M., & Dunning, I. (2015). Computing in operations research using Julia. *INFORMS Journal on Computing*, 27(2), 238-248.

On modelling optimization problems via *Julia JuMP*

Prof. Mayron César de O. Moreira

mayron.moreira@ufla.br

<http://professores.dcc.ufla.br/~mayron/>