JuMP: A Julia-based modeling language for mathematical optimization

Juan Pablo Vielma

Massachusetts Institute of Technology

Columbia Business School

New York, NY, October, 2016.

Audience poll

- AMPL
- GAMS
- AIMMS
- OPL
- CVX
- YALMIP
- PuLP
- Pyomo
- Gurobi Python API
- JuMP (or Julia)

What is Julia?

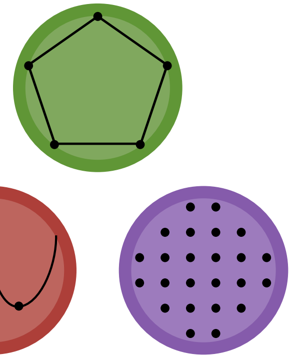
julia

- 21st century programming language
- MIT licensed: free and open source
- (Almost) as fast as C and as easy as Matlab
- http://julialang.org

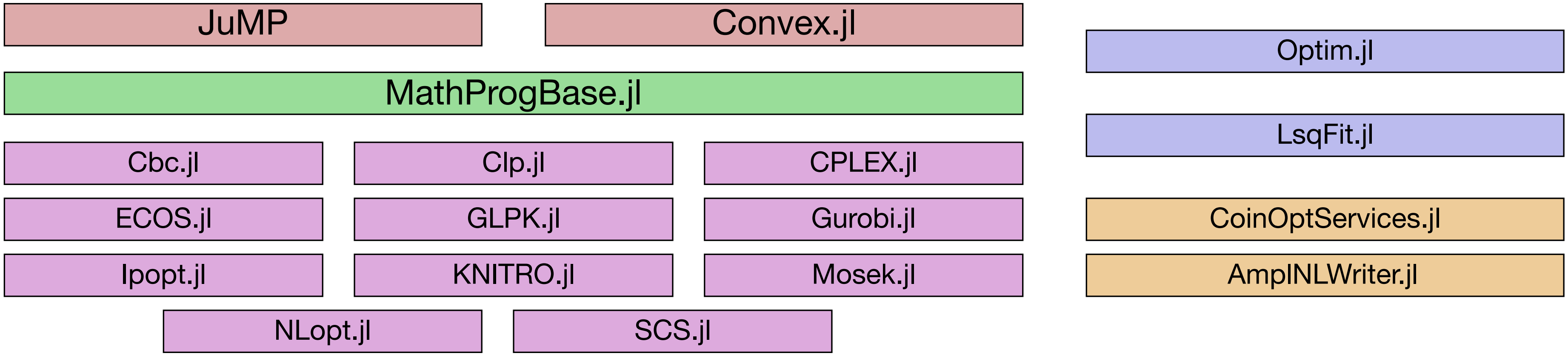
Why choose Julia for optimization?

- “I want to model and solve a large LP/MIP within a programming language, but Python is too slow and C++ is too low level”
- “I want to implement optimization algorithms in a fast, high-level language designed for numerical computing”
- “I want to create an end-user-friendly interface for optimization without writing MEX files”
- Lubin and Dunning, “Computing in Operations Research Using Julia”, INFORMS Journal on Computing (27), 2015, pp. 238–248

Optimization and Julia



JuliaOpt



JuMP

Convex.jl

MathProgBase.jl

Cbc.jl

Clp.jl

CPLEX.jl

ECOS.jl

GLPK.jl

Gurobi.jl

Ipopt.jl

KNITRO.jl

Mosek.jl

NLopt.jl

SCS.jl

Optim.jl

LsqFit.jl

CoinOptServices.jl

AmplNLWriter.jl

http://www.juliaopt.org

Why JuMP?

- Speed: Model generation should not be a bottleneck
- Programmability
  - Extensibility (JuMPeR, JuMPChance, MultiJuMP, JuGP, etc)
  - Embeddability (e.g. StochDynamicProgramming.jl)
- Solver independence
- Advanced features
  - Branch & bound callbacks
  - Automatic differentiation + user-defined functions
- Interactivity and visualization
  - Requires speed and programmability
- Dunning, Huchette and Lubin, “JuMP: A Modeling Language for Mathematical Optimization”, 2015, arXiv:1508.01982