Optimal Control in Julia with JuMP and InfiniteOpt

Presenter: Joshua Pulsipher (University of Waterloo)

Summary

This workshop will be an interactive tutorial on how to model complex nonlinear, continuous-time optimal control problems via InfiniteOpt.jl and JuMP.jl. Leveraging a unifying abstraction for infinite-dimensional optimization (InfiniteOpt) problems, InfiniteOpt.jl is a Julia-based open-source software package that builds upon JuMP.jl to provide an intuitive symbolic modeling environment for many problem classes in dynamic, PDE-constrained, and stochastic optimization. Moreover, its extensibility allows researchers to make their cutting-edge techniques accessible to a wide audience of individuals. All these aspects make InfiniteOpt.jl a powerful tool for tackling advanced optimal control problems.

Intended Audience

This workshop I intended for researchers (both academic and industrial) and practitioners interested in solving optimal control problems. The workshop will be accessible for those with limited experience modeling optimization problems in Julia and will feature a collection of interactive Jupyter worksheets to actively engage those attending. Researchers with an advanced knowledge of JuMP.jl will also benefit from learning how to use InfiniteOpt.jl to deploy their research methods.

Tentative Schedule

The core modules of this workshop will include:

- Background/motivation
    - Why should I use InfiniteOpt.jl?
    - Overview of its unifying modeling abstraction for InfiniteOpt problems
    - Brief review of optimal control problems
- Modeling with JuMP.jl
    - Overview the capabilities of JuMP.jl
    - Model and solve a quadcopter control example
- Modeling with InfiniteOpt.jl
    - Walkthrough the high-level symbolic modeling syntax (via a quadcopter example)
    - Solve it and interrogate the results
    - Show how we can quickly change the solution techniques and re-solve
- Novel formulations in InfiniteOpt.jl
    - Implement novel objective formulations for a pandemic control problem with uncertainty
    - Observe how the solution changes with different objectives
- Deploying new techniques with InfiniteOpt.jl
    - Highlight API available for researchers to implement their advanced techniques
    - Use the previous novel objective example to motivate this

The tentative schedule for covering these modules is as follows:

Times	Topic	Duration
9:00 a.m. – 9:30 a.m.	*Introduction* - The why and what of Julia, JuMP.jl, and InfiniteOpt.jl.	30 min.
9:30 a.m. – 10:00 a.m.	*Installation and Setup* - Configure software on personal laptop. Online interface provided as an alternative.	30 min.
10:00 a.m. – 10:10 a.m.	Break	10 min.
10:10 a.m. – 11:10 a.m.	*Julia: A Practical Introduction* – Overview of core types, programmatic syntax, and package management.	60 min.
11:10 a.m. – 11:20 a.m.	Break	10 min.
11:20 a.m. – 12:00 p.m.	*JuMP.jl: A Brief Introduction* – The basics of modeling and solving mathematical optimization problems in JuMP.jl.	40 min.
12:00 p.m. – 1:00 p.m.	Lunch	60 min.
1:00 p.m. – 2:00 p.m.	*JuMP.jl: Beyond the Basics* – A deeper dive into the core modeling/solution strategies including variables, constraints, containers, and more.	60 min.
2:00 p.m. – 2:10 p.m.	Break	10 min.
2:10 p.m. – 3:20 p.m.	*InfiniteOpt.jl: The Basics* – An introduction on how to compactly model and solve complex infinite-dimensional optimization problems.	70 min.
3:20 p.m. – 3:30 p.m.	Break	10 min.
3:30 p.m. – 4:15 p.m.	*InfiniteOpt.jl: New Modeling Strategies* – A tutorial on how InfiniteOpt.jl enables new formulation/solution approaches.	45 min.
4:15 p.m. – 4:45 p.m.	*InfiniteOpt.jl: Deployment Tools* – An overview of the API to enable rapid deployment of new modeling/solution techniques.	30 min.
4:45 p.m. – 5:00 p.m.	*Final Thoughts* – Summary of key points and panned future development. Provide resources for further learning.	15 min.