LyoPronto v2:

Code Verification, Equipment Capability Modeling, and More

Isaac Wheeler, Vivek Narsimhan, Alina Alexeenko* alexeenk@purdue.edu
Purdue University, Davidson School of Chemical Engineering

Introduction

- LyoPRONTO first published in 2019 [1]
  - Python script/package
  - Web interface (written in JavaScript) hosted by Purdue
    `https://lyopronto.geddes.rcac.purdue.edu`
- Now:
  - Python package hosted on GitHub

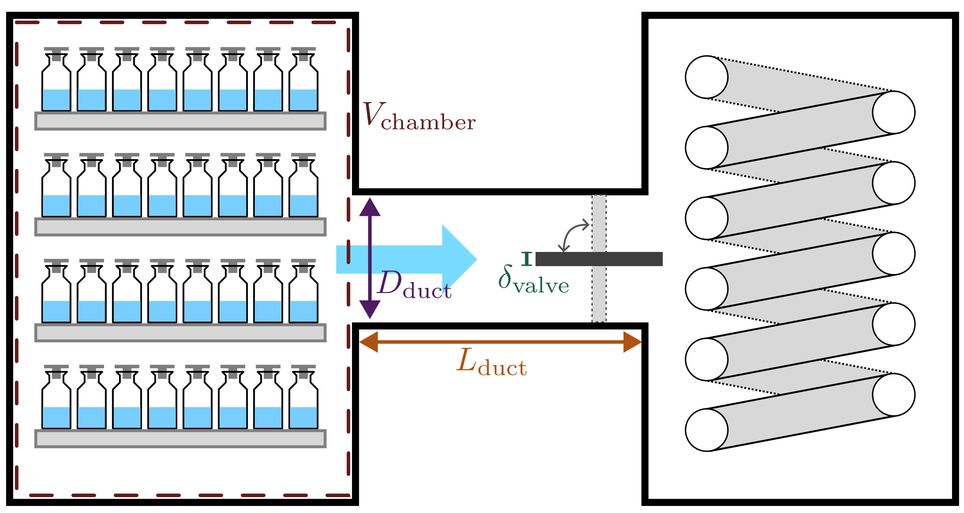
```
pip install git+https://github.com/LyoHUB/LyoPRONTO
```

  - Julia package `LyoPronto.jl` has additional tools
  - Web interface written in Python, hosted at Purdue but with a Docker container that can deploy anywhere

`https://lyopronto2.geddes.rcac.purdue.edu`

Equipment Capability Compact Model

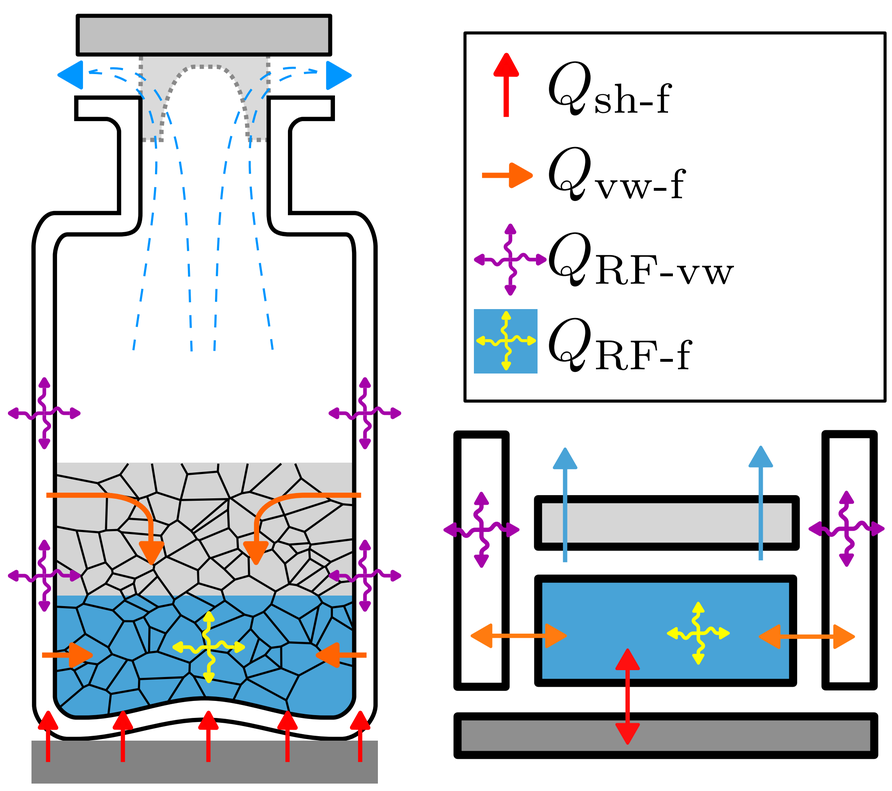
- Compact model: interpolates across CFD results validated with experiments [2]
- Provided now in web interface, implemented in `LyoPronto.jl`



- With V_{chamber}, D_{duct}, δ_{valve}, L_{duct}, predicts a and b for the choked flow limit: $\dot{m}_{\text{sub}} = a + bp_{\text{chamber}}$
- a and b can be plugged in to LyoPRONTO's design space

Microwave Model

- Microwave model from [3] is implemented in `LyoPronto.jl`
- Can be used to simulate microwave-assisted lyophilization
- Three empirical parameters: product absorption B_{f}, vial absorption B_{vw}, radial heat transfer $K_{\text{vw-f}}$; fit to experiment



Software Verification

- Draw on software engineering best practices
- Unit testing: tests which run single functions of the code, checking that it behaves correctly

```
def test_mass_conservation(example_conditions):
    output = simulate(example_conditions)
    assert water_mass_initial == total_flux(output)
```

- Integration testing: tests that multiple small pieces give correct results when used together

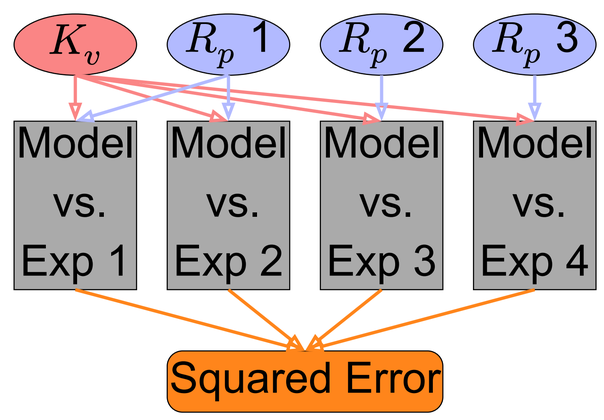
```
def test_correct_plots(input_filename):
    params = read_input(input_filename)
    output = simulate(params)
    plots = generate_plots(output)
    assert len(plots) == 3 # 3 plots generated
```

- Continuous integration (CI): at every code change, run the full suite of tests to ensure that no functions were broken
- LyoPRONTO now uses CI with a suite of tests!

`LyoPronto.jl`: Multi-Experiment Fitting

Consider :

- 4 experiments
- 3 different formulations
- Same lyophilizer & vials
- K_v & R_p unknown
- Thermocouple or other process data available



- Least squares **fit to all 4 cycles** can give less uncertainty in K_v than fitting to each one separately
- With Julia, can take derivatives of the squared error with respect to K_v and each parameter of R_p to find the minimum: more rigor than manual tuning

Future Work

- Design a way to carry out multi-experiment fitting with the web interface (including specifying thermocouples, etc.)
- Add microwave model simulation & fitting to web interface
- Add existing secondary drying models
- More LLM-legible documentation of the code, for agent use

References & Acknowledgements

The original LyoPRONTO Python code was written by Gayathri Shivkumar & Petr Kazarin, with a web interface by Ivan Soroko. David Bernal and Yirang Park are contributing to ongoing development of the Python version.

[1] G. Shivkumar, P. S. Kazarin, A. D. Strongrich, and A. A. Alexeenko, "LyoPRONTO: An Open-Source Lyophilization Process Optimization Tool," *AAPS PharmSciTech*, vol. 20, no. 8, p. 328, Nov. 2019, doi: 10.1208/s12249-019-1532-7.

[2] P. Kazarin *et al.*, "A Compact Model for Lyophilizer Equipment Capability Estimation," *AAPS PharmSciTech*, vol. 23, no. 1, p. 14, Jan. 2022, doi: 10.1208/s12249-021-02167-8.

[3] I. S. Wheeler, A. Darwish, V. Narsimhan, and A. A. Alexeenko, "Mechanistic Models with Experimental Comparison for Microwave-Assisted Lyophilization in a Vial," *AAPS Open*, vol. 12, no. 1, p. 42, Jul. 2026, doi: 10.1186/s41120-026-00173-3.