Virtual Engineering Software Framework for Integrated Biomass Conversion Modeling

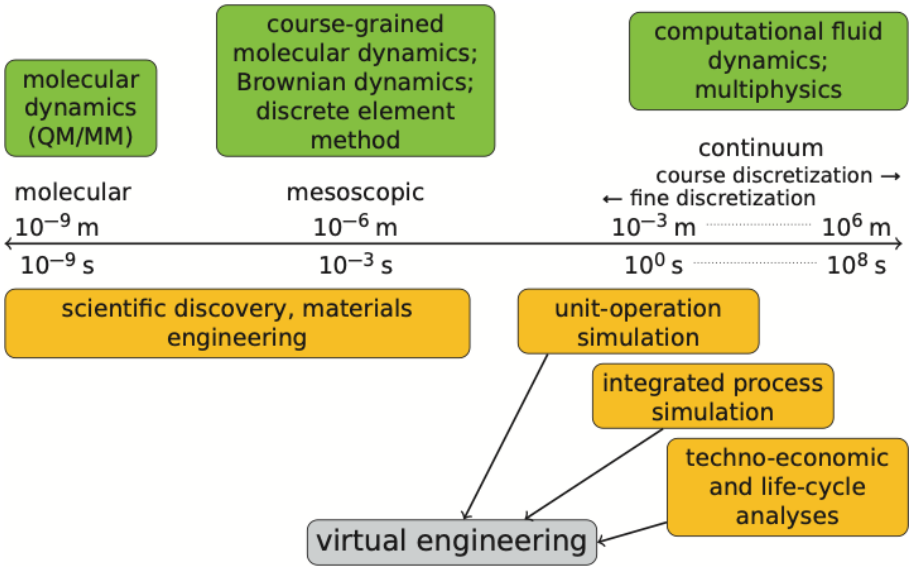
Ethan Young, Hariswaran Sitaraman, Andrew Glaws, Andrew Bartling, James Lischeske, and Jonathan Stickel

2021 AIChE Annual Meeting, November 10, 2021

“Virtual Engineering”

Virtual Engineering (VE)

The process of connecting mathematical models of unit operations and predicting outcomes for an entire chemical process



VE Application

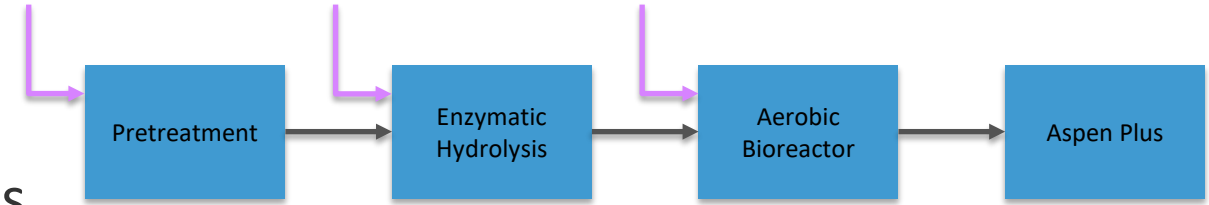
- We apply this VE approach to simulating the **low-temperature conversion of lignocellulosic biomass** to a fuel precursor
- Conversion of this challenging feedstock is modeled through three central unit operations: **pretreatment**, **enzymatic hydrolysis**, and **aerobic bioreaction**
- Numerical models of these operations are the subject of active research across diverse groups

Unit Operations

- **Pretreatment**: Opens up the biomass' resistant lignin shell to expose *more* cellulose chains to the hydrolysis step
- **Enzymatic Hydrolysis**: Enzymes digest the now-exposed cellulose and form sugars like glucose and xylose
- **Aerobic Bioreaction**: These sugars are converted into alcohols/fatty-acids via microbial action for eventual upgrading to fuels like ethanol or bio-diesel

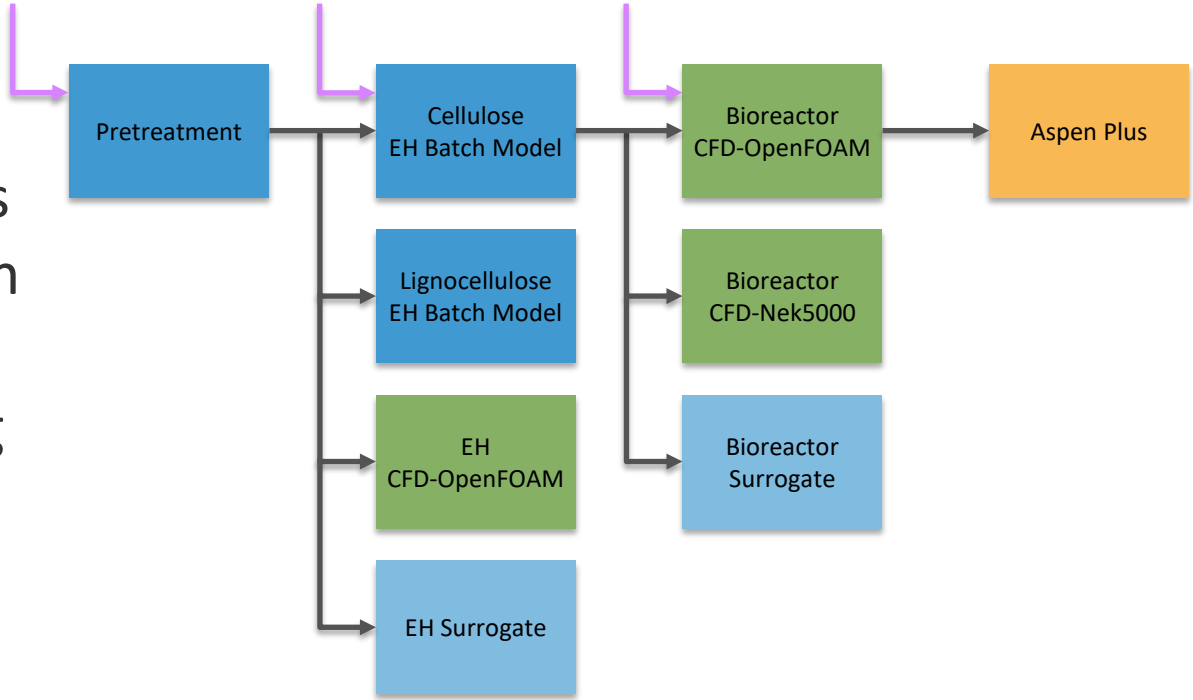
VE Flowchart

Goal: Construct a framework to link together all the models to enable start-to-finish calculations while exposing key operating decisions to the user



VE Flowchart

Goal: Construct a framework to link together all the models to enable start-to-finish calculations while exposing key operating decisions to the user



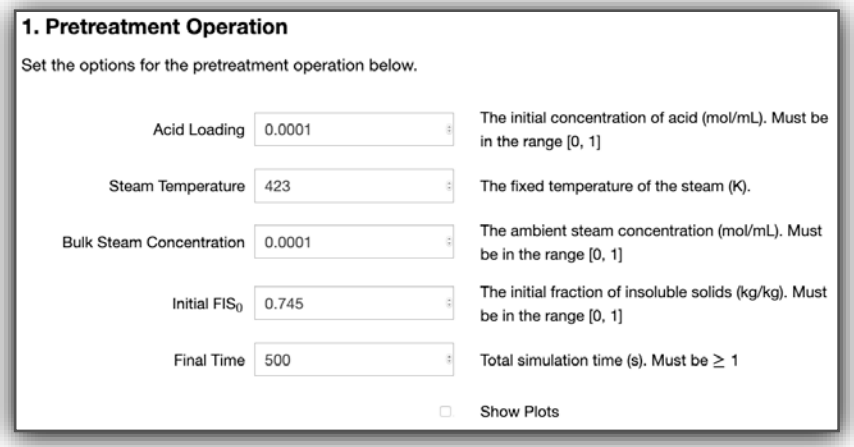
Python + Notebook

- We chose **Python** for the VE package for its flexibility and wide selection of APIs for interfacing with other languages/systems
- The user-facing code is deployed in a **Jupyter Notebook** for the ability to deploy both markdown and code on different hardware, including Mac/Windows/Linux laptops and HPC
- A **Conda environment** makes sharing this package with users and developers relatively easy

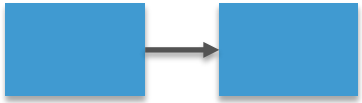
ipywidgets

Using **ipywidgets** within the Notebook provides an easy way to solicit user input for key parameters in the conversion process

```
# Create the collection of widgets
pt_options = wf.WidgetCollection()

# Add option for Acid Loading
pt_options.initial_acid_conc = widgets.BoundedFloatText(
    value = 0.0001,
    max = 1.0,
    min = 0.0,
    description = 'Acid Loading',
    description_tooltip = 'The initial concentration...'
)
```



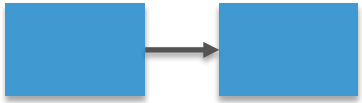
Linking Python Units



- All unit models communicate through a shared YAML file
- At the beginning/end of each step, values are transferred between the YAML file and a Python dictionary

```
pretreatment_input:
  initial_acid_conc: 0.0001
  steam_temperature: 416.0
  bulk_steam_conc: 0.0001
  initial_solid_fraction: 0.75
  final_time: 500.0
  show_plots: false
pretreatment_output:
  fis_0: 0.24501120794572953
  conv: 0.7497337597062437
  X_X: 0.0819860235910653
  X_G: 0.4982436778338859
  rho_x: 79.35199267597184
  rho_f: 0.36666211248901875
```

Linking Python Units



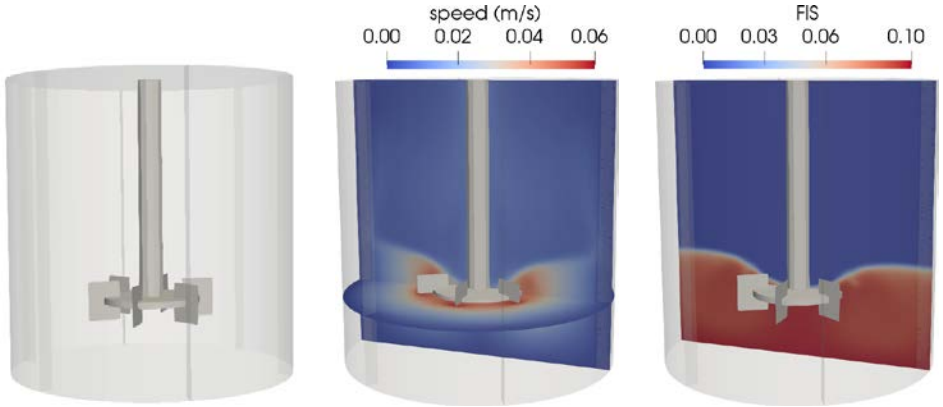
- Minimal function calls at beginning and end of Python models manage this IO

```
from vebio.Utilities import yaml_to_dict, dict_to_yaml

params_filename = sys.argv[1]
ve_params = yaml_to_dict(params_filename)

# Code for unit operation
# ...

dict_to_yaml([ve_params, output_dict], params_filename)
```

- Depending on the level of code abstraction, these values may need to be manually assigned

Linking HPC Jobs

- Variables in the YAML file are written to input files defining either Nek5000 (Fortran) or OpenFOAM (C++) simulations
- These may require as many as 6 days to solve on HPC resources



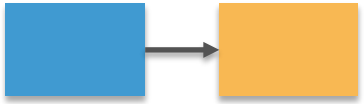
Schematic of EH stir-tank reactor, velocity magnitude of fluid, and FIS contours after 10 hours of hydrolysis

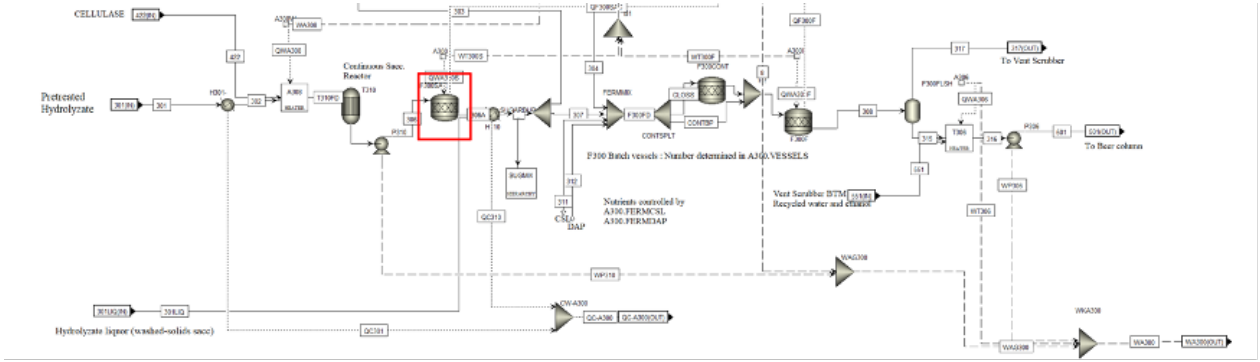
Linking HPC Jobs

- Jupyter Notebook server can be **launched from an HPC compute node** accessed via SSH tunnel
- Enable CFD units based on available resources, fall back to computationally simpler models where necessary

```
In [164]: nodeid_list = !srun hostname

          for nodeid in nodeid_list:
              print()
              !scontrol show node {nodeid}


          NodeName=r3i4n26 Arch=x86_64 CoresPerSocket=18
             CPUAlloc=36 CPUTot=36 CPULoad=0.01
```

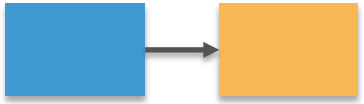
Linking Aspen Jobs





- Connections to Aspen Plus are made using another file-writing technique
- A backup file exported using the Aspen Plus GUI can be **edited through targeted file-writing operations**

Linking Aspen Jobs



- This edited Aspen Plus definition can be deployed directly from the command line using the `pywin32` package
- Spreadsheet macros/calculations can be carried out after the Aspen Plus solve in this same fashion

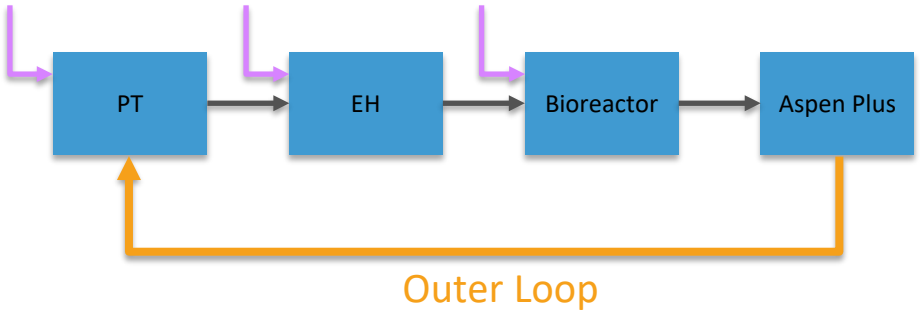
Estimated Glucan Conversion		Enzymatic Hydrolysis Time (hr)				
		72	78	84	90	96
Enzyme Loading (mg/g)	10			0.710		
	15			0.751		
	20	0.765	0.774	**0.782**	0.790	0.797
	25			0.808		
	30			0.830		

MFSP ($/GGE)		Enzymatic Hydrolysis Time (hr)				
		72	78	84	90	96
Enzyme Loading (mg/g)	10			3.96		
	15			4.03		
	20	4.13	4.13	**4.12**	4.11	4.09
	25			4.22		
	30			4.32		

Demo of VE Notebook for 3-Unit Simulation

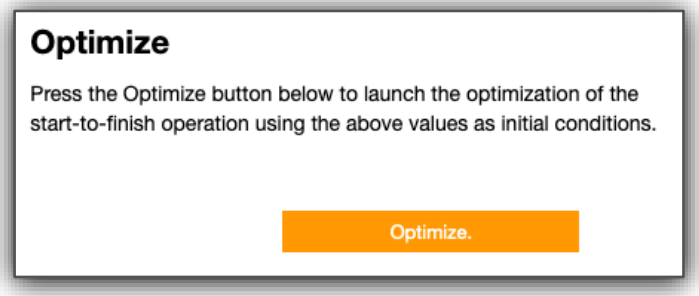
Optimization Studies

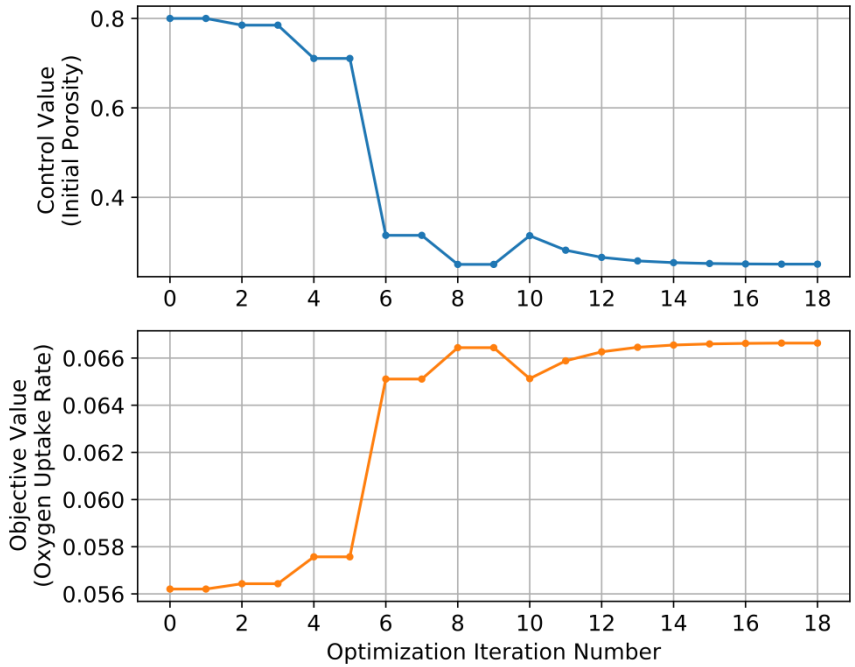
- `SciPy` optimization algorithms (L-BFGS-B, SLSQP) can be used for outer-loop optimization studies
- Objectives, controls, and constraints can be directly defined through Notebook widgets



Optimization Studies

As a proof of concept, we maximize an output of the aerobic bioreactor by adjusting the feedstock's initial porosity subject to constraints





Concluding Remarks

- Connected simple Python scripts, CFD simulations, and Aspen Plus calculations with `vebio`
- For future optimization studies, **accurate surrogate models** will take on even greater importance
- We're currently developing documentation and examples as part of enabling collaboration with new projects and **moving toward a public software release**

Questions?

Ethan.Young@nrel.gov

www.nrel.gov

NREL/PR-2C00-81423

This work was authored by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by U.S. Department of Energy Bioenergy Technologies Office. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes. A portion of the research was performed using computational resources sponsored by the Department of Energy's Office of Energy Efficiency and Renewable Energy and located at the National Renewable Energy Laboratory.