Virtual Engineering: Python framework for engineering process design

Olga Doronina[1][¶]**, Ethan Young**[1]**, Nicholas Carlson**[1]**, Andrew Glaws**[1]**, Hariswaran Sitaraman**[1]**, and Jonathan Stickel**[2]

1 National Renewable Energy Laboratory, Golden, CO, United States **2** Carbon America, Arvada, CO, United States **¶** Corresponding author

Summary

`Virtual Engineering` (`VE`) is a Python software framework designed to accelerate the research and development of engineering processes that are fundamentally defined by multiple unit operations executed in series. VE supports a wide variety of different multi-physics models and integrates them to simulate a complete end-to-end process. To automate the execution of this model sequence, `VE` provides (i) a robust method to communicate between models, (ii) a high-level, user-friendly interface to set model parameters and enable optimization, and (iii) an overall model-agnostic approach that allows new computational units to be swapped in and out of workflows. Although the `VE` framework was developed to support the biochemical conversion of biomass to fuel, we have designed each component to easily accommodate new domains and unit models.

Statement of need

Many industrial and manufacturing operations consist of a sequence of discrete processing steps, including physicochemical transformations, to produce a final product. Often, optimizing the performance of each individual step—such as yield or energy efficiency—does not lead to the overall best outcome. Therefore, it is essential to ensure connectivity between each step and optimize the whole process. This process optimization can depend not only on operating parameters for each step but also on the choice and order of these steps.

Numerical simulations that can support the analysis and optimization of such systems frequently require linking multiple individual models together—each associated with different steps in the overall operational process—so that the outputs of one model can inform the inputs of the next. These models can span multiple levels of physical fidelity and computational costs. `Virtual Engineering` (`VE`) is a Python package that enables the creation of this type of model sequence.

`VE` was originally developed to support the simulation and optimization of the biochemical conversion of lignocellulosic biomass to fuels. This bioconversion process was modeled by linking previously developed computational models as three important unit operations (Figure 1): (i) the *pretreatment* of the feedstock to make cellulose more accessible (Sitaraman et al., 2015), (ii) an *enzymatic hydrolysis* step to digest lignocellulose into sugars (Lischeske & Stickel, 2019; Sitaraman et al., 2019), and (iii) a *bioconversion* step to convert sugars into products in a bioreactor (Rahimi et al., 2018). Finally, the capital and operating costs of the process, and the product's (in this case, ethanol) subsequent minimum feasible selling price through discounted cash flow analysis, were calculated using an Aspen Plus process simulation with techno-economic analysis (TEA) (Humbird et al., 2011).

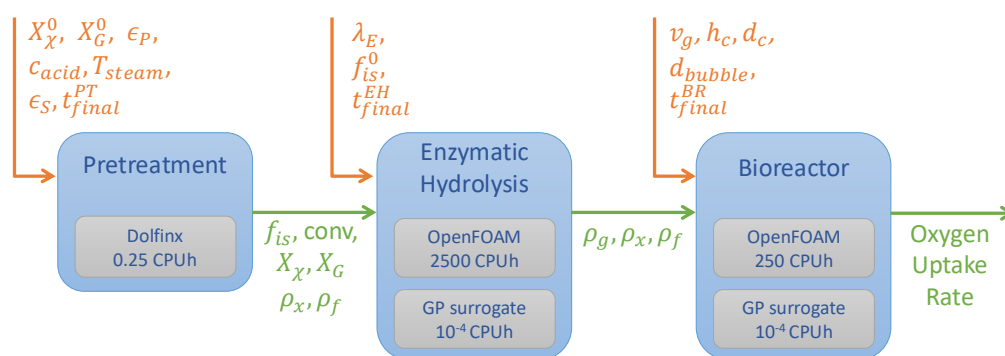**Figure 1:** An example of the end-to-end process for biochemical conversion of biomass to fuel defined and automated within `VE` by connecting different unit models.

Figure 2: Example of `ipywidgets` element defining the choices of EH unit model and its input parameters.

Figure 3: Example of the user-friendly interface for end-to-end process execution.

The models representing these steps vary from one-dimensional finite element models that can be carried out on a modern laptop in seconds to three-dimensional computational fluid dynamics (CFD) simulations that require high-performance computing (HPC) resources and several hours of CPU time. Maintaining a continuous simulation between different operating systems and hardware across potentially long wallclock times (often more than 24 hours) is not well supported with existing workflows. The `VE` package offers an interface with the SLURM job scheduler (Yoo et al., 2003) and modifies and launches OpenFOAM jobs (Weller et al., 1998) to automate the end-to-end bioconversion process. Although this set of functions was necessary to support the original use case of the bioconversion process, the underlying programming interface and object-oriented model wrappers were designed to be as model-agnostic as possible.

To enable users to easily set up and launch new simulations, either by swapping in alternate computational models or specifying a different set of prescribed input parameters, the `VE`

package uses Jupyter notebooks ([Kluyver et al., 2016](#)) for their ability to deploy both GUI elements and performant code on different hardware and operating systems. The GUI elements comprise groups of `ipywidgets` ([Grout et al., 2010](#)) that offer easy methods to solicit and error-check user-input values (see Figure 2). Additionally, this notebook interface enables users to specify either a once-through simulation problem (Figure 3) or an iterative optimization in which the controls and bounds can be easily set via widgets. This combination of features, along with the unique methods developed to support our specific use cases, distinguishes `VE` from other workflow solutions like Airflow[1], Luigi[2], or Dagster[3].

Example of bioconversion optimization

The `VE` package provides an optimization capability that can be easily accessed through the user-friendly `ipywidgets` GUI interface. The notebook supports the selection of multiple control variables, in which case all chosen variables will be tuned to minimize a given objective function over a multidimensional design space. Additionally, users can choose to minimize either the final output or any intermediate output of the process. In the latter case, the number of unit models in the process is automatically adjusted based on the specified objective; for example, executing only the first element in a three-element series if the objective function depends only on element one.

To demonstrate these capabilities of the `VE` package, we optimize the oxygen uptake rate (OUR) that determines fuel yields in the bioconversion process (Figure 1), by solving the box-constrained optimization problem with two controls summarized in Table 1. It is important to note that the objective function, OUR, is an output of the bioreactor (BR) model, while acid loading and enzyme loading are independent inputs to the pretreatment (PT) and enzymatic hydrolysis (EH) models, respectively, which indirectly affect overall fuel yields.

Table 1: Optimization problem with objective and control variables in different unit operations.

		Lower bound	Upper bound	Units
Maximize:	OUR (BR output)			
By Varying:	Acid Loading (PT input)	5×10^{-5}	1×10^{-3}	mol/mL
	Enzyme Loading (EH input)	5	300	mg/g

We solve this optimization problem for two different modeled feedstocks: switchgrass and corn stover. The key distinction between them is the amount of cellulose (glucan), hemicellulose (xylan), and lignin present in the initialization of the pretreatment model (Table 2).

Table 2: Feedstock parameters for two optimization cases adopted from ([Ragauskas et al., 2014](#)).

Feedstock	Xylan (kg/kg)	Glucan (kg/kg)	Lignin (kg/kg)
Switchgrass	0.325	0.463	0.213
Corn Stover	0.360	0.430	0.209

To accelerate the optimization process, we developed surrogate models for the computationally expensive unit operations of EH and BR, utilizing Gaussian process regression and leveraging dimension reduction and active importance sampling. As a result, one can obtain predictions of EH and BR outputs within seconds.

---

[1] airflow.apache.org
[2] github.com/spotify/luigi
[3] dagster.io

Figure 4 illustrates the optimization process for these two different feedstocks. The contour plots of OUR are obtained by sweeping through the parameter space and serve as the background to visualize the behavior of the objective function. These contour plots verify that our optimization algorithm follows the gradients as expected and converges to a reasonable final solution.

To demonstrate the robustness of the optimization algorithm, we used different initial values for the control variables in the switchgrass and corn stover cases. Table 3 displays the initial and final values of the controls, the objective, and the change in the OUR.
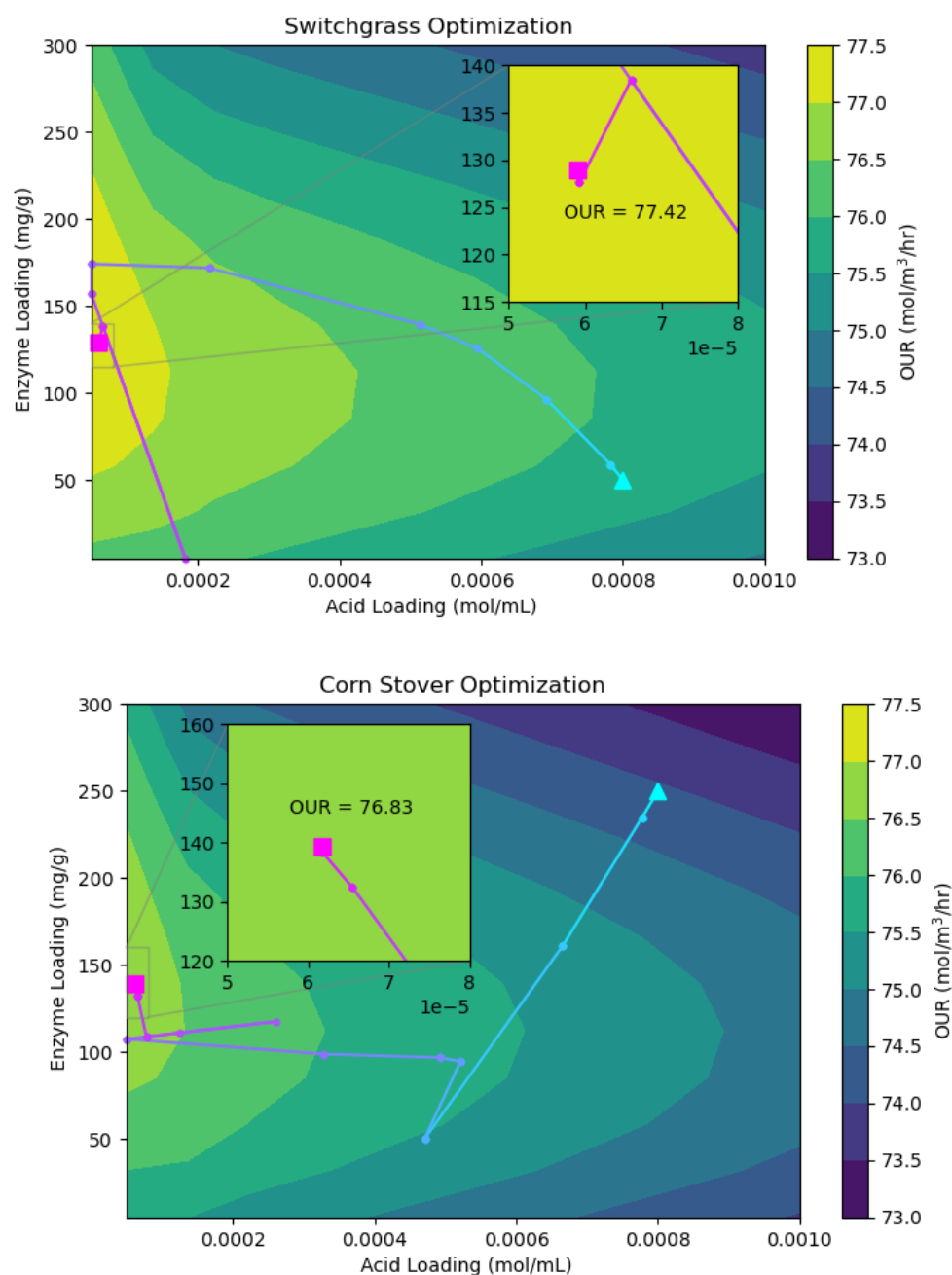

Figure 4: The result of optimizing the amount of acid loading during PT and enzyme loading during EH to maximize OUR for a modeled switchgrass (top) and corn stover (bottom) feedstock. The triangles represent the initial control values, the squares indicate the optimized values, and the lines connecting them show the algorithm's paths to the maximum.

Table 3: Optimization results

Feedstock	Initial Acid	Initial Enzyme	Initial OUR	Final Acid	Final Enzyme	Final OUR	OUR Change
Switchgrass	8×10^{-4}	50.00	75.76	5.89×10^{-5}	128.90	77.42	+2.19%
Corn Stover	8×10^{-4}	250.00	74.06	6.17×10^{-5}	139.28	76.83	+3.73%

One interesting outcome is the difference in the optimal amount of enzyme loading between the two feedstocks, where corn stover is optimally processed with an enzyme loading $\sim 8\%$ greater than that for switchgrass (139.28 vs 128.90 mg/g). Additionally, while a higher acid loading could improve pretreatment conversion, it is not the optimal for the overall process because it produces more byproducts that can negatively affect bioreaction. This emphasizes the value of the VE framework for easily enabling multi-model optimization studies over a wide variety of user-specified conditions and constraints.

VE usage in the research

The VE package was used to quantify the effects of modifying the enzyme loading and total enzymatic hydrolysis processing time on the minimum fuel selling price and the optimization features were used to identify the porosity of the initial feedstock necessary to maximize the OUR. These results, plus a live demonstration of the VE notebook usage, were presented at the 2021 American Institute of Chemical Engineers (AIChE) Annual Meeting (Young et al., 2021). A more comprehensive analysis—including details on the surrogate modeling methodology, model validation against experimental results, and multi-variable optimization outcomes to maximize OUR—was presented at the 2023 Bioenergy Technologies Office (BETO) peer review (Young, 2023).

Finally, we reiterate that, while developed for the specific biomass conversion process discussed above, the VE framework is comprised of generalizable components that can readily extend to model a wide array of industrial and manufacturing workflows. An immediate example for future extension is on thermochemical conversion pathways such as biomass/plastics/solid-waste pyrolysis or gasification that involves similar chemically reacting flows except in a gas-solid environment. VE framework also provides a pathway to combine biochemical and thermochemical approaches and identify overall optimal process conditions, in line with current research directions on hybrid integrated approaches (Begum et al., 2024).

As another example, we may consider the processing of iron ore for steelmaking. This industrial process generally includes multiple stages of grinding, separation, treatment, and some form of pelletizing. Each of the processes can be performed in different ways—e.g., grinding may be performed with bar or ball grinders or using high-pressure grinding rollers that all have different pros and cons as well as different operational parameters. The VE framework could be used to study different grinding methods within the context of a broader iron ore processing operation to optimize different objectives, such as yield, cost, or energy usage. Similar design process questions can arise in applications such as pharmaceutical production, microelectronics fabrication, agricultural processing, and more.

Acknowledgements

References

Begum, Y. A., Kumari, S., Jain, S. K., & Garg, M. C. (2024). A review on waste biomass-to-energy: Integrated thermochemical and biochemical conversion for resource recovery. *Environmental Science: Advances*. https://doi.org/10.1039/D4VA00109E

Grout, J., Jonathan Frederic, & Corlay, S. (2010). Ipywidgets: Interactive widgets for the jupyter notebook. In *GitHub repository*. https://github.com/jupyter-widgets/ipywidgets; GitHub.

Humbird, D., Davis, R., Tao, L., Kinchin, C., Hsu, D., Aden, A., Schoen, P., Lukas, J., Olthof, B., Worley, M, Sexton, D., & Dungeon, D. (2011). *Process design and economics for biochemical conversion of lignocellulosic biomass to ethanol: Dilute-acid pretreatment and enzymatic hydrolysis of corn stover*. National Renewable Energy Lab.(NREL), Golden, CO (United States). https://doi.org/10.2172/1013269

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B. E., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J. B., Grout, J., Corlay, S., & others. (2016). Jupyter notebooks-a publishing format for reproducible computational workflows. *Elpub*, *2016*, 87–90. https://doi.org/10.3233/978-1-61499-649-1-87

Lischeske, J. J., & Stickel, J. J. (2019). A two-phase substrate model for enzymatic hydrolysis of lignocellulose: Application to batch and continuous reactors. *Biotechnology for Biofuels*, *12*, 1–15. https://doi.org/10.1186/s13068-019-1633-2

Ragauskas, A. J., Beckham, G. T., Biddy, M. J., Chandra, R., Chen, F., Davis, M. F., Davison, B. H., Dixon, R. A., Gilna, P., Keller, M., Langan, P., Naskar, A. K., Saddler, J. N., Tschaplinski, T. J., Tuskan, G. A., & Wyman, C. E. (2014). Lignin valorization: Improving lignin processing in the biorefinery. *Science*, *344*(6185), 1246843. https://doi.org/10.1126/science.1246843

Rahimi, M. J., Sitaraman, H., Humbird, D., & Stickel, J. J. (2018). Computational fluid dynamics study of full-scale aerobic bioreactors: Evaluation of gas–liquid mass transfer, oxygen uptake, and dynamic oxygen distribution. *Chemical Engineering Research and Design*, *139*. https://doi.org/10.1016/j.cherd.2018.08.033

Sitaraman, H., Danes, N., Lischeske, J. J., Stickel, J. J., & Sprague, M. A. (2019). Coupled CFD and chemical-kinetics simulations of cellulosic-biomass enzymatic hydrolysis: Mathematical-model development and validation. *Chemical Engineering Science*, *206*, 348–360. https://doi.org/10.1016/j.cej.2015.01.020

Sitaraman, H., Kuhn, E. M., Nag, A., Sprague, M. A., Tucker, M. P., & Stickel, J. J. (2015). Multiphysics modeling and simulation of high-solids dilute-acid pretreatment of corn stover in a steam-explosion reactor. *Chemical Engineering Journal*, *268*, 47–59. https://doi.org/10.1016/j.cej.2015.01.020

Weller, H. G., Tabor, G., Jasak, H., & Fureby, C. (1998). A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computer in Physics*, *12*(6), 620–631. https://doi.org/10.1063/1.168744

Yoo, A. B., Jette, M. A., & Grondona, M. (2003). SLURM: Simple Linux Utility for Resource Management. In D. Feitelson, L. Rudolph, & U. Schwiegelshohn (Eds.), *Job Scheduling*

Strategies for Parallel Processing (pp. 44–60). Springer Berlin Heidelberg. https://doi. org/10.1007/10968987_3

Young, E. (2023). Virtual engineering of low-temperature conversion. *Presented at the 2023 U.S. Department of Energy's Bioenergy Technologies Office (BETO) Project Peer Review.* https://www.nrel.gov/docs/fy23osti/85573.pdf

Young, E., Sitaraman, H., Glaws, A., Bartling, A., Lischeske, J., & Stickel, J. (2021). Keynote: Virtual engineering software framework for integrated biomass conversion modeling. *Presented at the 2021 American Institute of Chemical Engineers (AIChE) Annual Meeting.* https://www.nrel.gov/docs/fy23osti/81423.pdf