Chemical Process Flowsheet Optimization with Full Space, Surrogate, and Implicit Formulations of a Gibbs Reactor

Sergio Bugosena, Carl Lairda, and Robert Parkerb\*

a Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, Pennsylvania, United States

b Los Alamos National Laboratory, Los Alamos, New Mexico, United States

\* Corresponding Author: [rbparker@lanl.gov](mailto:rbparker@lanl.gov).

**Keywords**: Machine learning, Surrogate modeling, Chemical process design, Chemical process optimization, Nonlinear optimization

Computing optimal operating conditions of a chemical process flowsheet is a fundamental aspect of process design. However, full space optimization of these processes is challenging due to the nonlinear complexity of these first principles models, which may cause their Jacobians to become singular. In addition, convergence of these optimization models is sensitive to initial guess, scaling, and problem formulation.

To improve solver convergence reliability when optimizing a chemical process flowsheet, we implement two alternative approaches. First, data-driven optimization based on surrogate modeling. This approach relies on representing complex thermodynamics and chemical reaction equations with tractable and simple polynomials with reduced dimensionality. Second, an implicit function formulation based on [1]. In this approach, the complex first-principles algebraic equations are converged separately as an implicit function and the resulting values and derivatives are communicated back to the optimization solver.

We aim to optimize the autothermal reforming (ATR) section of a hydrogen manufacturing process. We implement the flowsheet using IDAES-PSE [2], and compare instances with full space, surrogate, and implicit function formulations for the reforming reactor.

The full space flowsheet (baseline) is modeled with the first-principles Gibbs Reactor. In the second flowsheet, the Gibbs Reactor is replaced by a surrogate block containing polynomials generated by the ALAMO [3] machine learning framework. Finally, in the third flowsheet, the Gibbs Reactor is replaced by an external implicit function using PyNumero [4]. The optimization problems were solved with IPOPT [5]. We compare convergence reliability, solve time, and solution quality between these three flowsheets.

To demonstrate the potential of surrogate-based and implicit function-based optimization, we solve 64 instances parameterized by inlet natural gas pressure and temperature. Preliminary convergence reliability results are shown in **Figure 1**. The full space formulation was able to converge 38 out of 64 instances in an average of 3.7 s, while the surrogate-based formulation converges all instances in an average of 0.9 s with only a minor increase in solution error. The mean relative error between objective values is 1.5%. The implicit function-based formulation converged 53 of these instances in an average of 34 s, with no solution error.

A screenshot of a computer

Description automatically generated with low confidence

**Figure 1.** Convergence status for the three formulations.

References

1. R. Parker et al., “An implicit function formulation for optimization of discretized index-1 differential algebraic systems,” Computers & Chemical Engineering, vol. 168. Elsevier BV, p. 108042, Dec. 2022.
2. A. Lee et al., “The IDAES process modeling framework and model library—Flexibility for process simulation and optimization,” Journal of Advanced Manufacturing and Processing, vol. 3, no. 3. Wiley, May 13, 2021.
3. A. Cozad, N. V. Sahinidis, and D. C. Miller, “Learning surrogate models for simulation-based optimization,” AIChE Journal, vol. 60, no. 6. Wiley, pp. 2211–2227, Mar. 13, 2014.
4. J. S. Rodriguez et al., “Scalable Parallel Nonlinear Optimization with PyNumero and Parapint,” INFORMS Journal on Computing, vol. 35, no. 2. Institute for Operations Research and the Management Sciences (INFORMS), pp. 509–517, Mar. 2023.
5. A. Wächter and L. T. Biegler, “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” Mathematical Programming, vol. 106, no. 1. Springer Science and Business Media LLC, pp. 25–57, Apr. 28, 2005.