Systems & Control Transactions

Research Article - Peer Reviewed Conference Proceeding
ESCAPE 35 - European Symposium on Computer Aided Process Engineering
Ghent, Belgium. 6-9 July 2025
Jan F.M. Van Impe, Grégoire Léonard, Satyajeet S. Bhonsale,
Monika E. Polańska, Filip Logist (Eds.)

PSE PRESS

Phenomena-Based Graph Representations and Applications to Chemical Process Simulation

Yoel R. Cortés-Peña[a] and Victor M. Zavala[a]

[a] University of Wisconsin Madison, Department of Chemical and Biological Engineering, Madison, Wisconsin, United States
* Corresponding Author: zavalatejeda@wisc.edu.

ABSTRACT

Rapid and robust simulation of chemical production processes is critical to address core scientific questions related to process design, optimization, and sustainability. Efficiently solving a chemical process, however, remains a challenge due to their highly coupled and nonlinear nature. Graph abstractions of the underlying physical phenomena within unit operations may help identify potential avenues to systematically reformulate the network of equations and enable more robust convergence of flowsheets. To this end, we further refined a flowsheet graph-theoretic abstraction that consists of a mesh of interconnected variable nodes and equation nodes. The new network of equations is formulated at the phenomenological level—agnostic to the thermodynamic property package—by extending equation formulations widely used to solve multistage equilibrium columns. Decomposition of the graph by phenomena linearizes material and energy balances across the flowsheet by decoupling phenomenological nonlinearities (e.g., phase equilibrium, chemical reactions). Additionally, we further improved a preliminary simulation algorithm which employs this phenomena-based decomposition and demonstrated that it can perform more rapid and robust simulation of large, highly-coupled systems than sequential modular simulation.

Keywords: Process Simulation, Graph-Theory, Flowsheet Convergence, Distillation, Liquid Extraction

INTRODUCTION

Critical to the design and development of sustainable production processes, chemical process simulation seeks to model the flow rate of material and energy between unit operations and their internal components. Physical phenomena (e.g., phase equilibrium, chemical reactions, adsorption, and heat exchange) which drive a production process also introduce nonlinearities to the material and energy balances. This nonlinear coupling between unit operations is the core difficulty in solving for the steady-state solution of a production process in any algorithmic paradigm (e.g., sequential modular simulation, equation-oriented simulation, pseudo-transient modeling). Large, highly coupled systems lead to substantially longer computation time and lower rates of successful convergence than small, sparse systems [1]. This inability to scale the simulation of complex systems limits researchers' ability to optimize the sustainability of emerging production processes.

In sequential-modular simulation, each unit operation is treated as a black-box model with only material streams as inputs and outputs [2]. All mass, energy, and thermodynamic equations are formulated and solved independently within each unit operation. While specialized convergence strategies are employed to converge individual units, the convergence of recycle systems can be challenging due to nonlinear coupling between unit operations. Equation-based modeling leverages the full set of equations and employs algebraic differentiation to aid numerical methods in finding the steady state solution [3]. Equation-based modeling may be faster than the sequential modular approach particularly when the initial guess is close to the steady state solution. However, highly-coupled networks of equations introduce instabilities that lead to convergence failures.

If a process flowsheet could become aware of the connectivity of the governing phenomenological equations, robust solution strategies could be formulated that aggregate linear relationships and decouple nonlinearities. With the wealth of decomposition algorithms and computing architectures that enable the solution of

complex and large-scale problems, it may be feasible to screen for alternative problem formulations of a production process that result in more rapid and robust convergence. The problem formulation and the selection of a suitable solution method, however, is not a simple task. Graph-theoretic abstractions of the equations within unit operations may provide an avenue to systematically reformulate the network of equations and enable advanced, equation-based convergence of flowsheets.

In classical representations of a chemical process, units are represented as nodes and streams as edges. Fundamentally, each unit encapsulates a set of equations with internal variables and each stream carries a set of variables with a unique connection. At the phenomenological level, however, each variable may relate multiple equations and may not have a unique connection. For example, temperature may play multiple roles in thermodynamic phase equilibrium, energy balances, and reaction rates. Graphical abstractions of process phenomena may unfold the sets of equations and variables present in unit operations to capture how mass, energy, equilibrium, reaction, and transport equations are related through shared variables. Distillation column models are a classic example where equations are decoupled by phenomena. For example, the Wang-Henke bubble point method converges all stages by iteratively solving mass, equilibrium, summation, and enthalpy (MESH) equations [4].

In our previous work, we introduced a flowsheet graph abstraction which represents all the equations and variables as a bipartite graph of equation nodes and variable nodes [5]. Additionally, it was shown that decomposing the network of equations by phenomena improved the convergence speed of an extractive distillation process flowsheet. However, because of the redundancy between variables and how highly coupled the equations are, the abstraction was too complex to provide any key topological insights. The graph abstraction needs to be further refined by eliminating variable redundancy and strategically aggregating intermediary equations not relevant to the decomposition. Harmonizing the graph with respect to the decomposition is the next step towards gaining mathematical insight into how the decomposition improves convergence.

The goal of this study was to further refine and improve the graph abstraction and phenomena-based decomposition introduced in our last study for the broader goal of improving the convergence of large and highly coupled systems. To this end, we completed the following objectives: (i) refined the phenomena-based graph abstraction of chemical process flowsheets, (ii) generalized the graph-based decomposition scheme to any set of unit operations, and (iii) benchmarked a convergence algorithm leveraging this decomposition scheme. We implemented the phenomena-oriented simulation algorithm in BioSTEAM—an open-source process simulation platform implemented in Python [6]—and benchmarked its convergence against sequential modular simulation for an industrial process for the dewatering of acetic acid using ethyl acetate solvent.

MATERIALS & METHODS

Phenomenological Graph Framework

The graph abstractions in this study represent variables as grey ring nodes and equations as solid nodes. The color of equation nodes depends on what subset of equations they pertain to within the phenomena decomposition. Edges connect variables to equations and define shared variables among equations. It is also possible to obtain different graph representations by reformulating the equations. For example, variable nodes can be eliminated via substitution of equations and new edges may be formed to reflect changes in the equation formulation. Nodes that were not relevant to the phenomena decomposition scheme were eliminated to minimize the complexity. By aggregating equations in this manner, the various decomposition schemes are represented at a phenomenological level that is agnostic to the specific thermodynamic property package (e.g., the selection between ideal gas or mixture equations of state has no impact on the equations). Input variables that are held constant within a subgraph are not displayed to avoid creating edges between equation nodes that are decoupled. This reduces connectivity between equations and highlights how the decomposition scheme reduces complexity. PhenomeNode [5], an open-source Python library for phenomena-based graph representations, was used to automate the generation of graphical abstractions.

Phenomenological decomposition

Key material and energy variables are represented by “x” while variables related to coefficients for the material and energy balances are denoted by “y”. Aggregated models (e.g., shortcut columns) are effectively stages with respect to the resolution of the flowsheet. To generalize the equations to an arbitrary reference stage, we introduce subscripts “i” and “o” that describe the unique connectivity of an inlet or outlet to its source or sink, respectively. If a variable does not have an “i” or “o” subscript, it means that the variable pertains to the reference stage. An “s” subscript is used to denote a variable specific to the source stage of a stream (e.g., temperature, phase ratio). In special cases, an additional subscript “j” may be used to reference outlet streams other than the arbitrary “o”. Function definitions are abstracted at the level that is relevant to the decomposition; detailed equations to compute phenomenological variables (e.g., enthalpy) are not described as they depend on the specific property package. Subscript, function, and constant definitions are detailed in Tables 1-3, respectively.

Central equations are presented in the same color as their respective nodes in the graph representations (e.g., material balances are purple). Equations to compute coefficients to the material and energy balances are not considered central equations. Within the graph abstraction, we strategically aggregate these equations to their respective central equations for simplicity. In this study, pressures are assumed to be constants (controlled by valves and pumps) and do not appear as a variable in the graphs nor equations.

Table 1. Subscript definitions.

	Definition		Definition
i	arbitrary inlet	j	arbitrary outlet
o	arbitrary outlet	F	flow rate
s	arbitrary stage	T	temperature
c	arbitrary component	Φ	phase ratio
γ	activity coefficient	S	separation factor
VLE	VLE-specific	R	reaction rate
LLE	LLE-specific	M	molar fraction
E	key energy variable	A	light phase
K	partition coefficient	B	heavy phase

Table 2. Function outputs.

Function	Definition
H	enthalpy flow rate
h	specific enthalpy
N	total flow rate
C	heat capacity flow rate
R	reaction rate
P	phase equilibrium variables
M	molar composition
S	separation factor
Φ	phase ratio (by Rashford-Rice equation)
γ	activity coefficient

Table 3. Constants.

Constant	Definition	Constant	Definition
Q	duty	t, b	split

The material balance equations are used to solve for all flow rates, $x_{F,c}$, given the separation factors, $y_{S,c}$, and the generation (or consumption) of chemicals, $y_{R,c}$. The conservation of mass, material separation, and splitting equations are generalized for any stage as follows:

$$\Sigma_o x_{F,c,o} - \Sigma_i x_{F,c,i} = y_{R,c} \quad (1)$$

$$y_{S,c}\Sigma_{o \in B} x_{F,c,o} - \Sigma_{o \in A} x_{F,c,o} = 0 \quad (2)$$

$$x_{F,c,o \in A} - t_o \Sigma_{j \in A} x_{F,c,j} = 0 \quad (3)$$

$$x_{F,c,o \in B} - b_o \Sigma_{j \in B} x_{F,c,j} = 0 \quad (4)$$

$$y_{R,c} = R_c(x_{F,c,o}, x_T) \quad (5)$$

$$y_{S,c} = y_{K,c} x_\Phi \quad (6a)$$

$$y_{S,c} = S(x_{F,c,o}, x_T) \quad (6b)$$

Eq 6a is only relevant for an equilibrium stage while eq 6b is applicable to any module. The function S will depend on the module. For shortcut columns, S is an iteration of the Fenske Underwood Gilliland method, without bubble and dew point temperature calculations.

At steady state, the enthalpy balance equation at the reference stage and can be generalized as follows:

$$0 = \Sigma_o H_o(x_{F,c,o}, x_T) - \Sigma_i H_i(x_{F,c,i}, x_{T,i}) - Q \quad (7a)$$

Assuming Q is constant, we can reformulate equation 7a in terms of a key energy variable (denoted by "E") by a 1st degree Taylor expansion:

$$\Sigma_{i,s} \Delta x_{E,s} \frac{\partial H_i(x_{F,c,i}, x_T)}{\partial x_{E,s}} - \Delta x_E \Sigma_o \frac{\partial H_o(x_{F,c,o}, x_T)}{\partial x_E} = \Sigma_o H_o(x_{F,c,o}, x_T) - \Sigma_i H_i(x_{F,c,i}, x_{T,i}) - Q \quad (7b)$$

We can reformulate eq 7b as a linear equation by isolating the nonlinear terms and separating variables:

$$\Sigma_{i,s} x_{E,s} y_{E,i} - x_E \Sigma_o y_{E,o} = y_{H,s} \quad (7c)$$

This equation may solve for a range of parameters that depend on the type of phenomena exhibited by the stage and its specifications. For example, VLE is most sensitive to the boil-up ratio, x_Φ, (i.e., the ratio of vapor to liquid flow rate) while that LLE is more sensitive to the temperature, x_T, than the ratio of extract to raffinate. Assuming a constant liquid flow rate, the local sensitivity of Φ on the enthalpy of a stream leaving a stage in VLE is given by:

$$y_{E,o \in A} = \frac{\partial H_{o \in A}}{\partial x_{\Phi,s \in VLE}} = t_o h_{gas}(x_{F,c,o}, x_T) N(x_{F,c,j \in B}) \quad (8)$$

$$y_{E,o \in B} = \frac{\partial H_{o \in B}}{\partial x_{\Phi,s \in VLE}} = 0 \quad (9)$$

When a stream leaving a stage is not in VLE (i.e., superheated or subcooled), the key energy variable is assumed to be the temperature:

$$y_{E,o} = \frac{\partial H_o}{\partial x_T} = C(x_{F,c,o}, x_T) \quad (10)$$

Given these definitions, the term on the right-hand side of eq 7c becomes:

$$y_{H,s} = \Sigma_o H_o(x_{F,c,o}, x_T) - \Sigma_i H_i(x_{F,c,i}, x_{T,i}) - Q - \Sigma_{i,s} x_{E,s} y_{E,i} + x_E \Sigma_o y_{E,o} \quad (11)$$

The enthalpy balance is only relevant for stages where the duty, Q, is specified (e.g., adiabatic stages). Otherwise, when either temperature or vapor fraction is user-specified, the duty can be computed at the end of simulation, after all variables have converged.

Equations related to the VLE criteria depend on user specifications. The variables y_K and x_T for a VLE stage can be evaluated by a bubble point routine (within a

thermodynamic property package; eq 12a). If the temperature of a stage is specified, then y_K and x_Φ are computed through a pressure and temperature flash routine (eq 12b). For other unit operations which include outlet streams in VLE (e.g., shortcut columns), the outlet temperature of each stream can be computed by bubble and dew point routines (eq 12c):

$$y_K, x_T = P_{VLE,c}(x_{F,c,o\in B}) \quad (12a)$$

$$y_K, x_\Phi = P_{VLE,c}(x_{F,c,o}) \quad (12b)$$

$$x_{T,o} = P_{VLE,c}(x_{F,c,o}) \quad (12c)$$

The partition coefficient, y_K, and the phase ratio, x_Φ, for a LLE stage are evaluated using the pseudo-equilibrium approach and the Rashford-Rice equation, respectively:

$$y_{M,c} = M_c(x_{F,c,B}, T, y_{\gamma,c,A}) \quad (13)$$

$$y_{\gamma,c,A} = \gamma_c(y_{M,c}, T) \quad (14)$$

$$y_{K,c} = P_{LLE,c}(x_{F,c,B}, T, y_{\gamma,c,A}) \quad (15)$$

$$x_\Phi = \Phi(y_{K,c}, x_{F,c,o}) \quad (16)$$

Phenomena-based simulation algorithm

The new simulation algorithm iteratively solves the complete flowsheet by sequential substitution. The phenomenological nonlinearities are solved first, then material and energy balances solved as systems of linear equations. To prevent the algorithm from failing due to poor conditioning of the initial guess (i.e., material and energy variables of all streams), we integrate sequential modular simulation for more robust convergence. All together, we propose the following algorithm as a preliminary architecture for phenomena-based simulation:

1. In the absence of an initial guess for $x_{F,c,o}$ and x_E, for each stage, run each unit operation sequentially and use the result as the initial guess.
2. For each unit operation:
   - 2.1 Simulate the unit operation and update all variables associated to the unit.
   - 2.2 Using the linear material balance equations (eqs 1–6), solve $x_{F,c,o}$ for all stages.
   - 2.3 Using the linear energy balance equations (eqs 7c–11), solve $x_{E,s}$ for all stages.
3. Using the VLE criteria (eq 12a-c), solve for y_K and x_T at each stage and/or stream.
4. Solve $y_{K,c}$ and $x_{F,c,o}$ for all LLE stages iteratively until they converge within a specified tolerance:
   - 4.1 Using the pseudo-equilibrium concept (eqs 13-15), solve $y_{K,c}$ for each stage.
   - 4.2 Using the material balance equations (eqs 1–6), solve $x_{F,c,o}$ for all LLE stages as a system of linear equations.
   - 4.3 If $y_{K,c}$ and $x_{F,c,o}$ have not converged, repeat steps 4.1 and 4.2
5. Using the Rashford-Rice equation (eq 16), solve x_Φ for all LLE stages.
6. Solve material and energy balances for the complete flowsheet (steps 2.2 and 2.3).
7. If any variables have not converged under a specified tolerance, repeat steps 2–5.

Acetic Acid Purification Benchmark

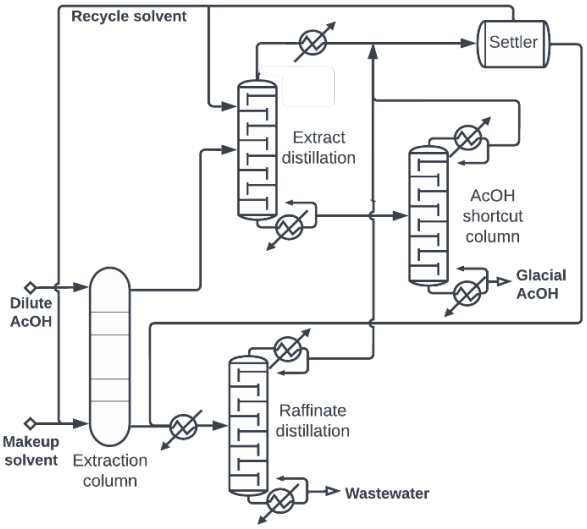

Figure 1. Process flowsheet for acetic acid dewatering.

The industrial separation of acetic acid from water (Figure 1) was chosen as a representative case to benchmark the new convergence algorithm against sequential modular simulation. The system includes highly coupled vapor-liquid equilibrium (VLE) and liquid-liquid equilibrium (LLE) stages and recycle loops that connect the end of the process with the start. In this process, ethyl acetate (EtOAc) is used to extract acetic acid from a dilute aqueous mixture [7]. The extract is distilled to recycle the solvent and recover glacial acetic acid. The raffinate is also distilled to recover EtOAc from the wastewater. Both distillates are sent to a decanter to separate the aqueous phase that forms after condensation. Only results pertinent to flowsheet convergence were considered, including duties and material flows across unit operations and stages. Sizing and costing of unit operations were not considered as they do not impact mass and energy balances and are therefore irrelevant to benchmarking.

The "true" solution of the process was considered to be the best solution (i.e. the solution with the minimum error) out of the two simulation strategies at the final iteration. The simulation error is defined as the sum of the

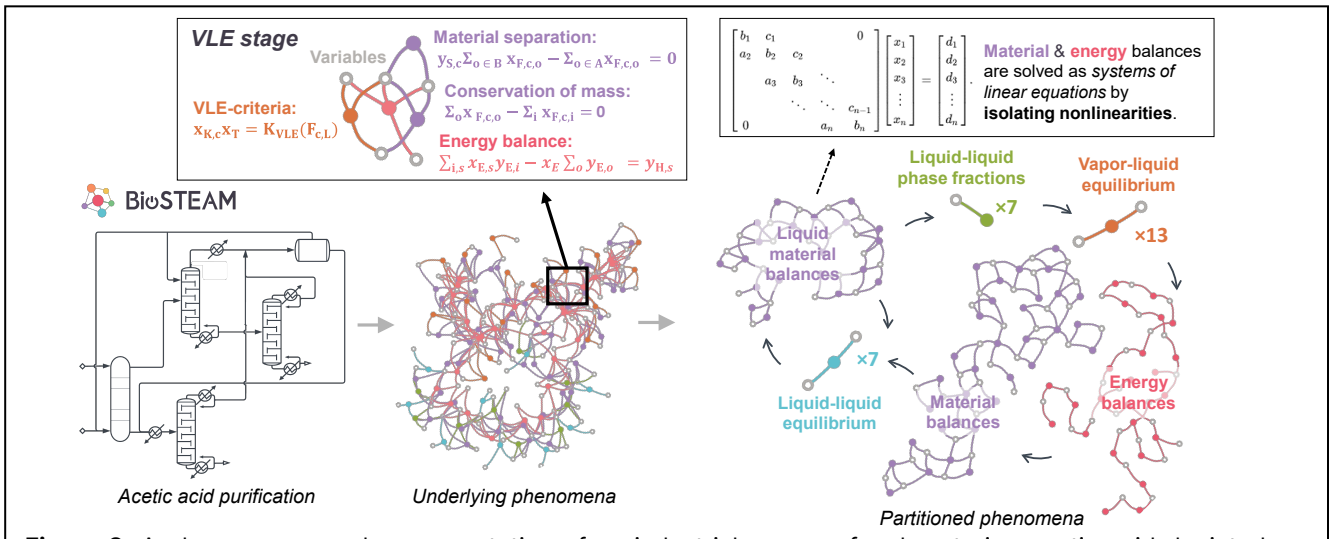

Figure 2. A phenomena graph representation of an industrial process for dewatering acetic acid depicts how decoupling phenomenological equations may result in a more maganeable formulation for complex systems.

change in component flow rates and temperatures of each stream after the simulation of each stage in parallel:

$$\text{Error} = \sum_{s}\sum_{o}\left(\left|x^{n}_{T,s,o} - x^{n-1}_{T,s,o}\right| + \sum_{c}\left|x^{n}_{F,c,s,o} - x^{n-1}_{F,c,s,o}\right|\right)$$

RESULTS & DISCUSSION

The graph representation of the flowsheet for the purification of acetic acid, representing a complex system with 3 recycles and 48 stages, depicts how the phenomenological equations have an underlying structure (Figure 2). While there can be many effective ways to decompose the flowsheet (e.g, sequential or parallel modular), the phenomena-based decomposition has the advantage of linearizing the material and enthalpy balances across all unit operations and their stages by decoupling phenomenological nonlinearities. This decomposition can be considered a generalization of MESH formulations commonly used to solve multistage unit operations. For the special case of multistage vapor-liquid equilibrium columns, these equations reduce to the equations in Wang-Henke's bubble point method [4], which iteratively solves for the bubble point partition coefficients and temperature, then the boil-up ratios of all stages using the linear energy balance equations, and finally the material flow rates using the linear mass balance equations. Additionally, for the special case of multistage liquid-liquid equilibrium, these equations reduce to the decomposition in the Tsuboka method [8], which employs an inner loop using the pseudo equilibrium concept to iteratively solve partition coefficients and the material balance, phase fractions are solved by the Rashford-Rice equation, and, lastly, stage temperatures are solved using the linear energy balances. Although the decomposition of equations is specialized for the type of phenomena (e.g., vapor-liquid equilibrium vs. liquid-liquid equilibrium), these methods have an underlying structure that enables material flows and key energy variables to be solved as systems of linear equations. Through the phenomena-based graph abstraction and the generalization of the mass and energy balances, we can unfold the underlying equations in unit operations in manner that is consistent with classical decomposition approaches for multistage unit operations and leverage existing solution strategies at the unit level to develop a phenomena-oriented flowsheet simulation algorithm.

The new simulation algorithm is a fusion of the sequential modular approach with sequential substitution of the phenomena decomposition. In step 2, the material and energy balances are solved after each unit operation is simulated, enabling new information gathered from a single unit simulation to spread through the flowsheet while consolidating mass and energy balances. Phenomenological nonlinearities are then solved in steps 3-5 and the material and energy balances are solved in step 6, which follows common solution strategies for multistage equilibrium columns. Together, steps 3-6 may help accelerate convergence by treating the entire flowsheet as if it were a single unit operation.

In the benchmark flowsheet for the dewatering of acetic acid, the preliminary phenomena-based simulation algorithm converged in 75% less time than the sequential modular approach (Figure 3 A, B). The phenomena-based algorithm was also able to achieve greater numerical accuracy in simulations than the sequential modular approach. While the phenomena-based approach treats the flowsheet as a single unit, sequential modular accumulates numerical error as it simulates each unit

Ultimately, these results suggest that the phenomena-based approach may enable faster and more robust convergence for large, highly-coupled systems. It is important to note, however, that this approach was only

benchmarked against sequential modular simulation and equation-oriented simulation may converge faster. Additionally, this separation process is limited to only phase-based separations. Future work will focus on benchmarking systems with reactions and mass and energy transfer-based unit operations that are common in a production process against both the sequential modular and the equation-oriented approaches. It is possible that different decompositions may be more effective for different sets of physical phenomena and the decoupling strategies presented here may need to further refinement. We envision a family of phenomena-based simulation architectures that are optimized for different problems based on the relevant phenomena of the system, level of connectivity between unit operations, and chemical interactions that drive coupling between phenomenological equations. Ultimately, the phenomena graph representations introduced in this study may aid in the development of unified decoupling and linearization strategies that can allow for more rapid and robust flowsheet convergence.

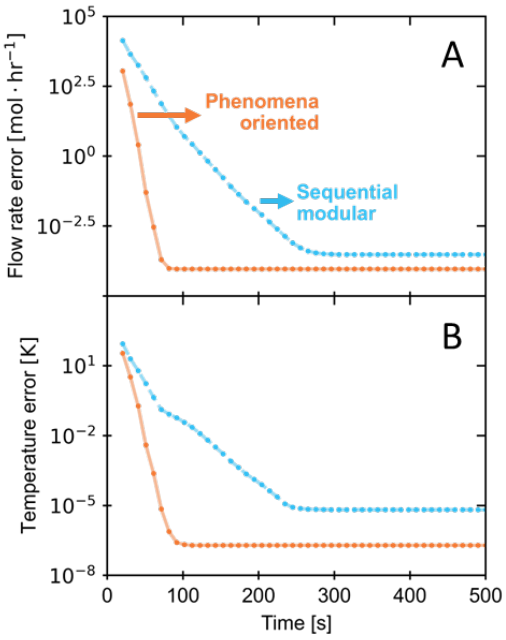

Figure 3. The total error of the flow rate (A) and temperature (B) across all streams is plotted over simulation time of the benchmark case for the phenomena-based (orange) and sequential modular (blue) approaches.

CONCLUSION

Efficiently solving a flowsheet remains a challenge due to the tightly coupled, non-linear nature of chemical processes. The graph representations developed in this study helped identify how decoupling equations by phenomena can be used to formulate a strategy to converge the complete flowsheet. While the proposed algorithm showed promising advantages in convergence speed compared to sequential modular, further analysis is needed to fully characterize the speed, robustness, and applicability to a broader set of chemical processes.

ACKNOWLEDGEMENTS

This material is based upon work supported by the U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy, Bioenergy Technologies Office under Award Number DE-EE0009285.

REFERENCES

1. Mah RSH. Chemical Process Structures and Information Flows. Butterworths; 1990.
2. Motard RL, Shacham M, Rosen EM. Steady state chemical process simulation. AIChE J. 1975;21(3):417-436. doi:10.1002/aic.690210302
3. Bogle IDL, Perkins JD. Sparse newton-like methods in equation oriented flowsheeting. Computers & Chemical Engineering. 1988;12(8):791-805. doi:10.1016/0098-1354(88)80018-8
4. Monroy-Loperena R. Simulation of Multicomponent Multistage Vapor–Liquid Separations. An Improved Algorithm Using the Wang–Henke Tridiagonal Matrix Method. Ind Eng Chem Res. 2003;42(1):175-182. doi:10.1021/ie0108898
5. Cortes-Pena YR, Zavala VM. Graph-Based Representations and Applications to Process Simulation. LAPSE; 2024:129-136. doi:10.69997/sct.184650
6. Cortes-Peña Y, Kumar D, Singh V, Guest JS. BioSTEAM: A Fast and Flexible Platform for the Design, Simulation, and Techno-Economic Analysis of Biorefineries under Uncertainty. ACS Sustainable Chem Eng. 2020;8(8):3302-3310. doi:10.1021/acssuschemeng.9b07040
7. Seader JD, Henley EJ, Roper DK. Separation Process Principles: Chemical and Biochemical Operations. 3rd ed. Wiley; 2011.
8. Tsuboka T, Katamaya T. General design algorithm based on pseudo-equilibrium concept for multistage multi-component liquid-liquid separation processes. Journal of Chemical Engineering of Japan. 1976;9(1):40-45.