Equation-Oriented Flowsheet Simulation and Optimization Using Pseudo-Transient Models

Richard C. Pattison and Michael Baldea
McKetta Dept. of Chemical Engineering, The University of Texas at Austin, Austin, TX 78712

DOI 10.1002/aic.14567
Published online August 17, 2014 in Wiley Online Library (wileyonlinelibrary.com)

Tight integration through material and energy recycling is essential to the energy efficiency and economic viability of process and energy systems. Equation-oriented (EO) steady-state process simulation and optimization are key enablers in the optimal design of integrated processes. A new process modeling and simulation concept based on pseudo-transient continuation is introduced. An algorithm for reformulating the steady-state models of process unit operations as differential-algebraic equation systems that are statically equivalent with the original model is presented. These pseudo-transient models improve the convergence of EO process flowsheet simulations by expanding the convergence basin. This concept is used to build a library of pseudo-transient models for common process unit operations, and this modeling concept seamlessly integrates with a previously developed time-relaxation optimization algorithm. Two design case studies are presented to validate the proposed framework. © 2014 American Institute of Chemical Engineers *AIChE J*, 60: 4104–4123, 2014

Keywords: design (process simulation), optimization, simulation

Introduction

The chemical and petrochemical industries are under intense competitive and regulatory pressure to improve the economic performance, increase energy efficiency, and lower the environmental impact of their facilities. Meanwhile, new developments in the extraction of natural gas are providing the impetus for greenfield investments and the construction of new chemical plants. Process modeling, simulation, and optimization are fundamental aides to these efforts.

Process flowsheet optimization consists of finding the set of operating conditions, material flows, and unit sizes that maximize the economic benefit and/or energy efficiency. It has received significant attention from the advent of Process Systems Engineering as a discipline, and developments in this area have been covered in many publications summarized in several reviews.[1–4]

In spite of these extensive efforts, the deployment of advanced optimization algorithms in practical applications continues to be relatively slow. One of the reasons is that the robust and reliable solution of the underlying equations of a process flowsheet model (i.e., flowsheet simulation) remains a challenging task. From a mathematical perspective, flowsheet simulation requires solving the detailed equations describing the material and energy balances of the process units, as well as the correlations defining the physical properties of the components present in the process. The corresponding system of algebraic equations (model) is typically highly nonlinear, ill conditioned, and poorly structured.

The approaches used to solve this system (simulate the process model) fall broadly into two categories. Sequential modular (SM) simulators set up the problem following the unit-operations structure and solve the system in an iterative manner by "tearing" the recycle streams and solving individual units sequentially until the recycle streams converge.[5] Equation-oriented (EO) modeling environments solve all the nonlinear model equations simultaneously, and are advantageous from the perspective of process optimization due to the simplified calculation of Jacobian and Hessian matrices via automatic or symbolic differentiation. The SM approach is at a significant disadvantage in this regard, being relegated to estimating gradient matrices via computationally expensive and potentially inaccurate finite difference calculations. Despite the optimization-related deficiencies of SM environments, they remain the preferred simulation approach in practical industrial applications as they have a distinct advantage in solving systems of nonlinear equations from poor initial guesses.[2,6,7] It is worth noting that several flowsheet simulation software packages have EO capabilities and are capable of automatic differentiation; however, they all require a SM simulation to initialize the process models, particularly when the flowsheet is rather complex.

Owing to these factors, the optimization of process flowsheets in industrial practice is often a time consuming empirical procedure that requires a considerable amount of effort and skill from a knowledgeable engineer. In turn, this situation highlights the need for developing a robust methodology for solving simultaneously the highly coupled nonlinear equations that correspond to the models of process systems, which can then be used with advanced optimization algorithms.

Further insight in the origin of this challenge can be gleaned from considering the most widely used methods for

Correspondence concerning this article should be addressed to M. Baldea at mbaldea@che.utexas.edu.

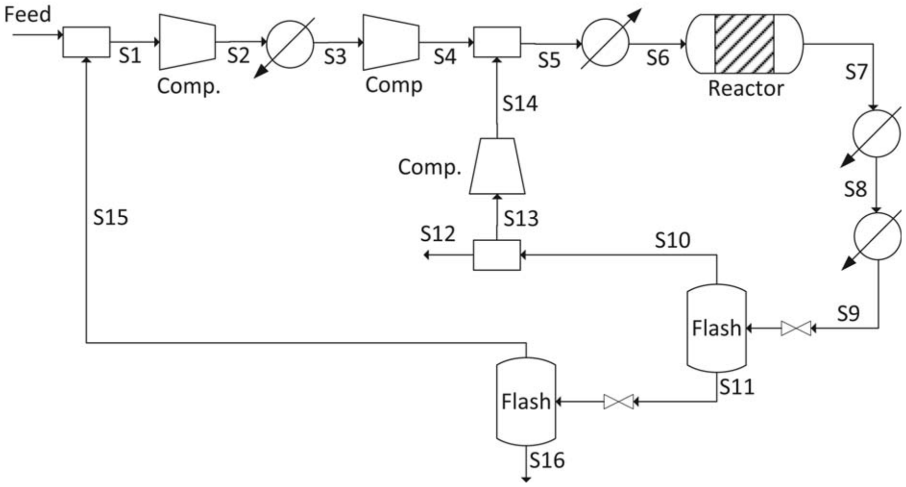

Figure 1. Ammonia synthesis process.[2]

solving nonlinear algebraic equations. They are of the Newton or quasi-Newton-type, which are preferred due to their superlinear convergence rate. However, it is recognized that Newton methods are only locally convergent and require "good" initial guesses that are "close" (in a norm-sense) to the actual solution of the system.[6,8–11]

The need to overcome the difficulty of identifying an appropriate "good" initial guess has spurred the development of alternatives to the Newton class of methods. These include, (1) homotopy continuation (HC),[10,12,13] (2) interval Newton methods,[14–18] (3) terrain methods,[19,20] and (4) various optimization-based approaches where the global minima correspond to the solution.[9,21] Of these, HC has proved the most successful and has found use in many chemical engineering applications, including solving distillation column models[10,22] and detailed reactor models.[11]

Yet, in spite of simulation successes at the unit level, the development of a transparent method for robust and efficient EO flowsheet simulation (and optimization), which is not vitally dependent on identifying an elusive "good initial guess" remains an open question.

In this article, we report on a new process modeling concept based on pseudo-transient continuation (PTC). Our approach is predicated on converting a subset of the algebraic equations of the model of each unit to ordinary differential equations (ODEs), resulting in a differential algebraic equation (DAE) description of the unit operation. The conversion is based on the principle of static equivalence, that is, on obtaining a DAE model that has locally the same steady-state solution as the original system of equations. We show that, from a mathematical point of view, identifying a statically equivalent DAE model presents the benefit of replacing the need to find a "close" initial guess with setting initial conditions for a subset of the unit variables. We show that these initial conditions can in effect be "far away" from the steady-state solution, thereby improving the convergence properties of flowsheet models. The dynamics of the pseudo-transient models are defined by utilizing the natural hierarchy of the dynamics of process phenomena. We use the pseudo-transient concept to build a library of models for the most common process unit operations, which we then seamlessly integrate with a previously developed time-relaxation-based optimization algorithm. We rely on two case studies from the process systems literature to illustrate our novel framework.

Motivating Examples

Consider the ammonia synthesis process in Figure 1 and the ethylene-to-ethanol process in Figure 2. Although the purpose and modeling of these flowsheets is straightforward (in effect, one of these examples is drawn from a well-known textbook), they pose multiple simulation and optimization challenges: In both cases, (1) the models of the individual units contain nonlinear (and potentially discontinuous) relationships, furthermore, (2) the presence of (multiple, significant) material recycle streams results in highly coupled model equations, (3) the equations of state (EOS) used to correctly describe the behavior of the mixtures present in the process are nonlinear and implicit functions of pressure, temperature, and composition; finally, (4) staged equilibrium units (e.g., the absorber and distillation columns in the ethylene-to-ethanol process) rapidly expand the problem size as the number of stages and components increase. Corroborating these observations, we can infer that the simulation of these flowsheets entails, in effect, the solution of a large, highly nonlinear, ill conditioned and poorly structured system of nonlinear equations, which is a challenging task.

Moreover, from a design optimization perspective, each flowsheet features multiple degrees of freedom (e.g., temperatures, pressures, flow rates, heat rates, etc.). A manual search of the decision space (involving seven degrees of freedom for the ammonia case, and nine for the ethylene-to-ethanol process) will not find a (globally) optimal solution in reasonable time. Clearly, advanced optimization algorithms are required to this end; however, their development has been hindered by simulation difficulties as described earlier.

We emphasize that the simulation and optimization problems related to these two case studies are quite small compared to those encountered in many practical situations. Yet, larger and more complex flowsheets pose similar challenges, which will be addressed in this article in a generic context. We will return to these particular examples at the end of the article to demonstrate the effectiveness of our proposed modeling, simulation, and optimization framework.

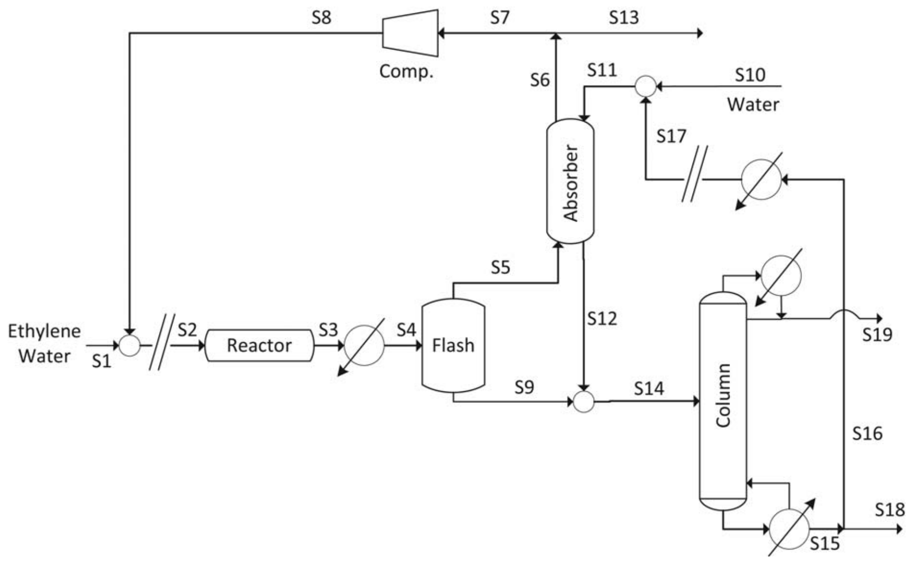

Figure 2. Ethylene-to-ethanol process flowsheet.[5]

Background

In the course of this work, we will rely on several mathematical tools and concepts, which we briefly review below.

Differential algebraic equations

Differential algebraic equation (DAE) systems have seen extensive use in the modeling and simulation of the dynamics of process systems. An excellent introduction to DAEs and the numerical methods used for their solution can be found in the book by Brenan et al.[23] Below, we review some basic ideas that are relevant to our approach, where we will rely on DAE systems of the form

$$\begin{aligned}\dot{\mathbf{a}}(t)&=\boldsymbol{\phi}(\mathbf{a}(t),\mathbf{b}(t),\boldsymbol{\pi},t)\\ \mathbf{0}&=\boldsymbol{\gamma}(\mathbf{a}(t),\mathbf{b}(t),\boldsymbol{\pi},t)\\ \mathbf{a}(0)&=\mathbf{a_0}\end{aligned}\tag{1}$$

In the case of process systems, the dynamic (state) variables $\mathbf{a}(t)$ are typically temperatures and compositions that are computed from mass and energy balances around each process unit operation, $\mathbf{b}(t)$ are the algebraic variables and $\boldsymbol{\pi}$ are process parameters. The algebraic equations γ constrain the evolution of the differential variables and usually include the EOS and other constitutive relations. Process system models are typically described by index-1 DAEs*, although examples of high-index process models are not uncommon.[24] The algebraic equations of index-1 DAE systems are thus formulated such that the Jacobian $\partial\gamma/\partial\mathbf{b}$ is nonsingular.

Consistent Initialization. The simulation of DAE systems comprises two important steps, consistent initialization and time integration. Before the time integration starts, the set of equations $\gamma(\mathbf{a}(t),\mathbf{b}(t),\boldsymbol{\pi},t)$ must be solved to find $\mathbf{b}(0)$ that is consistent with the initial conditions $\mathbf{a_0}$ and the parameter set $\boldsymbol{\pi}$. This step is referred to as "consistent initialization"[25] and, in the case of process systems, is often a difficult problem to solve due to the nonlinear nature of the EOS and physical property correlations. Typically, a nonlinear solver is used to carry out the consistent initialization in an iterative manner. In this article, we will aim to formulate the dynamic component of "pseudo-transient" process models such that the remaining algebraic equations form a linear system [with respect to the algebraic variables $b(0)$] to ensure that the consistent initialization step reliably converges.

Time Integration. The second step is a time stepping routine carried out with an implicit integration method (typically using backward difference formulae) that requires the solution of a nonlinear system of algebraic equations at each time step. Variable-length time steps are typically used to ensure computational efficiency.

Pseudo-transient continuation

PTC has been discussed in the literature in the context of finding the steady-state solution of complex problems described by ill-conditioned models. The essence of the method consists of converting a system of algebraic equations to DAEs by incorporating the natural dynamics of the process. Initial conditions for the differential variables replace initial guesses and a time integration is carried out until steady state is reached (see Figure 3). Assuming that only one steady-state solution exists, the solution is the same as that of the original algebraic equation set; in the case of nonlinear systems, the solution of the transient model will coincide with one of the possible steady-state solutions. The consistent initialization step[25] required to simulate DAE systems can be facilitated by providing initial conditions that allow for a simple computation of the algebraic variables. Simulating to steady state is relatively fast because accuracy in the trajectory is not important (only the steady-state solution must be accurate) and advanced variable step integration methods are available.

Kelley and Keyes[26] studied the convergence of PTC for a set of ODEs if the time step at each iteration is chosen judiciously (see also Refs. 27 and 28), and later extended the analysis to DAE systems.[29] It was shown that as the time

The index of a DAE system of form (1) is defined as the minimum number of differentiations that the algebraic equations must undergo in order to obtain an ordinary differential equation (ODE) representation for the algebraic variables .

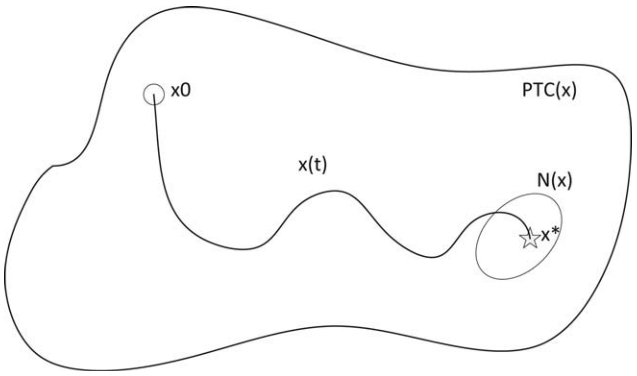

Figure 3. Pseudo-transient continuation increases the basin of initial guesses that converge to the solution (basin of convergence).

Here, $\mathbf{x_0}$ is the initial condition for the pseudo-transient system that follows the trajectory $\mathbf{x}(\boldsymbol{t})$ to the solution, $\mathbf{x}^*$. The region $\mathrm{PTC}(\mathbf{x})$ represents the set of initial conditions for the transient system that will converge to the solution $\mathbf{x}^*$. The region $\boldsymbol{N}(\mathbf{x})$ represents the set of initial guesses that will converge to $\mathbf{x}^*$ when Newton's method is implemented. The regions $\boldsymbol{N}(\mathbf{x})$ and $\mathrm{PTC}(\mathbf{x})$ typically cannot be found analytically, and are usually determined by trial-and-error.

step approaches infinity, the method is exactly Newton's method applied to the original system. However, taking intermediate, small time steps to approach steady state has the advantage that the previous time steps provide good initial guesses for the Newton iterations required for the following time step (assuming that an implicit integration method is used). Most applications in the literature use realistic dynamics to march to the steady-state solution (often starting from "far-away" initial conditions). Modifying the dynamics of the equations for faster convergence has also been attempted.[30] We note that in the latter case (as well as in this article), the transient component of the solution is not physically meaningful; rather, it is a mathematical device used to improve the convergence of the model.

To date, the vast majority of the applications of PTC involve the solutions of discretized partial differential equations (PDEs). The method has proven successful in solving combustion-flow problems,[29] air foil modeling and optimization,[26,31] Poisson–Bolzmann equations,[32] circuit simulation,[33] and structural analysis.[30]

It is important to acknowledge the distinction between PTC and HC; this is discussed in Appendix.

REMARK 1. *Typical implementations of PTC alternate between DAE integration and regular Newton or quasi-Newton iterations; the switch to the Newton method occurs when the system reaches a point within the Newton convergence basin (i.e., the region $N(\mathbf{x})$ in Figure 3). If the Newton iterations fail, the algorithm proceeds by switching back to time integration.*

Pseudo-Transient Flowsheet Modeling and Simulation

In this article, we propose a novel extension of the PTC concept to process systems, with the goal of creating a robust and efficient flowsheet simulation and optimization environment. Consider Figure 4. The dashed lines depict the conventional flowsheet optimization routine: the decision variables ($\mathbf{z}$) are selected by the optimization algorithm, the process model (or the corresponding extended system) is solved ($\mathbf{x}$), and the objective, constraints, and gradients with respect to the decision variables are computed and fed back to the optimization algorithm; the process is repeated until optimality criteria are met. One of the main challenges is solving the algebraic process model with conventional methods. The algorithm presented in this section aims to remedy this problem by converting the algebraic model into a pseudo-transient process model described by a DAE system that can be integrated quickly to steady state.

Rigorous dynamic process models require extensive and detailed information, for example, unit sizing, tray structuring, packing shapes, and material properties. This information is not readily available (or needed) at the design stage, and such models are thus not directly usable to support PTC process flowsheet simulation for process design purposes. Thus, in order to exploit the PTC concept, a simplified transient flowsheet model should be formulated. This model should be, (1), statically equivalent (i.e., have the same steady-state solution as the original algebraic equations)[34] and, (2) have superior numerical solution properties, in the sense that the consistent initialization and time integration steps (described earlier) should proceed transparently and reliably.

The principal task in addressing this intuitive goal is to establish an appropriate structure for the pseudo-transient, DAE process models. Specifically, the dynamic variables in the DAE formulation should be selected judiciously so that the consistent initialization step is simplified, for example, by ensuring that the algebraic equations that must be solved at the consistent initialization step are either linear or decoupled. Then, once a consistent initial value for the algebraic variables is computed, the time integration to steady state must be stable and rapidly converge to the solution.

In this section, we describe a strategy for deriving pseudo-transient models of unit operations that follow the aforementioned paradigm. We work within the unit operation paradigm (i.e., treating unit operations as building blocks to construct the flowsheet) because, as it will be shown, the inherent structure of process unit models constitutes a sound starting point for the derivation of the desired pseudo-transient models. We will extend these ideas to the flowsheet level later in the article.

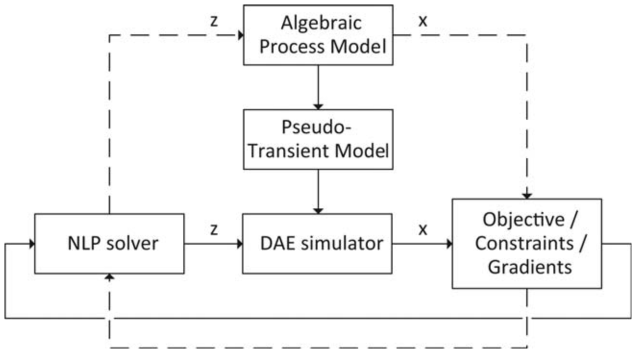

Figure 4. The conventional flowsheet optimization routine follows the dashed line.

The framework developed in this article consists of converting the algebraic model into a pseudo-transient process model with an equivalent steady-state solution, which can be obtained by a DAE time integration.

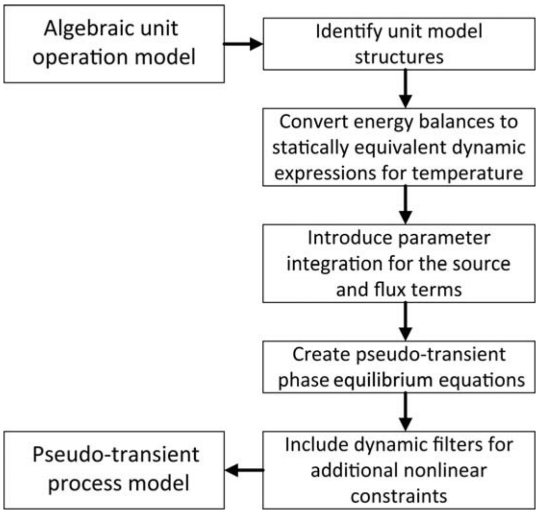
Algebraic unit operation model
Identify unit model structures
Convert energy balances to statically equivalent dynamic expressions for temperature
Introduce parameter integration for the source and flux terms
Create pseudo-transient phase equilibrium equations
Include dynamic filters for additional nonlinear constraints
Pseudo-transient process model

Figure 5. The flowchart describes the process of converting the algebraic models of unit operations into statically equivalent pseudo-transient models.

Figure 5 outlines the process for converting the algebraic system of equations describing the model of a unit operation to a statically equivalent index-1 DAE system. We start by considering the steady-state algebraic model of a unit operation. The algorithm consists of an initial step for selecting the dynamic variables based on the structure of the unit operation model, followed by a filtering step whereby the dynamic variables (as well as a subset of the model parameters) are redefined as states in a subsystem of ODEs.

Step 1: Unit model structure

The steady-state models of process unit operations can be represented by the general structure

$$0=\mathrm{CONV}+\mathrm{GEN}+\mathrm{FLUX} \tag{2}$$

where CONV represents the convective terms in the unit (i.e., inlet and outlet material and energy flow rates), GEN represents the rate of generation (or consumption), and FLUX represents the nonconvective addition or subtraction of material or energy (e.g., a heating element in a reaction vessel). Additional algebraic equations, that is, the equation of state and physical property correlations, are also present in each unit operation model.

Conventional unit operation models can be broken down into three distinct classes: general single-phase units (reactors, heat exchangers, etc.), vapor–liquid separators (flash tanks, equilibrium stages in distillation/absorption/stripping), and pressure changers (compressors, turbines, pumps, etc.). Unit operation libraries in most flowsheet simulators typically follow this classification paradigm. We will assume, for now, that the inlet mass and energy flow rates to each unit are known and can be treated as parameters (an assumption that will be relaxed later). We will also assume that a physical properties package is available that returns properties when required (e.g., enthalpy, fugacity, vapor pressure), along with their analytical derivatives with respect to relevant quantities.

Single-Phase Units. Referring to Eq. 2, a general single-phase unit typically has a mass and energy balance structure of the form

$$0=(M_{i,\mathrm{in}}-M_{i,\mathrm{out}})+r_i \tag{3}$$

$$0=(H_{\mathrm{in}}-H_{\mathrm{out}})+H_{\mathrm{gen}}+Q_{\mathrm{ext}} \tag{4}$$

where M is the molar flow rate of each component i, r is the generation (or depletion) rate of each component, H is the energy flow rate, H_{gen} is the heat generated, and Q_{ext} is the external heat added to or removed from the system. These terms can be grouped into the terms in Eq. 2 above: the convective terms are within parentheses (*CONV*), the molar and energy generation terms (*GEN*) are typically a result of chemical reaction, and there is heat added or removed (*FLUX*) from the environment. Notice that the convective outlet terms (i.e., $M_{i,\mathrm{out}}$ and H_{out}) appear linearly in this case, while the *GEN* and *FLUX* terms can be nonlinear functions of temperature and composition.

Multi-Phase Units. Units where multiple phases are present (e.g., vapor–liquid separators) have similar overall mass and energy balance structures

$$0=(M_{i,\mathrm{in}}-M^L_{i,\mathrm{out}}-M^V_{i,\mathrm{out}}) \tag{5}$$

$$0=(H_{\mathrm{in}}-H^L_{\mathrm{out}}-H^V_{\mathrm{out}})+Q_{\mathrm{ext}} \tag{6}$$

where the superscripts L and V correspond to the liquid and vapor phase, respectively. Again, the terms in parentheses correspond to the convective terms in the unit, and Q_{ext} corresponds to the external heat addition rate. In this case, there are no generation terms, but these could be added for, for example, multiphase reactors or reactive distillation stages. Assuming again that the inlet flows can be treated as parameters, the outlet convective terms ($M^L_{i,\mathrm{out}}$, $M^V_{i,\mathrm{out}}$, H^L_{out}, and H^V_{out}) are typically determined from nonlinear relationships, that is, the fraction defining the separation of each component in each phase is a nonlinear function of the temperature, pressure, and composition (e.g., equating the fugacities in each phase).

Pressure Change Units. Finally, the structure of the mass and energy balances of units that result in pressure changes, and consequently, temperature changes is

$$0=(M_{i,\mathrm{in}}-M_{i,\mathrm{out}}) \tag{7}$$

$$0=(H_{\mathrm{in}}-H_{\mathrm{out}})+(\dot{W}-Q_{\mathrm{ext}}) \tag{8}$$

In such units, there is work done on or by the system, $\dot{W}$ as well as heat input or loss, Q_{ext}. The corresponding FLUX terms are typically nonlinear, whereas the convective (CONV) terms are linear.

Step 2: Define temperature as a state variable

In all the generic unit operation models above, either the material or energy balance (typically both) are nonlinear functions of temperature. For example, in the generic single-phase unit model, the reaction term in the material balance likely follows an Arrhenius relationship, where temperature appears in the denominator of an exponential, and the enthalpy calculations in the energy balance are nonlinear and implicit functions of temperature. At the consistent initialization step, these nonlinear and coupled equations may not converge to the solution if the initial estimate of the temperature is not close to the solution.

To alleviate this problem, we define temperature as a state variable, rewriting the energy balance as an ODE with a steady-state solution that is equivalent to that of the original energy balance. The dynamic energy balance is written as a first-order ODE in temperature, with the derivative of temperature with respect to pseudo-time being proportional to the residual of the static energy balance equation

$$\left(\tau_T \frac{H_{\text{in}}}{T_0}\right)\frac{dT}{dt} = (H_{\text{in}} - H_{\text{out}}) + H_{\text{gen}} + Q_{\text{ext}} \tag{9}$$

$$T(t=0) = T_0 \tag{10}$$

where T is the temperature in the unit (or exiting the unit), t is the pseudo-time variable, T_0 is the initial condition for temperature, and τ_T is a time constant (the selection of this will be addressed later in the article). Notice that the filter time constant is weighted by $\frac{H_{\text{in}}}{T_0}$ in order to ensure that the units are consistent.

Stability Considerations. We analyze the stability of the resulting dynamical system based on the following premises:

- Defining the right-hand side of Eq. 9 as $F(T)$, stability requires that $\frac{\partial F}{\partial T} < 0$
- H_{in} in Eq. 9 is known and fixed per our previous assumption
- the enthalpy H_{out} is a globally (nonstrictly) increasing function of temperature (i.e., $-\frac{\partial H_{\text{out}}}{\partial T} \leq 0$, if temperature increases, enthalpy will not decrease, assuming constant pressure and composition)
- the term Q_{ext} is often a function of a temperature difference, and thus of temperature. For example, consider a jacketed vessel (for either heating or cooling) with heat-transfer coefficient U and heat-transfer area A with the jacket kept at constant temperature T_{jacket}

$$Q_{\text{ext}} = UA(T_{\text{jacket}} - T) \tag{11}$$

thus, $\frac{\partial Q_{\text{ext}}}{\partial T} < 0$. In other cases (e.g., modeling heat integrated reactors and heat integrated distillation columns), Q_{ext} is determined in other sections of the flowsheet. In such cases, we will assume that Q_{ext} is known and fixed similar to H_{in}.

In light of the above, we likely have $\frac{\partial H_{\text{in}} - H_{\text{out}} + Q_{\text{ext}}}{\partial T} < 0$ as required by the stability condition. However, $\frac{\partial H_{\text{gen}}}{\partial T} > 0$ if the reaction is exothermic; if the contribution of this term outweighs the contribution of the other terms, then $\frac{\partial F}{\partial T} > 0$ and the system (9) will be unstable. We will rely on the time scale properties of the pseudo-transient model (including the gradual increase of heat source terms, as described in the following section) to improve the stability properties of this pseudo-transient model, and ensure convergence of the simulation to the steady state at the flowsheet level. Other approaches for ensuring stability of this model (e.g., assuming that Q_{ext} is an optimization decision variable, its initial guess should be set to be sufficiently large) can be taken on a case by case basis.

Step 3: Introduce state variables to modulate the contribution of source terms

The source terms in the material and energy balance equations may be at the origin of significant stiffness and nonlinearity in process models. We are interested in eliminating these traits at the consistent initialization step of the pseudo-transient model (i.e., at pseudo-transient time $t = 0$), and gradually reintroducing them so that they are fully accounted for in the steady-state solution.

We accomplish this by defining a parameter, α that multiplies the source or flux terms and converts it to a state variable that is initially zero, but increases gradually to 1 at steady state. A differential equation must be added for this additional state.

$$\begin{aligned} \tau_p \frac{d\alpha}{dt} &= 1 - \alpha \\ \alpha(t=0) &= 0 \end{aligned} \tag{12}$$

As an example, consider the equations for a jacketed reaction vessel where multiple reactions (denoted by subscript j) have kinetics defined by the Arrhenius rate law, and heat transfer between the vessel and jacket is driven by the heat-transfer coefficient times the heat-transfer area, UA, multiplied by a (potentially nonlinear) temperature driving force ΔT (e.g., temperature difference or log mean temperature difference)

$$\begin{aligned} 0 &= (M_{i,\text{in}} - M_{i,\text{out}}) + \sum_j k_j^0 f_j(T, \mathbf{M}) \\ \left(\tau_T \frac{H_{\text{in}}}{T_0}\right)\frac{dT}{dt} &= (H_{\text{in}} - H_{\text{out}}) + \alpha\left[\sum_j \Delta H_j k_j^0 f_j(T, \mathbf{M}) + UA\Delta T\right] \\ \tau_p \frac{d\alpha}{dt} &= 1 - \alpha \\ \alpha(t=0) &= 0 \end{aligned} \tag{13}$$

where $f_j(T, M)$ is the reaction rate expression for reaction j, τ_p is the parameter integration time constant, ΔH_j is the heat of reaction j, and k^0 is the pre-exponential factor.

The model is now a DAE system with the same steady-state solution as the original model equations (if the time derivatives on the left-hand side are set to zero, the parameters are equivalent to their nominal values, and the material and energy balance equations are static).

Remark 2. *Following the transformations described above, the consistent initialization of the pseudo-transient models of general single-phase units requires solving a simple linear system of equations (generally just $M_{i,\text{out}} = M_{i,\text{in}}$). Furthermore, these transformations can be easily extended to the multiphase units and pressure change units described above. This is addressed in the sequel.*

Remark 3. *Along the same lines, it is interesting to note that the remaining algebraic variables in the pseudo-transient simulation framework are mostly material flow rates (the total mass balances around each unit operation will always hold). This result is similar to the IDEAS approach to flowsheet optimization proposed by Wilson and Manousiouthakis[35]; where the outlet mass flow rates for each unit have a linear relationship to the input. The nonlinear components of the model are computed a priori over a discretized space corresponding to different compositions and temperatures, thus limiting the mass exchange network search space to a convex set.*

Step 4: Phase equilibrium

Earlier, it was noted that the convective terms in the vapor–liquid phase separators are nonlinear. Applying the rules stated above to convert the nonlinear algebraic model to an equivalent DAE model will still not guarantee that the consistent initialization step will converge as the equations remain nonlinear. To deal with this, a robust method for initializing the material balance equations for a vapor–liquid separation calculation is introduced (assuming that pressure and heat rate are specified). First, we rewrite the equations

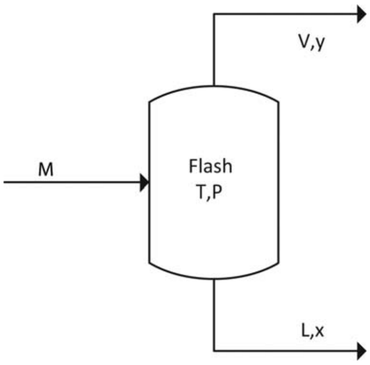

Figure 6. Flash tank model.

terms of vapor and liquid mole fractions (y and x, respectively) and the total vapor and liquid flow rates leaving the unit (V and L, respectively). The material balance equations for an N component flash calculation (Figure 6) are[36]

$$0=M_i-Vy_i-Lx_i \tag{14}$$

$$0=y_i-K_ix_i \tag{15}$$

$$0=\Sigma_i x_i-1 \tag{16}$$

$$0=\Sigma_i y_i-1 \tag{17}$$

$$K_i=\frac{\Phi_i^L}{\Phi_i^V}=f_{\text{split}}(T,P,\mathbf{x},\mathbf{y}) \tag{18}$$

$$0=H_{\text{in}}-H_{\text{out}}^L-H_{\text{out}}^V+Q_{\text{ext}} \tag{19}$$

The split fractions (K_i) are defined by the ratio of the liquid and vapor fugacity coefficients (Φ_i^L and Φ_i^V) which are a function, $f_{\text{split}}(T,P,\mathbf{x},\mathbf{y})$, (depending on the physical property model used) of temperature (T), pressure (P), and composition ($\mathbf{x},\mathbf{y}$). The remaining unknown variables are the vapor and liquid flow rates and the compositions (V, L, $\mathbf{x}$, $\mathbf{y}$). The split fractions can be calculated explicitly from these variables. Notice that the nonlinearities in the remaining equations are the bilinear multiplications of the vapor and liquid flow rates with the respective phase compositions, and the nonlinear relationship defining the split fractions in terms of the compositions.

To ensure the convergence of the consistent initialization step of the resulting pseudo-transient model, these nonlinearities must be dealt with. To eliminate (at $t=0$) the nonlinear relationship between the split fractions and the compositions, the split fractions, K_i, are selected as dynamic variables, using the corresponding values computed via Raoult's law as initial conditions. We note that this choice makes the calculation agnostic to composition, and only a function of temperature and pressure, which are assumed to be available. A time-dependent relation then ensures the transition of the split fraction to the ratio of the fugacities in each phase during the time integration. To deal with the bilinear terms in Eq. 14, the vapor and liquid flow rates, V and L, are selected as state variables, and the material balance of the lightest and heaviest components (denoted by subscripts l and h, respectively) are the corresponding differential equations. With V, L, and K given through initial conditions, the remaining algebraic variables (the vapor and liquid compositions) can be computed by solving a linear system of equations at the consistent initialization step. The resulting pseudo-transient model is thus given by

$$(\tau_f y_l)\frac{dV}{dt}=M_l-Lx_l-Vy_l \tag{20}$$

$$(\tau_f x_h)\frac{dL}{dt}=M_h-Vy_h-Lx_h \tag{21}$$

$$0=M_i-Vy_i+Lx_i \quad i\neq h,l \tag{22}$$

$$0=y_i-K_ix_i \tag{23}$$

$$0=\Sigma_i x_i-1 \tag{24}$$

$$0=\Sigma_i y_i-1 \tag{25}$$

$$\tau_p\frac{dK_i}{dt}=\frac{\Phi_i^L}{\Phi_i^V}-K_i \tag{26}$$

$$(\tau_T\frac{H_{\text{in}}}{T_0})\frac{dT}{dt}=(H_{\text{in}}-H_{\text{out}}^L-H_{\text{out}}^V)+Q_{\text{ext}} \tag{27}$$

The set of equations (20–27) has the same steady-state solution as the original system (14–19). V, L, T, and K are given initial conditions as follows

$$V_0=\frac{1}{2}\Sigma_i M_i \tag{28}$$

$$L_0=\frac{1}{2}\Sigma_i M_i \tag{29}$$

$$T_0=\frac{T_{\text{bub}}+T_{\text{dew}}}{2} \tag{30}$$

$$K_{i,0}=K_{i,I}=P_{\text{vap},i}(T)/P \tag{31}$$

where T_{bub} and T_{dew} are the bubble and dew points, respectively, and $K_{i,I}$ is the ideal split fraction calculated from Raoult's Law. Assuming, for example, $N=4$ components, the initialization problem consists of solving the linear system

$$\begin{bmatrix} -K_1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -K_2 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -K_3 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -K_4 & 0 & 0 & 0 & 1 \\ 0 & L & 0 & 0 & 0 & V & 0 & 0 \\ 0 & 0 & L & 0 & 0 & 0 & V & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}=\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ M_3 \\ M_4 \\ 1 \\ 1 \end{bmatrix} \tag{32}$$

Stability Considerations. For a two-component flash, assume that the split fractions, K_i are initially fixed at the values given by Raoult's law. This assumption is valid if τ_f in Eqs. 20 and 21 is much smaller than τ_T or τ_p, because the split fractions (and temperature) will not exhibit a significant change while the vapor and liquid molar flow rates, V and L, respectively, approach their pseudo-steady-state values. The compositions, x_i and y_i can be computed directly as

$$x_l=\frac{1-K_h}{K_l-K_h} \tag{33}$$

$$x_h=\frac{K_l-1}{K_l-K_h} \tag{34}$$

$$y_l=K_l\frac{1-K_h}{K_l-K_h} \tag{35}$$

$$y_h = K_h \frac{K_l - 1}{K_l - K_h} \tag{36}$$

where the subscripts l and h correspond to the light and heavy components in the separation, respectively. The dynamic material balance then becomes a linear system of ODEs

$$\frac{d}{dt}\begin{bmatrix} V \\ L \end{bmatrix} = \frac{1}{\tau_\mathrm{f}} \begin{bmatrix} \dfrac{K_l K_h - K_l}{K_l - K_h} & \dfrac{K_h - 1}{K_l - K_h} \\ \dfrac{K_h - K_h K_l}{K_l - K_h} & \dfrac{1 - K_l}{K_l - K_h} \end{bmatrix} \begin{bmatrix} V \\ L \end{bmatrix} + \begin{bmatrix} M_l \\ M_h \end{bmatrix} \tag{37}$$

which has eigenvalues

$$\lambda_1 = \frac{1 - 2K_l + K_l K_h + \sqrt{1 + 4K_h - 6K_l K_h - 4K_h^2 + 4K_l K_h^2 + K_l^2 K_h^2}}{2(K_l - K_h)} \tag{38}$$

$$\lambda_2 = \frac{1 - 2K_l + K_l K_h - \sqrt{1 + 4K_h - 6K_l K_h - 4K_h^2 + 4K_l K_h^2 + K_l^2 K_h^2}}{2(K_l - K_h)} \tag{39}$$

The linear system is stable if the eigenvalues have negative real parts. Both are negative if the following criterion is satisfied

$$(K_l - 1)(K_h - 1)(K_l + K_h) < 0 \tag{40}$$

and in the two-phase region, $K_l \in (1, \infty)$, and $K_h \in (0, 1)$. Thus, the material balance is always stable for the two-phase (two component) system. For more than two components, this derivation is not complete, but many trials of different systems (including some with many components) have proven to be stable, suggesting that a general stability proof could be derived.

Switching Phase Regime. The solution of the pseudo-transient models involves the time integration of the corresponding DAE system. During the course of integrating the DAE system to steady state, it is possible that the process conditions will change such that they no longer correspond to a two-phase regime (i.e., the material in the unit will be single phase). This change could be temporary or permanent depending on the steady-state temperature and the assigned unit pressure. This situation can be dealt with by extending Eqs. 20 and 21 as follows

$$\begin{bmatrix} (\tau_\mathrm{f} y_l)\dfrac{dV}{dt} \\ (\tau_\mathrm{f} x_h)\dfrac{dL}{dt} \end{bmatrix} = \begin{cases} \begin{bmatrix} 0 \\ M_h - V y_h - L x_h \end{bmatrix} & T \leq T_\mathrm{bub} \\ \begin{bmatrix} M_l - L x_l - V y_l \\ M_h - V y_h - L x_h \end{bmatrix} & T_\mathrm{bub} < T < T_\mathrm{dew} \\ \begin{bmatrix} M_l - L x_l - V y_l \\ 0 \end{bmatrix} & T \geq T_\mathrm{dew} \end{cases} \tag{41}$$

Specifically, if system temperature approaches a phase boundary, the flow rate of the relevant outlet stream will approach zero. For example, the vapor stream will disappear at the bubble point. The expressions above ensure that once the phase boundary is crossed, the vapor flow will not change ($\frac{dV}{dt} = 0$) (i.e., it will stay at a near-zero value) until the two-phase region is reentered.

The formulation (41) introduces an implicit discontinuity in the pseudo-transient model, which is defined by changes in the phase regime. From a numerical simulation point of view, this requires a reinitialization of the corresponding DAE system, which proceeds according to the same principles as the consistent initialization procedure described previously. Reinitialization thus consists of finding the new values of the algebraic variables at a time instant t^+ that immediately follows the occurrence of the discontinuity, assuming that the state variables are continuous and thus remain at their values at t^-, that is, before the discontinuity has occurred (rather than being set to their initial conditions as is the case in consistent initialization).

We note, however, that the discontinuity does not affect the structure of the algebraic equations of the pseudo-transient model, and it is thus to be expected that the reinitialization procedure will have the same favorable numerical properties as the consistent initialization. We also note that the detection of discontinuities and reinitialization procedures are standard features in most DAE solver packages.

Step 5: Dynamic reformulation of nonlinear constraints

In a (relatively small) number of models of common unit operations, additional nonlinear algebraic constraints are present. Such constraints may involve solving for intermediate temperatures (e.g., the isentropic temperature in a compressor or turbine, or the temperature in an equilibrium-based reactor) that are defined implicitly in the algebraic equations. These situations can be dealt with by defining the variable of interest as a state, whose derivative with respect to pseudo-time is equal to the residual of the nonlinear constraint (this follows the same principle applied to formulating the dynamic energy balance equations above)

$$\tau_T C \frac{dX}{dt} = f_\mathrm{NL}(X, T, \mathbf{M}) \tag{42}$$

where f_NL is the nonlinear algebraic constraint that has not been resolved through the methods mentioned above, C is a coefficient necessary to make the units consistent, and X is the variable that must be determined from the nonlinear constraint equation.

The two examples mentioned in this section will be addressed later.

Flowsheet Simulation

Earlier in the article, the assumption was made that the flow rates of the inlet material and energy streams to each unit are known and can be treated as parameters. When simulating a flowsheet, this is clearly not the case; the outputs of upstream units (which are variables in the unit's model) are the inputs of downstream units. Moreover, when recycle streams are present, the outputs of downstream units become inputs for units located upstream. These interactions between unit operations are reflected in a flowsheet model that consists of highly coupled sets of equations which are difficult to solve. In this section, the process model is considered at the flowsheet level, and we describe how to represent the connections between pseudo-transient units in a manner that ensures that the inputs can be dealt with as parameters during the consistent initialization step of a process flowsheet comprising pseudo-transient unit models.

Step 6: Dynamic tearing

A significant body of work has been dedicated to dealing with recycle streams in SM simulators, resulting in the

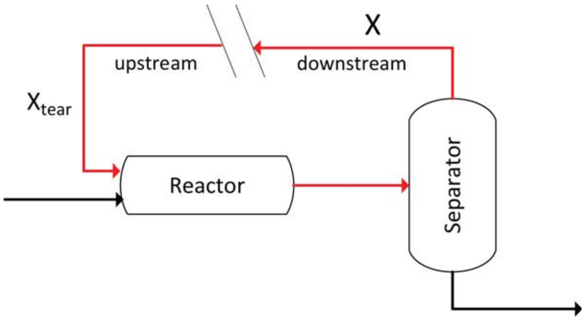

Figure 7. Tear stream representation.

X_{tear} represents the upstream states of the tear stream that are provided initial values, and X represents the downstream states of the tear stream. At the end of the simulation, the properties of the two streams must match. Selecting the time constants for the dynamic variables should follow the natural hierarchy of the process dynamics. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

concept of tear streams. Recycle streams are first "torn" (i.e., the relevant variables, including enthalpy, pressure, flow rate, and composition, are given initial values) and the outputs of individual units in the recycle loop are computed in sequence to determine a new, calculated value for the recycle stream variables (see Figure 7). We refer to the "torn" stream variables as the "upstream" states, and the calculated variables in the loop as the "downstream" states. Successive iterations use the downstream states as the new initial conditions for the upstream variables until the two converge.[5]

Remark 4. *A similar approach is applied to process optimization using algebraic equation oriented flowsheet models, where recycle streams are described by equality constraints between the values of a set of "torn" upstream variables and their corresponding values calculated from downstream units.[2] Implementing this concept in flowsheet optimization is referred to as "infeasible path optimization:" the values of the upstream and downstream variables of the recycle streams may not be the same in all iterations, but the recycle streams must converge at the optimum.[37]*

Using these insights, we replace the tear stream by a pseudo-transient connection whereby the upstream state $\mathbf{X}_{\mathbf{tear}}$ of the tear stream is given initial flow, composition, pressure, and enthalpy conditions so that inlet and outlet states of downstream units can be readily computed during consistent initialization. The upstream states (i.e., the origin of the tear stream) are decoupled from the downstream states by a first-order filter that allows the stream end-point $\mathbf{X}$ to have the same properties as the origin of the tear stream, $\mathbf{X}_{\mathbf{tear}}$, at the end of the time integration

$$\tau_r \frac{d\mathbf{X}_{\mathbf{tear}}}{dt} = -\mathbf{X}_{\mathbf{tear}} + \mathbf{X} \tag{43}$$

where $\mathbf{X}_{\mathbf{tear}}$ are the upstream states of the tear stream (e.g., molar flow rate, enthalpy, pressure, etc.), and $\mathbf{X}$ are the downstream (calculated) values of the respective variables. Notice that this system is in effect error-feedback loop with an integral-only (I) controller with zero setpoint; the I controller is suitable in this case as we are not interested in taking immediate control action (in which case a PI controller would be required), but rather in gradually eliminating the offset between the upstream and downstream values of the torn stream variables.

Step 7: Selection of filter parameters

To facilitate the initialization and integration of pseudo-transient flowsheet models, we proceed by dynamically decoupling the equations of the individual unit operation models. To this end, we notice that, while the time constants of the pseudo-transient models developed above can be chosen arbitrarily, process phenomena occur according to a natural hierarchy of time scales (Figure 8) (see also Refs. 38–40). In the sequel, we proceed with the selection of the filter time constants according to the time scale in which each phenomenon of interest occurs. Broadly speaking, physical phenomena such as the establishment of vapor–liquid equilibrium, mass and heat transfer occur rather quickly; unit operations dynamics evolve over an intermediate time horizon,[38,41] and process wide dynamics are relatively slow.[42–44] Conversely, the rates of chemical reactions can span the spectrum of the aforementioned time scales, and the contribution of chemical reactions to the process model can be adjusted via the parameter continuation strategy outlined above.

This dynamic hierarchy has advantages from a stability perspective as well. The model equations describing a unit operation are agnostic to the rest of the process variables other than the material and energy inlet flows. If stability is established for the individual unit operations with dynamics much faster than that of the process, then the stability of the system will only depend on the slow, process-wide dynamics.[42–45]

For the case studies below, we found that time constants spanning several orders of magnitude tend to have good simulation results. Phase separations evolve at the fastest time scale, the energy balances evolve on a time scale one order of magnitude greater, the parameter integrations and intraunit flow dynamics occur at a time scale two orders of magnitude greater, and process wide dynamics evolve at a time scale three orders of magnitude larger. Thus, the time constants used are $\tau_f = 0.01$ (recall that there are no physical time units associated with the pseudo-transient approach to steady state), $\tau_T = 0.1$, $\tau_p = 1$, $\tau_u = 5$, and $\tau_r = 25$. τ_u is the time constant for flow equilibration between stages in staged vapor–liquid contactors; this will be discussed in subsequent sections.

Remark 5. *It is important to note that there is a tradeoff between the relative difference of the time constants. If the ratio between time scales is large, the dynamic system is stiff and will take longer to integrate to steady state. If the ratio is small, there is less time scale separation, and a greater possibility for instability in the system.*

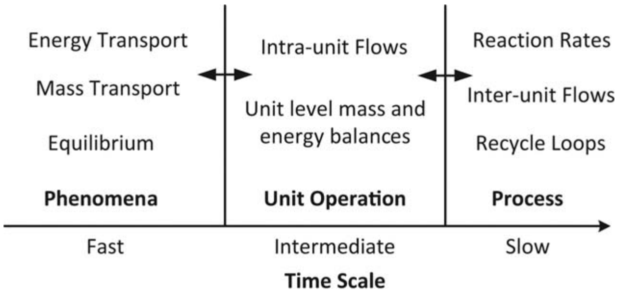

Figure 8. Selecting the time constants for the dynamic variables should follow the natural hierarchy of the process dynamics.

Remark 6. *An inherent problem associated with solving nonlinear systems is that multiple solutions may exist. Although the develpoments described here do not address this problem explicitly, they are meant to be implemented in an optimization framework, and constraints can be imposed to ensure that a physically meaningful steady state is reached.*

Toward a Pseudo-Transient Process Unit Library

The previous sections outlined the concept of pseudo-transient modeling of process units and flowsheets, and provided guidelines for converting the corresponding algebraic process models into a pseudo-transient system that is statically equivalent. Here, the algorithmic considerations introduced earlier in the article are applied to a set of unit operation models that are essential for building a process flowsheet.

Continuously stirred tank reactors

We present the development of a pseudo-transient model for a continuously stirred tank reactor. The exothermic reaction $A \rightarrow B$ that is first order and follows the Arrhenius rate law occurs in the reactor, which is cooled with a jacket kept at constant temperature, T_c. The system has two variables, the concentration, C_A, and the temperature T_r. The equations describing the steady state are

$$0=F(C_{A_0}-C_A)-Vk^0\exp(-E_A/RT_r)C_A \tag{44}$$

$$0=\rho c_p F(T_{r_0}-T_r)-Vk^0\exp(-E_A/RT_r)C_A\Delta H-UA(T_r-T_c) \tag{45}$$

The parameters are listed in Table 1.

We will assume, for illustration purposes, that the physical dynamic model is not known, and we are only interested in solving the steady-state system. We apply the algorithm outlined above to convert the algebraic system to a statically equivalent DAE system.

- Step 1 of the algorithm is to identify the form of the model equations. Clearly, this system falls into the category of general single-phase units.
- Step 2: obtain an explicit dynamic expression for temperature from the energy balance

$$\begin{gathered}\tau_T\frac{dT_r}{dt}=(T_{r_0}-T_r)-\frac{Vk^0}{\rho c_p F}\exp(-E_A/RT_r)C_A\Delta H-\frac{UA}{\rho c_p F}(T_r-T_c)\\ T_r(t=0)=T_r^0\end{gathered} \tag{46}$$

Table 1. CSTR Operating Parameters

Parameter	Description	Value
V	Reactor volume	2 m^3
F	Flow rate	50 l/s
C_{A_0}	Inlet concentration	12 mol/l
k^0	Pre-exponential factor	1000 s^{-1}
E_A	Activation energy	24,900 J/mol
R	Gas constant	8.314 J/mol/K
ρ	Density	1000 kg/m^3
c_p	Heat capacity	2000 J/kg/K
T_{r_0}	Inlet temperature	350 K
T_c	Cooling temperature	280 K
ΔH	Heat of reaction	−50,000 J/mol
U	Heat-transfer coefficient	1000 W/m^2/K
A	Heat transfer area	40 m^2

- Step 3: modulate the contribution of source terms

$$\begin{gathered}0=F(C_{A_0}-C_A)-Vk^0\exp(-E_A/RT_r)C_A\\ \tau_T\frac{dT_r}{dt}=(T_{r_0}-T_r)-\alpha\left[\frac{Vk^0}{\rho c_p F}\exp(-E_A/RT_r)C_A\Delta H-\frac{UA}{\rho c_p F}(T_r-T_c)\right]\\ \tau_p\frac{d\alpha}{dt}=1-\alpha\\ T(t=0)=T_0\\ \alpha(t=0)=0\end{gathered} \tag{47}$$

The system (47) now has three differential equations and one algebraic equation. The consistent initialization step is simple ($C_A(t=0)=C_{A_0}$), and the time integration will proceed until steady state.

- Steps 4, 5, and 6 are not needed in this simple case because there are no multiphase separations, and no recycle streams.
- Step 7 is the selection of the time constants in the process. Here, we use $\tau_T=0.1$ and $\tau_p=1$, the same values suggested earlier.

Results. The pseudo-transient system (Eqs. 47) converges to steady state ($C_A=0.075$ M and $T_r=543$ K) from the full set of initial conditions provided ($T_r=[300,700]$ K, and $k^0=UA=0$). However, Newton's method only converges to the solution when the initial guess is close to the solution. Figure 9 shows in black the set of initial guesses that solve to the steady-state solution. The white region diverges from the solution.

The result suggests that, even with a relatively simple problem with only two variables, a transient approach to the steady-state solution is more robust than finding the roots via Newton's method when the initial guess is not close to the solution. We will show later that this benefit extends to more complex process models.

Multistage vapor–liquid contactors

In staged equilibrium units like distillation, absorption, and stripping columns, the input material and energy flows to each equilibrium stage are the model output variables of the adjacent stages. Each stage is solved with a flash calculation, but due to the connectivity, the resulting

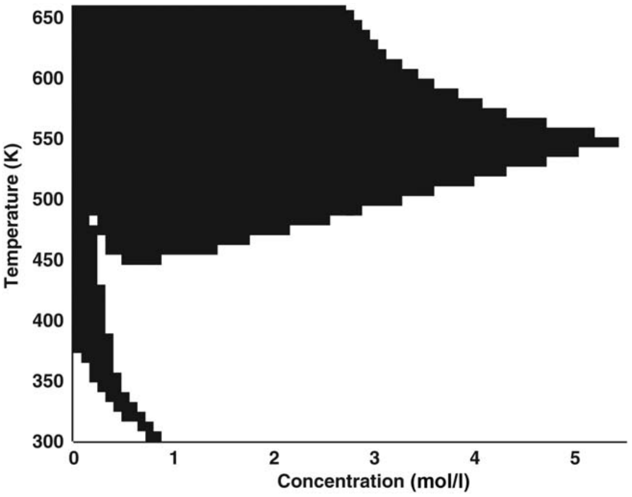

Figure 9. Basin of convergence for Newton's Method.

The solution is at $C_A=0.075$ M and $T_r=543$ K.

model is highly coupled, and typically difficult to solve.[10] The equations describing material exchanges between stages are

$$\begin{aligned} 0&=M_{i,\text{in},j}+L_{j+1}x_{i,j+1}+V_{j-1}y_{i,j-1}-V_jy_{i,j}-L_jx_{i,j} \\ 0&=H_{\text{in},j}+H^L_{j+1}+H^V_{j-1}-H^V_j-H^L_j \end{aligned} \tag{48}$$

where index j refers to the stage number, and $M_{i,\text{in},j}$ and $H_{\text{in},j}$ are nonzero only at the feed stage. The molar flow to the stage, L_{j+1} and V_{j-1} and their corresponding compositions, $x_{i,j+1}$ and $y_{i,j-1}$ have enthalpy flows of H^L_{j+1} and H^V_{j-1}.

The equations can be separated into two steps (note that this is in a way similar to the steps taken in the Ponchon–Savarit method for column design[46]), a mixing step and an equilibrium step. The first step consists of mixing the material streams and associated energy contents from the two adjacent trays (and the column feed if present)

$$\begin{aligned} M^F_{i,j}&=M_{i,\text{in},j}+L_{j+1}x_{i,j+1}+V_{j-1}y_{i,j-1} \\ H^F_j&=H_{\text{in},j}+H^L_{j+1}+H^V_{j-1} \end{aligned} \tag{49}$$

The second step is an equilibration step that consists of splitting the total material and energy of the stage between the vapor and liquid phases

$$\begin{aligned} M^F_{i,j}&=V_jy_{i,j}+L_jx_{i,j} \\ H^F_j&=H^V_j+H^L_j \end{aligned} \tag{50}$$

Such gas liquid contactors can thus be regarded as a flowsheet comprising multiple single-stage equilibrium units. Thus, in a manner similar to the tearing strategy discussed earlier for flowsheet simulation, the vapor–liquid equilibrium calculations on each stage can be decoupled at the consistent initialization step by converting the combined molar feed rate, $M^F_{i,j}$, and the combined energy feed rate H^F_j into dynamic variables

$$\begin{aligned} \tau_u\frac{dM^F_{i,j}}{dt}&=-M^F_{i,j}+M_{i,\text{in},j}+L_{j+1}x_{i,j+1}+V_{j-1}y_{i,j-1} \\ \tau_u\frac{dH^F_j}{dt}&=-H^F_j+H_{\text{in},j}+H^L_{j+1}+H^V_{j-1} \end{aligned} \tag{51}$$

Equation 51 is simply a dynamic expression of the mixing step given by Eq. 49. These dynamic mixing equations, along with the algebraic equilibration step given by Eq. 50 define the pseudo-transient multistage equilibrium flowsheet model.

For consistently initializing the resulting DAE system, the combined mass and energy feeds to each stage ($M^F_{i,j}$ and H^F_j) are regarded as parameters, and the flash calculations for each stage are solved via the procedure described earlier. To further simplify the problem, the initial conditions for the states of each stage can be assumed to be equivalent to the properties of the combined material and energy feeds entering the column. Notice that the dynamics are similar to that of a mixing tank where the flow from the adjacent trays is mixed before entering the stage. When the time derivatives in (51) are zero (at steady state), the system reduces to the original set of Eqs. 48. Figure 10 compares the structures of the steady-state column model and the new pseudo-transient column model. The hypothetical "tanks" adjacent to the stages are assumed to be well-mixed. Similar to the pseudo-transient tear stream concept proposed earlier, the streams flowing from the "tanks" to the stages are "torn." Initially, the variables of the streams flowing to the stages are provided, and over time these gradually converge to the combined flows from the adjacent stages.

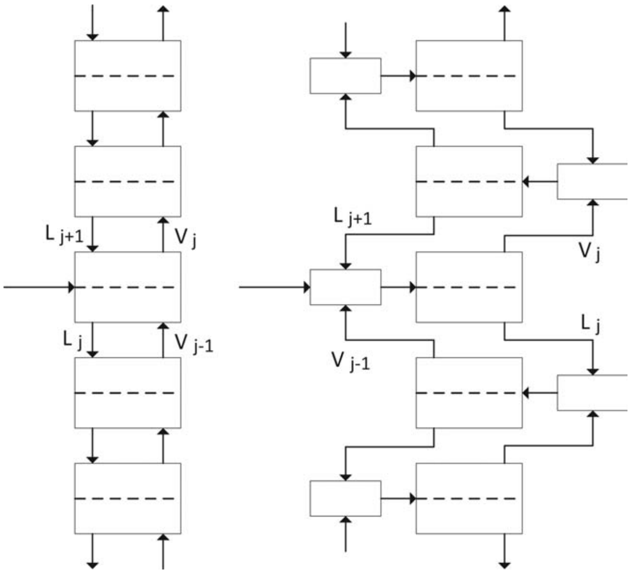

Figure 10. Comparison of typical column flowsheet (left) with pseudo-transient column flowsheet (right).

Two-stream heat exchanger

The models of countercurrent and cocurrent two-stream heat exchangers are shown in Figure 11.

- Step 1 is to determine the type of unit operation. Here, we assume that the streams do not undergo phase change within the unit. Thus, each stream can be treated as a general single phase unit. Consider the mass and energy balance equations of both streams simultaneously

$$\begin{aligned} 0&=M^{\text{h}}_{i,\text{in}}-M^{\text{h}}_{i,\text{out}} \\ 0&=M^{\text{c}}_{i,\text{in}}-M^{\text{c}}_{i,\text{out}} \\ 0&=H^{\text{h}}_{\text{in}}-H^{\text{h}}_{\text{out}}-UA\Delta T \\ 0&=H^{\text{c}}_{\text{in}}-H^{\text{c}}_{\text{out}}+UA\Delta T \end{aligned} \tag{52}$$

where the superscripts h and c refer to the hot and cold streams, respectively. A typical approximation for the temperature driving force for such heat exchangers is the log-mean temperature difference, which is given by

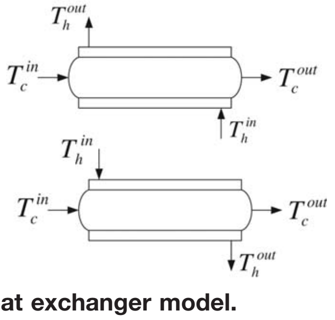

Figure 11. Heat exchanger model.

A countercurrent flow arrangement is represented on top and a cocurrent arrangement on the bottom.

$$\Delta T = \frac{(T^{\mathrm{h}}_{\mathrm{in}} - T^{\mathrm{c}}_{\mathrm{out}}) - (T^{\mathrm{h}}_{\mathrm{out}} - T^{\mathrm{c}}_{\mathrm{in}})}{\ln\left(\frac{(T^{\mathrm{h}}_{\mathrm{in}} - T^{\mathrm{c}}_{\mathrm{out}})}{(T^{\mathrm{h}}_{\mathrm{out}} - T^{\mathrm{c}}_{\mathrm{in}})}\right)} \tag{53}$$

for the countercurrent flow arrangement, and

$$\Delta T = \frac{(T^{\mathrm{h}}_{\mathrm{out}} - T^{\mathrm{c}}_{\mathrm{out}}) - (T^{\mathrm{h}}_{\mathrm{in}} - T^{\mathrm{c}}_{\mathrm{in}})}{\ln\left(\frac{(T^{\mathrm{h}}_{\mathrm{out}} - T^{\mathrm{c}}_{\mathrm{out}})}{(T^{\mathrm{h}}_{\mathrm{in}} - T^{\mathrm{c}}_{\mathrm{in}})}\right)} \tag{54}$$

for the cocurrent flow arrangement.

T^{h} and T^{c} are the hot and cold stream temperatures, respectively. The set of Eqs. 52 is required to calculate the exit mass and energy flows from both streams. There are no generation terms in the material balance equations, so the outlet molar flow rates are equal to the inlets.

- Step 2 of the pseudo-transient restructure approach is to make the energy balance equations dynamic, and
- Step 3: introduce an additional state to modulate source terms. The resulting DAE system is

$$\begin{aligned}
0 &= M^{\mathrm{h}}_{i,\mathrm{in}} - M^{\mathrm{h}}_{i,\mathrm{out}} \\
0 &= M^{\mathrm{c}}_{i,\mathrm{in}} - M^{\mathrm{c}}_{i,\mathrm{out}} \\
\left(\tau_T \frac{H^{\mathrm{h}}_{\mathrm{in}}}{T^{\mathrm{h}}_0}\right)\frac{dT^{\mathrm{h}}_{\mathrm{out}}}{dt} &= H^{\mathrm{h}}_{\mathrm{in}} - H^{\mathrm{h}}_{\mathrm{out}} - \alpha[UA\Delta T] \\
\left(\tau_T \frac{H^{\mathrm{c}}_{\mathrm{in}}}{T^{\mathrm{c}}_0}\right)\frac{dT^{\mathrm{c}}_{\mathrm{out}}}{dt} &= H^{\mathrm{c}}_{\mathrm{in}} - H^{\mathrm{c}}_{\mathrm{out}} + \alpha[UA\Delta T] \\
\tau_p \frac{d\alpha}{dt} &= 1-\alpha \\
T^{\mathrm{h}}_{\mathrm{out}}(t=0) &= T^{\mathrm{h}}_0 \\
T^{\mathrm{c}}_{\mathrm{out}}(t=0) &= T^{\mathrm{c}}_0 \\
\alpha(t=0) &= 0
\end{aligned} \tag{55}$$

Note that, at steady state, the Eqs. 55 are equivalent to (52).

Compression and turbine expansion

- Step 1 is to select the type of unit operation. Compressors and turbines are pressure change units with the following balance equations

$$\begin{aligned}
0 &= M_{i,\mathrm{in}} - M_{i,\mathrm{out}} \\
0 &= (H_{\mathrm{in}} - H_{\mathrm{out}}) + (\dot{W} - Q_{\mathrm{ext}})
\end{aligned} \tag{56}$$

where $\dot{W}$ is the work added to the system, and Q_{ext} includes inefficiency losses. Assuming the pressure at the inlet and outlet are design parameters, the equations are used to solve for the exit material and energy flows (and temperature). Obviously, the material flows are simply equal to the inlet flows. The efficiency of a compressor is defined as

$$\eta_{\mathrm{c}} = \frac{H^{\mathrm{ise}}_{\mathrm{out}} - H_{\mathrm{in}}}{H_{\mathrm{out}} - H_{\mathrm{in}}} \tag{57}$$

while the efficiency of a turbine is defined as

$$\eta_{\mathrm{t}} = \frac{H_{\mathrm{in}} - H_{\mathrm{out}}}{H_{\mathrm{in}} - H^{\mathrm{ise}}_{\mathrm{out}}} \tag{58}$$

where $H^{\mathrm{ise}}_{\mathrm{out}}$ refers to the enthalpy if the process was isentropic. These can be rearranged to obtain an expression for the *FLUX* terms

$$(\dot{W} - Q_{\mathrm{ext}}) = \frac{1}{\eta_{\mathrm{c}}}(H^{\mathrm{ise}}_{\mathrm{out}} - H_{\mathrm{in}}) \tag{59}$$

likewise, for turbines

$$(\dot{W} - Q_{\mathrm{ext}}) = \eta_{\mathrm{t}}(H^{\mathrm{ise}}_{\mathrm{out}} - H_{\mathrm{in}}) \tag{60}$$

- Step 2 is to make the energy balance dynamic, and
- Step 3 is to introduce additional state variables to modulate source terms

$$\begin{aligned}
0 &= M_{i,\mathrm{in}} - M_{i,\mathrm{out}} \\
\left(\tau_T \frac{H_{\mathrm{in}}}{T_0}\right)\frac{dT_{\mathrm{out}}}{dt} &= H_{\mathrm{in}} - H_{\mathrm{out}} + \frac{\alpha}{\eta_c}(H^{\mathrm{ise}}_{\mathrm{out}} - H_{\mathrm{in}}) \\
\tau_p \frac{d\alpha}{dt} &= 1-\alpha \\
T_{\mathrm{out}}(t=0) &= T_0 \\
\alpha(t=0) &= 0
\end{aligned} \tag{61}$$

In the turbine case, the equations are equivalent, but the continuation parameter multiplies α (i.e., $\alpha\eta_{\mathrm{t}}$ replaces $\frac{\alpha}{\eta_{\mathrm{c}}}$).

- Step 4 is not necessary as the system is only in the gas phase.
- Step 5 is necessary in this case, because we need to calculate $H^{\mathrm{ise}}_{\mathrm{out}}$. We can obtain this value by computing the temperature ($T^{\mathrm{ise}}_{\mathrm{out}}$) assuming an isentropic process

$$S_{\mathrm{in}} = S(T_{\mathrm{in}}, P_{\mathrm{in}}, \mathbf{M}) = S(T^{\mathrm{ise}}_{\mathrm{out}}, P_{\mathrm{out}}, \mathbf{M}) = S^{\mathrm{ise}}_{\mathrm{out}} \tag{62}$$

which is a nonlinear, implicit function of the isentropic temperature. A dynamic filter on the isentropic outlet temperature can be implemented by equating the entropy at the inlet and outlet

$$\begin{aligned}
0 &= M_{i,\mathrm{in}} - M_{i,\mathrm{out}} \\
\left(\tau_T \frac{H_{\mathrm{in}}}{T_0}\right)\frac{dT_{\mathrm{out}}}{dt} &= H_{\mathrm{in}} - H_{\mathrm{out}} + \frac{\alpha}{\eta_{\mathrm{c}}}(H^{\mathrm{ise}}_{\mathrm{out}} - H_{\mathrm{in}}) \\
\left(\tau_T \frac{S_{\mathrm{in}}}{T_0}\right)\frac{dT^{\mathrm{ise}}_{\mathrm{out}}}{dt} &= S_{\mathrm{in}} - S^{\mathrm{ise}}_{\mathrm{out}} \\
\tau_p \frac{d\alpha}{dt} &= 1-\alpha \\
T_{\mathrm{out}}(t=0) &= T_0 \\
T^{\mathrm{ise}}_{\mathrm{out}}(t=0) &= T_0 \\
\alpha(t=0) &= 0
\end{aligned} \tag{63}$$

Now, $H^{\mathrm{ise}}_{\mathrm{out}}$ can be computed explicitly if $T^{\mathrm{ise}}_{\mathrm{out}}$ is known. The DAE system has a steady-state solution equivalent to the original algebraic system.

Equilibrium reactor

Often, chemical reactors are designed so that the reaction mixture reaches equilibrium before exiting the unit. The model equations of an equilibrium reactor differ from a conventional (lumped parameter) reactor only in the expression of the reaction rates. In conventional reactors, the reaction rates have explicit expressions (e.g., Arrhenius rate law or Michaelis–Menten kinetics); in equilibrium reactors, the rates of reaction are specified by a constraint that defines the equilibrium concentrations[47]

$$K_a = \prod_i a_i^{v_i} = \exp\left(\frac{-\Delta G^{\circ}_{\text{rxn}}}{RT}\right) \tag{64}$$

where a_i is the chemical activity of component i that is computed explicitly from the temperature and composition, and v_i is the stoichiometric coefficient for each component in the reaction. $\Delta G^{\circ}_{\text{rxn}}$ is the Gibbs free energy change of the reaction, and R is the universal gas constant. If the equilibrium conversion (or compositions) are specified, Eq. 64 must be used to solve for the equilibrium temperature. If the reaction is exothermic, K_a will be a globally decreasing function of temperature, and if the reaction is endothermic, K_a will increase globally with temperature.

The mass and energy balance equations are given by

$$\begin{aligned} 0 &= (M_{i,\text{in}} - M_{i,\text{out}}) + r_i \\ 0 &= (H_{\text{in}} - H_{\text{out}}) + H_{\text{gen}} + Q_{\text{ext}} \end{aligned} \tag{65}$$

and the reaction rates are computed explicitly from conversion

$$r_i = X \frac{v_i}{v_{\text{key}}} \tag{66}$$

where v_i / v_{key} is the ratio of the stoichiometric coefficients of the products to the stoichiometric coefficient of the limiting, key reactant, and X is the specified conversion of the key component. Equations 64–66 define the reactor model, and it is necessary to compute the equilibrium temperature and the required external heat applied, Q_{ext}. The required heat rate Q_{ext} must be computed from the energy balance, and the temperature must be computed from the equilibrium relationship.

To obtain an explicit dynamic expression for temperature, we first take the logarithm of either side of the equilibrium constraint, then introduce a dynamic filter

$$\begin{aligned} \text{sign}(\Delta G^{\circ}_{\text{rxn}}) \left(\frac{\tau_T}{T_0}\right) \frac{dT}{dt} &= \Sigma_i v_i \ln(a_i) + \frac{\Delta G^{\circ}_{\text{rxn}}}{RT} \\ T(t=0) &= T_0 \end{aligned} \tag{67}$$

where $\text{sign}(\Delta G^{\circ}_{\text{rxn}})$ is the signum function that ensures the equation is stable as it changes sign if the reaction is exothermic or endothermic. Notice that either side of the equation is dimensionless. Equations 65–67 define the pseudo-transient system model. Consistent initialization consists of solving a linear system for the equilibrium compositions, and the steady state is equivalent to the solution of the original algebraic system of equations.

Process Optimization

The optimization of process flowsheets entails identifying the values of the stream flow rates and unit operating conditions (e.g., pressures and temperatures), as well as unit sizes (e.g., number of transfer units and number of stages) that lead to a minimal operating and capital cost. Evidently, this problem is an ideal candidate for an optimization-based solution approach. In many industries, flowsheet optimization remains to this day a laborious trial-and-error activity that requires considerable engineering skill to complete; the use of systematic optimization methods using detailed process models is limited in scope[48,49] and has not gained widespread acceptance.

From a mathematical perspective, complete plant-wide optimization has been hindered by the difficulty of solving the equations corresponding to detailed plant models. At each step of an optimization calculation, the solution corresponding to the new values of the decision variables can be quite far from the solution at the previous optimization iteration; as such, using the previous solution as an initial guess for the new solution can result in a numerical failure (see Biegler,[2] page 185); this is especially true when the process is tightly integrated with significant material and energy recycling and recovery (thus, the need for following the infeasible path approach). The framework established in this article provides a robust pathway for identifying the steady-state solution of a detailed, first-principles plant model in an EO framework where the optimization solver can follow a feasible solution path (i.e., the flowsheet model is solved at each optimization iteration). In this section, we integrate this framework with a time-relaxation-based optimization algorithm that has been developed and refined in our previous work[50–53] (see also a recent extension[54] to uncertain systems).

The algorithm is itself centered on the concept of static equivalence and uses dynamic models to obtain a solution that is optimal for the steady-state system. The algorithm, presented in Figure 12, consists of (1) providing appropriate initial conditions for the statically equivalent dynamic system and initial guesses for the optimization variables (which are typically operating parameters in the process), followed by (2) simulating the system over the time horizon T to steady state, without enforcing any path constraints (design constraints are enforced as end-point constraints). (3) At steady state, the objective function as well as the gradients of the objective and constraints are computed and (4) new decision variables are calculated. Then, (5), the system states at $t = T$ are used as new initial conditions, and steps (2)–(5) are repeated until an optimality criterion is satisfied.

Remark 7. *The proposed algorithm can be interpreted from the perspective of the control vector parametrization methods used to solve dynamic optimization problems,[55] in which a nonlinear programming (NLP) solver works in tandem with a DAE integrator. The differences are that in the present case there are no control vectors (i.e., the decision variables are time-invariant) and no path constraints. Figure 13 illustrates this point.*

After the initial optimization iteration, in order to further facilitate time integration, a filter is applied to the optimization decision variables in a manner exactly equivalent to the parameter continuation approach described earlier. The initial conditions for the state variables for the next optimization iteration are equivalent to their values at the end of the previous iteration, and the new values of the decision variables computed by the optimization algorithm are reached at the end of the time horizon. Figure 13 provides a graphical depiction of the solution principle.

We note that a similar concept has been used to optimize complex process units under uncertain operating conditions.[56] In this work, the authors used a transient variable to sample the uncertainty space (without modifying the steady-state model equations) during a dynamic optimization iteration. This simplified the simulation of the process unit for each disturbance realization by eliminating the initialization steps normally required at each optimization iteration.

Case Studies

The pseudo-transient models described above were implemented in gPROMS.[57] In the following sections, we

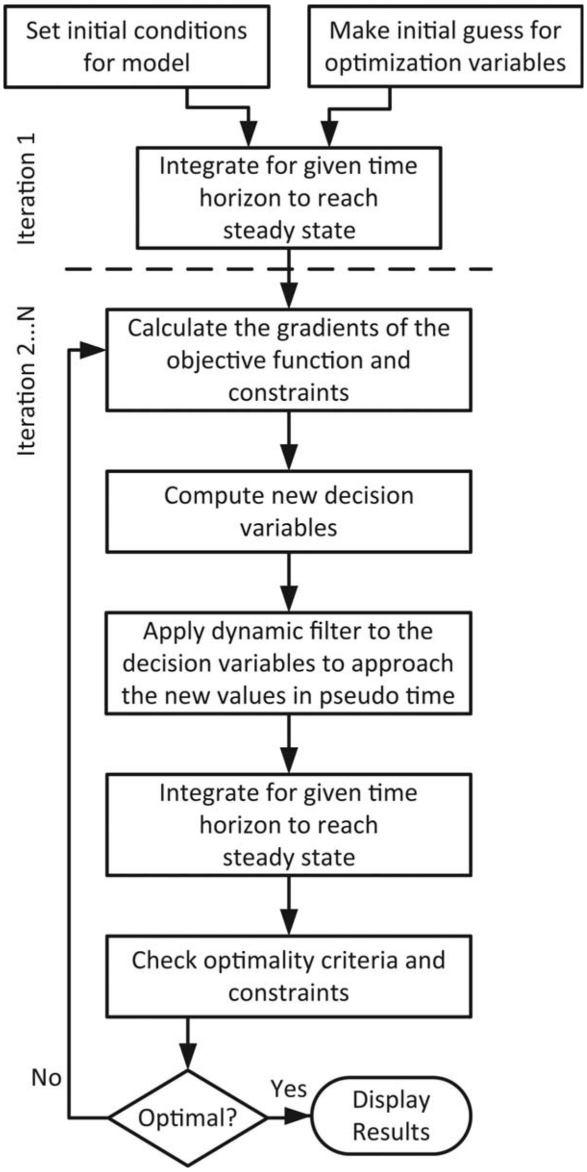
Set initial conditions for model
Make initial guess for optimization variables
Iteration 1
Integrate for given time horizon to reach steady state
Iteration 2...N
Calculate the gradients of the objective function and constraints
Compute new decision variables
Apply dynamic filter to the decision variables to approach the new values in pseudo time
Integrate for given time horizon to reach steady state
Check optimality criteria and constraints
No
Optimal?
Yes
Display Results

Figure 12. Steady-state optimization algorithm.

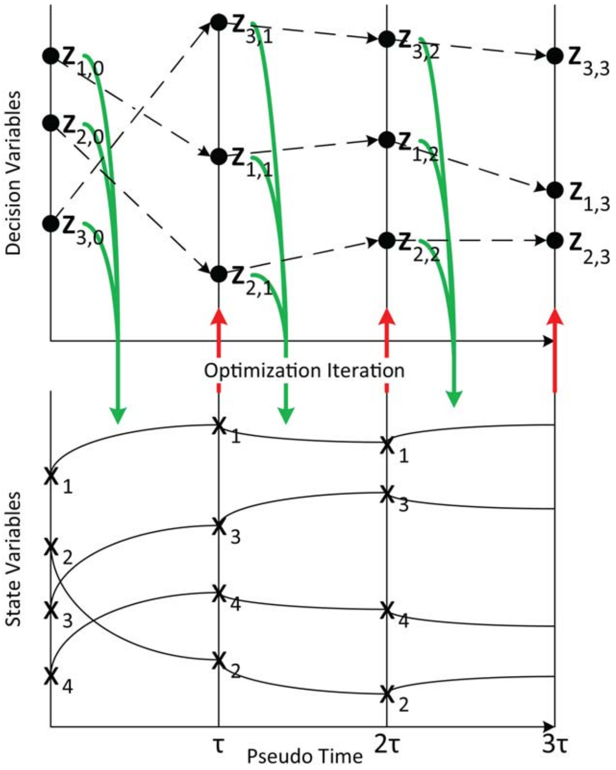

Figure 13. The top chart represents the value of the decision variables (z) at each optimization iteration. The bottom chart represents the evolution of the state variables (x) throughout the optimization.

The leftmost set of variables in the top chart represent the initial guesses for the optimization variables, while the leftmost set of state variables represent the set of initial conditions for the first optimization iteration. After the system has reached steady state in the first optimization iteration, the system does not reset with the original initial conditions, rather, the steady-state values from the previous optimization iteration are used as the initial condition to transition to the next steady state. This greatly reduces the simulation time between each optimization iteration. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

demonstrate the use of the entire framework introduced above in modeling and optimizing two prototype integrated process systems, ammonia synthesis and ethanol production.

Case study 1: Ammonia synthesis

The ammonia synthesis process is a basic continuous chemical process that converts hydrogen and nitrogen to ammonia by the following reaction

$$N_2+3H_2 \rightarrow 2NH_3 \tag{68}$$

The process (as described in Chapter 7 of Nonlinear Programming by Biegler[2]) is shown in Figure 1 (in the section titled Motivating Examples) with data given in Table 2. The reaction occurs at high temperature and pressure in a catalytic reactor. The effluent of the reactor, S7, is cooled and flashed to separate the ammonia (S11) from the hydrogen and nitrogen (S10). The reactants are purged (S12), recycled, and recompressed (S14), and the ammonia-rich stream (S11) is then flashed again at low pressure to further separate the product (S16). The recycled reactants (S15) are combined with the feed stream and undergo two-stage compression with intercooling (S1–S4). The stream is mixed with the vapor from the first flash and heated to the reaction temperature (S6). The reaction is assumed to be at equilibrium, with a fixed conversion of hydrogen. Computations are carried out using the Redlich–Kwong–Soave equation of state.

Simulation of this process in a SM environment is straightforward and likely to converge from a distant initial guess. However, due to the high recycle ratio and nonlinearities associated with the equation of state and the flash calculations, simulation in conventional EO environments requires a close initial guess to ensure convergence. By contrast, we verified that the pseudo-transient approach proposed in this article leads to a fast and robust solution of the EO model.

The pseudo-transient model equations were taken from the model library presented earlier in this article. The time constants for the pseudo-transient system were selected based on the values given in Section Step 7: Selection of Filter Parameters.

Table 2. Ammonia Synthesis Process

Variable	Initial	Lower	Upper	Optimal	
Pressure of S2 (MPa)	4.413	4.137	6.895	4.137	Lower Bnd.
Pressure of S4 (MPa)	20.68	19.99	29.72	20.05	
Reactor conversion	0.41	0.35	0.45	0.45	Upper Bnd.
Temperature of S8 (K)	299.8	295.9	310.9	310.9	Upper Bnd.
Temperature of S9 (K)	242.0	233.1	333.1	288.3	
Pressure of S10, S11 (MPa)	19.99	19.65	29.41	19.65	Lower Bnd.
Split ratio (S12/S13)	0.1	0.05	0.12	0.05	Lower Bnd.
Tear stream (S9)					Feed
Hydrogen (mol/s)	10			458.4	110.0
Nitrogen (mol/s)	10			63.43	35.53
Argon (mol/s)	10			69.65	0.89
Methane (mol/s)	10			110.7	1.63
Ammonia (mol/s)	10			328.3	0.0
Temperature (K)	242.0			288.3	299.82
Pressure (MPa)	20.68			20.05	1.013

To analyze the computational efficiency in computing the steady-state solution, we observe how rapidly the residual (the sum of the 2-norms of all the pseudo-time derivatives) approaches zero. In Figure 14, both the residual (left axis, dashed line) and the pseudo-time step length (right axis, solid line) are plotted against the DAE integration step during the initial optimization iteration. Remember that if the time step is infinite, the pseudo-transient method is equivalent to Newton's method (which will most likely not converge), but taking intermediate small time steps allows for a smooth approach to the steady-state solution. The DAE solver[57] automatically adjusts the time step in order to preserve stability and accuracy. At each integration time step, a nonlinear system must be solved once, so the number of integration steps is proportional to the computational time required to solve the system. In this example, 450 integration time steps were required to reach steady state which took approximately 2.15 s on an Intel Core i7 processor at 3.40 MHz. The pseudo-time integration horizon required to reach steady state was $\tau=5000$. Note that as the residual decreases, the time steps lengthen, and the residual rate of convergence accelerates.

The optimal set of operating conditions is computed based on an objective function that maximizes profit, consisting of the revenue generated from ammonia sales (S16) and the cost of the utilities (heat, compression, refrigeration, and cooling water). Constraints are enforced to ensure that the product stream is at least 99% ammonia, the purge stream (S12) contains no more than 3.4 mol/s ammonia, and the pressure drop in the first flash must be at least 0.4 MPa. There are seven optimization variables corresponding to the purge fraction (S13/S10), the pressure of S10 and S11, the

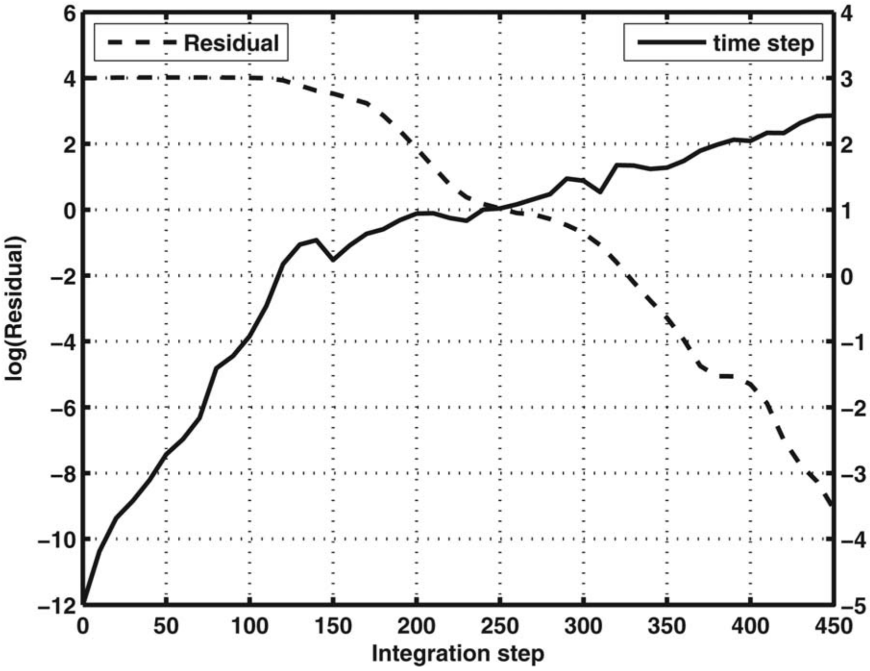

Figure 14. Log of the residual calculation (left axis, dashed line) and the pseudo-time step length (*dt*) (right axis, solid line) at each DAE integration time step during the initial optimization iteration.

The DAE integration time steps along the bottom axis of this graph correspond to one evolution of the state variables in pseudo-time at the bottom of Figure 13. When the residual is near zero, the state variables are at steady state. The cumulative sum of the time step length curve is equal to the pseudo-time integration horizon.

pressure of S2 (P2), the pressure in the reactor (P4), the conversion of hydrogen in the reactor (X), and the temperatures of S8 and S9 (T8 and T9, respectively).

The simulation is carried out for significantly longer than the longest time constant for each optimization iteration to ensure that steady state is reached in each optimization step. The optimal point is also presented in Table 2, and was found in 118 s and the profit function increases by 33.7% from the initial guess (as provided in the aforementioned text) to the optimal point.

Case study 2: Ethanol synthesis

The process for converting ethylene to ethanol follows the classical reaction-separation-recycle structure and is discussed in the process design textbook, "Systematic Methods of Chemical Process Design," by Biegler et al.[5] The chemical reactions occur at high pressure (68 atm) and are assumed to be at equilibrium and conversion is low (only about 6% of the ethylene reacts to form ethanol) and 10% of the ethanol formed reacts to form diethyl ether.

$$\begin{gathered} CH_2{=}CH_2+H_2O \rightarrow CH_3CH_2OH \\ 2CH_3CH_2OH \rightarrow C_2H_5OC_2H_5+H_2O \end{gathered} \tag{69}$$

The process diagram is shown in Figure 2 (in the Motivating Examples Section above). The low conversion of ethylene in the reactor calls for a high ethylene recycle rate. The gaseous reaction products (S3) are cooled into the two-phase region and flashed to separate some of the water and ethanol (S9) from the ethylene and ether (S5). This is followed by an absorber (modeled by 10 equilibrium stages) to extract the excess ethanol from the ethylene recycle stream using water (S11) as the absorbent. The ethanol and water (S14) are fed to a distillation column operating at lower pressure (2 atm) with 20 stages with the feed on Stage 8 (from the top). The distillate (S19) is mainly ethanol and water (at about 55/45 molar ratio) to be further separated, and the bottoms (S18) is mostly water of which a majority is recycled to the absorber. The Redlich–Kwong–Soave equation of state was used for modeling the physical properties, and the model has 1341 variables.

The pseudo-transient model equations were taken from the model library described earlier. Similar to the previous example, the time constants for the simulation were selected as 0.01, 0.1, 1, and 5 for τ_f, τ_T, τ_p, and τ_u, respectively. Finally, the tear streams, *S*2 and *S*17, were represented using pseudo-transient connections with time constants of $\tau_r=25$ (Eq. 43). The values of the time constants are consistent with the values in the previous case study, suggesting that the proposed formulation can potentially be generalized.

The system model is more complicated and significantly higher-dimensional than the previous example due to the staged equilibrium units. We have attempted simulating the process in a premier commercial SM modeling environment and found that we were not able to converge the model without resorting to a (manual) parameter continuation approach. Specifically, we gradually increased the flow rates of the recycle streams from zero to the actual values. Likewise, we gradually increased the number of equilibrium stages in the absorber and distillation column. Conversely, we verified that the pseudo-transient flowsheet model was very easily initialized even from a poor set of initial conditions for the dynamic variables. To this latter point, Figure 15 shows the residual (the sum of the 2-norms of all time derivatives) and the pseudo-time step lengths vs. the DAE integration step number during the first optimization iteration. Notice the ability of the method to climb uphill (increase the residual) as it approaches the solution; in conventional nonlinear solvers this often (but

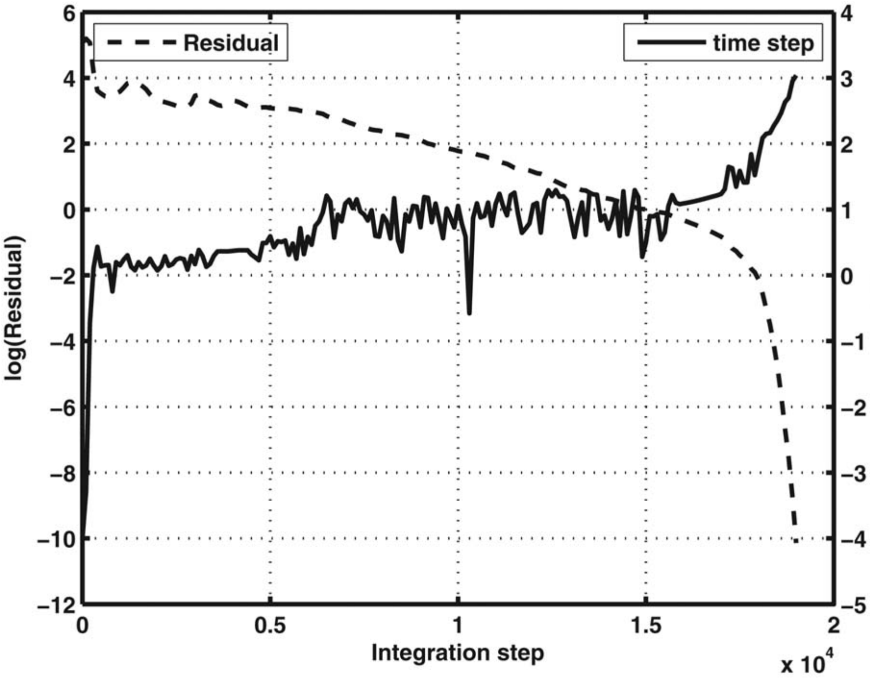

Figure 15. Log of the residual calculation (left axis, dashed line) and the pseudo-time step length (*dt*) (right axis, solid line) at each DAE integration time step during the initial optimization iteration.

The DAE integration time steps along the bottom axis of this graph correspond to one evolution of the state variables in pseudo-time at the bottom of Figure 13. When the residual is near zero, the state variables are at steady state. The cumulative sum of the time step length curve gives the pseudo-time integration horizon.

Table 3. Ethylene-to-Ethanol Process Variables

Variable	Initial	Lower	Upper	Optimal
Condenser temperature (K)	370	320.0	385.0	366.7
Reboiler duty (MW)	2.5	0.5	6.0	0.728
Split 1 ratio (S13/S6)	0.05	0.01	0.2	0.0345
Split 2 ratio (S18/S15)	0.25	0.01	0.9	0.152
Ethylene/water feed ratio	0.176	0.111	9	0.182
Pressure of S5, S9 (MPa)	5.500	5.0	6.8	5.906
Temperature of S4 (K)	430.0	340	480	425.1
Temperature of S17 (K)	350.0	340	420	411.9
Excess water feed (mol/s)	120	50	500	143.2
Objective ($Millions)	1014			836
Capital cost ($Millions)	7.72			6.06
Raw materials ($Millions per yr)	172			156
Utilities ($Millions per yr)	93.3			62.6
Tear stream (S2)				
Water (mol/s)	30			98.06
Ethylene (mol/s)	40			200.5
Diethyl Ether (mol/s)	1			15.82
Ethanol (mol/s)	1			2.63
Temperature (K)	400			419.1
Pressure (MPa)	6.8			6.8
Tear Stream (S17)				
Water (mol/s)	30			2049
Ethylene (mol/s)	1			0
Diethyl ether (mol/s)	1			0
Ethanol (mol/s)	1			1.004
Temperature (K)	310			411.9
Pressure (MPa)	5.48			5.87

not always) indicates divergence from the solution. Also, intuitively, as the residual is decreased, the time steps become larger and the convergence accelerates, that is, the closer the system is to steady state, the more rapidly the simulation approaches steady state.[26] This is beneficial from an optimization perspective because as the parameters of the system change between optimization iterations, the system has a relatively small residual compared to the start of the initial iteration, and rapidly converges to the next steady state.

In this example, 19,000 integration time steps were required to reach steady state during the initial optimization iteration. This took 187.5 s on an Intel Core i7 processor at 3.40 MHz. However, in subsequent optimization iterations, the number of integration time steps drastically reduces because the initial residuals are much smaller. The pseudo-time integration horizon required to reach steady state was $\tau = 150,000$.

Focusing now on design optimization, the process features nine decision variables (see Table 3). Constraints are placed on the purity of the distillate (must be greater than 55% ethanol) and the mole fraction of diethyl ether that is retained within the process (the diethyl ether mole fraction must be less than 5% in the reactor). The objective function to be minimized is the detailed net present cost calculation of the process (accounting for capital, utilities, raw materials, labor, depreciation, etc.) with the parameters listed in Table 4 and equipment sizing and costing details from Biegler et al.[5] The optimal point was found in 646 s and resulted in a 17.6% improvement in the objective function.

Table 4. EtOH Synthesis Process Parameters

Parameter	Value
Ethanol production	119 Million kg/yr
Electricity cost	$0.10/kWh
Heat cost (steam)	$0.093/kWh
Ethylene cost	$0.52/lb
Project life	15 years
Tax life	12 years
Rate of return	0.1
Tax rate	50%
Depreciation	Straight line

Remark 8. *Figure* 15 *reveals the effect of the time scale separation imposed through our choice of time constants for the pseudo-transient models used in the flowsheet. The small time steps taken at the beginning of the time horizon are needed to resolve the fastest dynamics, and are several orders of magnitude smaller than the time steps taken at the end, when the dynamics associated with the tear streams are accounted for. The significant computational effort (evinced by frequent variations in integration step) expended in the intermediate time scale likely corresponds to the state variables of the column models reaching their quasi steady-state values. Similar findings have been reported in previous works*[29] *that used PTC to obtain the steady-state simulation of otherwise multiple time scale systems.*

Conclusions

The pseudo-transient approach to steady-state process simulation presented in this article provides a robust means to simulate chemical processes following the equation oriented paradigm. We began by describing a generic model reformulation strategy based on the principle of static equivalence. Specifically, we prescribed converting the steady-state models of process units into sets of index-1 DAEs based on (1) preserving the same steady-state solution and (2) simple consistent initialization of the algebraic variables irrespective of the initial values provided for the differential variables. We used these principles to construct a library of pseudo-transient models for the most common process unit operations; furthermore, we provided a transient interpretation for stream tearing, an essential concept for ensuring the solvability of process flowsheet models.

We then formalized the selection of the tuning parameters for the dynamic component of the pseudo-transient models based on the natural hierarchy of the dynamics of process phenomena. Further, we discussed the stability of the resulting models. Proving stability properties in the general case will be the subject of our future research.

Another key development was to show that pseudo-transient models lend themselves naturally to use in flowsheet optimization calculations, and we demonstrated that this can be achieved by using a previously developed time relaxation-based optimization algorithm. Finally, we presented a validation of the proposed concepts using two case studies, an ammonia synthesis process and an ethylene-to-ethanol process. Here, we showed that the proposed solution approach exhibits excellent convergence properties, and demonstrated the use of optimization to improve the efficiency of the process.

We emphasize that these developments are generic, and we believe them to be applicable in the modeling and optimization of a broad gamut of process and energy systems.

Acknowledgment

Partial financial support for this work from the American Chemical Society – Petroleum Research Fund under grant 52335-DNI9 is acknowledged with gratitude.

Notation

$\mathbf{a}(t)$ = State variables in DAE formulation
$\mathbf{b}(t)$ = Algebraic variables in DAE formulation
π = Parameters in DAE formulation
ϕ = Differential equations in DAE formulation
γ = Algebraic equations in DAE formulation
$\mathbf{a}_0$ = Initial conditions in DAE formulation
$\mathbf{x}$ = Model variables
$\mathbf{z}$ = Optimization decision variables
CONV = Convective terms in unit op. model
GEN = Source terms in unit op. model
FLUX = Flux terms in unit op. model
M_i = Molar flow rate of component i, mol/s
r_i = Molar generation rate of component i, mol/s
H = Heat flow in/out of unit, W
H_{gen} = Heat generation rate, W
Q_{ext} = Rate of heat flux, W
M^L = Liquid molar flow rate, mol/s
M^V = Vapor molar flow rate, mol/s
H^L = Liquid heat flow, W
H^V = Vapor heat flow, W
$\dot{W}$ = Compression work applied to system, W
T = Temperature, K
T_0 = Initial condition for temperature, K
τ_T = Energy balance time constant, A.U.
U = Heat-transfer coefficient, W/m^2/K
A = Heat transfer area, m^2
T_{jacket} = Jacket temperature, K
α = Integrated parameter
τ_p = Parameter integration time constant, A.U.
k_j^0 = Pre-exponential factor for reaction j
ΔH_j = Heat of reaction j, J/mol
V = Liquid flow rate, mol/s
L = Vapor flow rate, mol/s
y_i = Vapor phase composition
x_i = Liquid phase composition
K_i = Split fraction of component i
Φ^L = Liquid fugacity coefficient
Φ^V = Vapor fugacity coefficient
τ_{f} = Flash time constant, A.U.
T_{bub} = Bubble temperature, K
T_{dew} = Dew temperature, K
$P_{\text{vap}}(T)$ = Vapor pressure, Pa
K_h = Split fraction of heaviest component
K_l = Split fraction of lightest component
$M_{i,\text{tear}}$ = Molar flow of component i in tear stream, mol/s
H_{tear} = Heat flow of tear stream, W
τ_{r} = Recycle time constant, A.U.
L_{j+1} = Liquid flow from stage $j+1$, mol/s
V_{j-1} = Vapor flow from stage $j-1$, mol/s
$x_{i,j+1}$ = Liquid mole fraction from stage $j+1$
$y_{i,j-1}$ = Vapor mole fraction from stage $j-1$
L_j = Liquid flow from stage j, mol/s
V_j = Vapor flow from stage j, mol/s
$x_{i,j}$ = Liquid mole fraction from stage j
$y_{i,j}$ = Vapor mole fraction from stage j
H_{j+1}^L = Liquid energy flow from stage $j+1$, W
H_{j-1}^V = Vapor energy flow from stage $j-1$, W
H_j^L = Liquid energy flow from stage j, W
H_j^V = Vapor energy flow from stage j, W
$M_{i,j}^F$ = Total molar flow of component i to stage j, mol/s
H_j^F = Total energy flow to stage j, W
ΔT = Temperature difference, K
T_{in}^{h} = Hot stream inlet temperature, K
$T_{\text{out}}^{\text{h}}$ = Hot stream outlet temperature, K
T_{in}^{c} = Cold stream inlet temperature, K
$T_{\text{out}}^{\text{c}}$ = Cold stream outlet temperature, K
η_{c} = Isentropic compressor efficiency
η_{t} = Isentropic turbine efficiency
$H_{\text{out}}^{\text{ise}}$ = Isentropic enthalpy, W
$T_{\text{out}}^{\text{ise}}$ = Isentropic temperature, K
$S_{\text{out}}^{\text{ise}}$ = Isentropic entropy, W/K
S = Entropic flow, W/K
P = Pressure, Pa
K_a = Equilibrium constant
a_i = Activity of each component
v_i = Stoichiometric coefficients
$\Delta G_{\text{rxn}}^{\circ}$ = Gibbs energy of reaction, J/mol
R = Gas constant, J/mol/K
X = Conversion
v_{key} = Stoichiometric coefficient of key component
τ_{u} = Intraunit flow time constant, A.U.

Literature Cited

1. Biegler LT, Grossmann IE. Retrospective on optimization. *Comput Chem Eng*. 2004;28(8):1169–1192.
2. Biegler LT. *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*. USA: SIAM, 2010.
3. Diwekar U, Shastri Y. Design for environment: a state-of-the-art review. *Clean Technol Environ Policy*. 2011;13(2):227–240.
4. Yuan Z, Chen B, Sin G, Gani R. State-of-the-art and progress in the optimization-based simultaneous design and control for chemical processes. *AIChE J*. 2012;58(6):1640–1659.
5. Biegler LT, Grossmann IE, Westerberg AW. *Systematic Methods for Chemical Process Design*. Old Tappan, NJ: Prentice Hall, 1997.
6. Pantelides CC, Renfro JG. The online use of first-principles models in process operations: review, current status and future needs. *Comput Chem Eng*. 2013;51:136–148.
7. Cox RK, Smith JF, Dimitratos Y. Can simulation technology enable a paradigm shift in process control?: modeling for the rest of us. *Comput Chem Eng*. 2006;30(10):1542–1552.
8. Soares RP. Finding all real solutions of nonlinear systems of equations with discontinuities by a modified affine arithmetic. *Comput Chem Eng*. 2013;48:48–57.
9. Stuber MD, Kumar V, Barton PI. Nonsmooth exclusion test for finding all solutions of nonlinear equations. *BIT Numer Math*. 2010; 50(4):885–917.
10. Malinen I, Tanskanen J. Homotopy parameter bounding in increasing the robustness of homotopy continuation methods in multiplicity studies. *Comput Chem Eng*. 2010;34(11):1761–1774.
11. Pantelides CC, Urban ZE. Process modelling technology: a critical review of recent developments. *Proceedings of the International Conference on Foundations of Process Design, FOCAPD*. Princeton, NJ: CaChE Corp., 2004:69–83.
12. Paloschi JR. Using sparse bounded homotopies in the speedup simulation package. *Comput Chem Eng*. 1998;22(9):1181–1187.
13. Rahimian SK, Jalali F, Seader JD, White RE. A new homotopy for seeking all real roots of a nonlinear equation. *Comput Chem Eng*. 2011;35(3):403–411.
14. Schnepper CA, Stadtherr MA. Robust process simulation using interval methods. *Comput Chem Eng*. 1996;20(2):187–199.
15. Maier RW, Brennecke JF, Stadtherr MA. Reliable computation of homogeneous azeotropes. *AIChE J*. 2004;44(8):1745–1755.
16. Gau CY, Stadtherr MA. New interval methodologies for reliable chemical process modeling. *Comput Chem Eng*. 2002;26(6):827–840.
17. Lin Y, Gwaltney CR, Stadtherr MA. Reliable modeling and optimization for chemical engineering applications: interval analysis approach. *Reliab Comput*. 2006;12(6):427–450.
18. Lin Y, Stadtherr MA. LP strategy for the interval-newton method in deterministic global optimization. *Ind Eng Chem Res*. 2004;43(14):3741–3749.
19. Lucia A, Feng Y. Global terrain methods. *Comput Chem Eng*. 2002; 26(4):529–546.
20. Lucia A, Feng Y. Multivariable terrain methods. *AIChE J*. 2004; 49(10):2553–2563.
21. Maranas CD, Floudas CA. Finding all solutions of nonlinearly constrained systems of equations. *J Global Optim*. 1995;7(2):143–182.
22. Fletcher R, Morton W. Initialising distillation column models. *Comput Chem Eng*. 2000;23(11):1811–1824.
23. Brenan KE, Campbell SL, Petzold LR. *Numerical Solution of Initial Value Problems in Differential Algebaric Equations. Classics in Applied Mathematics*. Philadelphia, PA, USA: SIAM, 1996.
24. Kumar A, Daoutidis P. *Control of Nonlinear Differential Equation Systems, Volume 397 of Research Notes in Mathematics Series*. Boca Raton, FL, USA: Chapman & Hall/CRC, 1999.
25. Cellier F, Koffman E. *Continuous System Simulation*. New York, NY: Springer Verlag, 2006.
26. Kelley CT, Keyes DE. Convergence analysis of pseudo-transient continuation. *SIAM J Numer Anal*. 1998;35(2):508–523.

27. Deuflhard P. *Adaptive Pseudo-Transient Continuation for Nonlinear Steady State Problems*. Berlin, Germany: Konrad-Zuse-Zentrum für Informationstechnik, 2002.
28. Fowler KR, Kelley CT. Pseudo-transient continuation for nonsmooth nonlinear equations. *SIAM J Numer Anal*. 2005;43(4):1385–1406.
29. Coffey TS, Kelley CT, Keyes DE. Pseudotransient continuation and differential-algebraic equations. *SIAM J Sci Comput*. 2003;25(2): 553–569.
30. Kant T, Patel S. Transient/pseudo-transient finite element small/large deformation analysis of two-dimensional problems. *Comput Struct*. 1990;36(3):421–427.
31. Hazra SB. *An Efficient Method for Aerodynamic Shape Optimization*. 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization, Albany, NY, 2004.
32. Shestakov AI, Milovich JL, Noy A. Solution of the nonlinear Poisson–Boltzmann equation using pseudo-transient continuation and the finite element method. *J Colloid Interface Sci*. 2002;247(1):62–79.
33. Grasser T, Selberherr S. Mixed-mode device simulation. *Microelectron J*. 2000;31(11):873–881.
34. Kravaris C, Niemiec M, Berber R, Brosilow CB. Nonlinear model-based control of nonminimum-phase processes. In: Berber R, Kravaris C, editors. *Nonlinear Model Based Process Control*. Dordrecht: Kluwer Academic Publishers, 1998.
35. Wilson S, Manousiouthakis V. Ideas approach to process network synthesis: application to multicomponent men. *AIChE J*. 2000; 46(12):2408–2416.
36. Skogestad S, Postlethwaite I. *Multivariable Feedback Control: Analysis and Design*, *Vol 2*. New York: Wiley, 2007.
37. Biegler LT, Hughes RR. Infeasible path optimization with sequential modular simulators. *AIChE J*. 1982;28(6):994–1002.
38. Baldea M, Daoutidis P. Control of integrated process networks–a multi-time scale perspective. *Comput Chem Eng*. 2007;31:426–444.
39. Nikačević NM, Huesman AEM, Van den Hof PMJ, Stankiewicz AI. Opportunities and challenges for process control in process intensification. *Chem Eng Process*. 2012;52:1–15.
40. Lutze P, Babi DK, Woodley JM, Gani R. Phenomena based methodology for process synthesis incorporating process intensification. *Ind Eng Chem Res*. 2013;52(22):7127–7144.
41. Baldea M, El-Farra NH, Ydstie BE. Dynamics and control of chemical process networks: integrating physics, communication and computation. *Comput Chem Eng*. 2013;51:42–54.
42. Baldea M, Daoutidis P. A general analysis and control framework for process systems with inventory recycling. *Int J Robust Nonlinear Control*. In press. Available at http://dx.doi.org/10.1002/rnc.3029. Accessed on July 29, 2014.
43. Baldea M, Touretzky CR. Nonlinear model predictive control of energy-integrated process systems. *Syst Control Lett*. 2013;62:723–731.
44. Baldea M, Daoutidis P. *Dynamics and Control of Integrated Process Systems*. Cambridge: Cambridge University Press,2012.
45. Vora NP, Contou-Carrere MN, Daoutidis P. Model reduction of multiple time scale processes in non-standard singularly perturbed form. *Model Reduction and Coarse-Graining Approaches for Multiscale Phenomena*. Berlin, Germany: Springer, 2006:99–113.
46. Henley EJ, Seader JD, Roper DK. *Separation Process Principles*. USA: Wiley, 2011.
47. Sandler SI. *Chemical, Biochemical, and Engineering Thermodynamics*, Vol 4. Hoboken, NJ: Wiley, 2006.
48. Klatt KU, Marquardt W. Perspectives for process systems engineering personal views from academia and industry. *Comput Chem Eng*. 2009;33(3):536–550.
49. Grossmann IE, Guillén-Gosálbez G. Scope for the application of mathematical programming techniques in the synthesis and planning of sustainable processes. *Comput Chem Eng*. 2010;34(9):1365–1376.
50. Zanfir M, Baldea M, Daoutidis P. Optimizing the catalyst distribution for countercurrent methane steam reforming in plate reactors. *AIChE J*. 2011;57(9):2518–2528.
51. Baldea M, Jibb RJ, Cano A, Ramos A. A framework for dynamic modeling and optimization of multi-stream plate-fin heat exchangers. *AIChE Annual Meeting*. Nashville, TN, 2009.
52. Pattison RC, Baldea M. A thermal-flywheel approach to distributed temperature control in microchannel reactors. *AIChE J*. 2013;59(6): 2051–2061.
53. Pattison RC, Estep FE, Baldea M. Pseudodistributed feed configurations for catalytic plate microchannel reactors. *Ind Eng Chem Res*. 2013;53(13):5028–5037.
54. Wang S, Baldea M. Identification-based optimization of dynamical systems under uncertainty. *Comput Chem Eng*. 2014;64:138–152.
55. Vassiliadis VS, Sargent RWH, Pantelides CC. Solution of a class of multistage dynamic optimization problems. 1. Problems without path constraints. *Ind Eng Chem Res*. 1994;33(9):2111–2122.
56. Blanco Gutiérrez RF, Pantelides CC, Adjiman CS. Risk analysis and robust design under technological uncertainty. *Computer Aided Chemical Engineering*, *Vol. 21*, Amsterdam, The Netherlands: Elsevier, 2006:191–196.
57. Process Systems Enterprise. General PROcess Modeling System (gPROMS), 1997–2014. Available at: www.psenterprise.com/gproms. Accessed on July 29, 2014.
58. Wayburn TL, Seader JD. Solution of systems of interlinked distillation columns by differential homotopy-continuation methods. *Proceedings of the Second International Conference on Foundations of Computer-Aided Process Design*, Snowmass, CO: CaChE Corp., 1984:765–862.
59. Lin WJ, Seader JD, Wayburn TL. Computing multiple solutions to systems of interlinked separation columns. *AIChE J*. 1987;33(6): 886–897.
60. Higler AP, Taylor R, Krishna R. Nonequilibrium modelling of reactive distillation: multiple steady states in MTBE synthesis. *Chem Eng Sci*. 1999;54(10):1389–1395.
61. Jalali F, Seader JD. Homotopy continuation method in multi-phase multi-reaction equilibrium systems. *Comput Chem Eng*. 1999;23(9): 1319–1331.
62. Bausa J, Marquardt W. Quick and reliable phase stability test in VLLE flash calculations by homotopy continuation. *Comput Chem Eng*. 2000;24(11):2447–2456.
63. Kubíček M. Algorithm 502: dependence of solution of nonlinear systems on a parameter [c5]. *ACM Trans Math Softw*. 1976;2(1):98–107.

Appendix A: Homotopy Continuation

HC is used to find the solution to a set of nonlinear algebraic equations, $\mathbf{f}(\mathbf{x})=0$ where $\mathbf{f}:\Re^n \rightarrow \Re^n$. The method gained popularity in chemical engineering in the 1980s,[58] and has been used to find solutions of separation column models,[59] as well as reactive distillation columns with multiple steady states.[60] Initializing reactors has also been performed with the use of homotopies,[11] along with finding multiphase multireaction equilibrium solutions,[61] and reliably solving vapor-liquid-liquid equilibria (VLLE) flash calculations.[62]

The concept of HC is to start from a system of equations $\mathbf{g}(\mathbf{x})=\mathbf{0}$ that has a solution that is either easy to find or known, and gradually modify the system to reach the solution of the original equations $\mathbf{f}(\mathbf{x})=\mathbf{0}$

$$\mathbf{H}(\mathbf{x},\lambda)=(1-\lambda)\mathbf{f}(\mathbf{x})+\lambda\mathbf{g}(\mathbf{x})=\mathbf{0} \tag{A1}$$

where $\mathbf{H}(\mathbf{x},\lambda)$ is the homotopy function and λ is the homotopy parameter. When $\lambda=1$, $\mathbf{H}(\mathbf{x},1)=\mathbf{g}(\mathbf{x})$, and when $\lambda=0$, $\mathbf{H}(\mathbf{x},0)=\mathbf{f}(\mathbf{x})$. A significant body of work focuses on the selection of $\mathbf{g}(\mathbf{x})$ and on defining the transition path from $\lambda=1$ to $\lambda=0$. The three most common choices for $\mathbf{g}(\mathbf{x})$ are Newton homotopy, fixed-point homotopy, and affine homotopy.[61]

Differentiating the homotopy problem was formalized in the 1970s.[63] The concept is to trace the homotopy path using ODEs and provide initial conditions for the state variables, $\mathbf{X}_0$.[59] Suppose we select $\mathbf{g}(\mathbf{x})=\mathbf{f}(\mathbf{x})-\mathbf{f}(\mathbf{x}_0)$ (Newton homotopy). This gives

$$\mathbf{H}(\mathbf{x},\lambda)=\mathbf{f}(\mathbf{x})-\lambda\mathbf{f}(\mathbf{x}_0) \tag{A2}$$

Differentiating with respect to the homotopy arclength s gives

$$\frac{d\mathbf{H}}{ds}=\frac{d\mathbf{f}}{d\mathbf{x}}\frac{d\mathbf{x}}{ds}-\frac{d\lambda}{ds}\mathbf{f}(\mathbf{x}_0)=\mathbf{0} \tag{A3}$$

This leads to the system of differential equations

$$\begin{bmatrix} \frac{d\mathbf{f}}{d\mathbf{x}} & -\mathbf{f}(\mathbf{x}_0) \end{bmatrix} \begin{bmatrix} \frac{d\mathbf{x}}{ds} \\ \frac{d\lambda}{ds} \end{bmatrix} = \mathbf{0}$$
$$\frac{d\mathbf{x}}{ds^T}\frac{d\mathbf{x}}{ds} + \left(\frac{d\lambda}{ds}\right)^2 = 1 \tag{A4}$$

An initial value problem formulation is completed by applying the initial conditions, $\mathbf{x}(0)=\mathbf{x}_0$ (which can be selected arbitrarily) and $\lambda(0)=1$, and $\lambda \in [0,1]$. Integrating these equations is typically done with a predictor-corrector step and accuracy and stability are accounted for.

HC differs in several aspects from PTC. In PTC, setting the time derivatives of the state variables to zero will recover the algebraic equation model, and the solution is found at the steady state of the dynamic model. In HC, the solution of the system is found when the dynamic parameter, λ is equal to zero; this, however, is not a steady state of the dynamic system. Another difference is that the HC dynamic system follows a trajectory that is dependent on the inverse of the Jacobian, $\frac{d\mathbf{f}}{d\mathbf{x}}^{-1}$ rather than the actual functions, $\mathbf{f}_i$ as in PTC. This can cause difficulties if the Jacobian matrix becomes singular in the course of the time integration.[61]

The significant performance difference between pseudo-transient flowsheet simulation and the use of HC methods in flowsheet simulation is the ability to decouple the solution of individual unit operation models and the solution of the entire flowsheet via time scale decomposition. PTC allows for the adjustment of the filter time dynamics for each function $\mathbf{f}_i$. Thus, the fastest components (i.e., individual unit operations) with the largest residuals will converge to a solution of the specific subsystem first, while the entire flowsheet will converge on a slower time scale.

From a numerical perspective (assuming a variable step integration method is used), small time steps will be necessary to initially converge the subsystems, but longer time steps can be used later in the time integration to converge the whole flowsheet. In HC, if the local Jacobian matrix is near singular ($\frac{d\mathbf{f}}{d\mathbf{x}} \approx 0$), the derivative of $\mathbf{x}$ with respect to the arclength (s) becomes very large (see Eq. A3 above), and small time steps are required to accurately follow the trajectory.

Manuscript received May 7, 2014, and revision received July 10, 2014.