Nonsmooth Distillation Models Robust to Convergence Errors: Numerical Methods and Topological Aspects

by

Suzane Martins Cavalcanti

Submitted to the Department of Chemical Engineering and Center for Computational Science & Engineering in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY IN CHEMICAL ENGINEERING AND COMPUTATION

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2024

Authored by: ....
Suzane Martins Cavalcanti
Department of Chemical Engineering
January 12, 2024

Certified by: ....
Paul I. Barton
Lammot du Pont Professor of Chemical Engineering
Thesis Supervisor

Accepted by: ....
Hadley D. Sikes
Willard Henry Dow Professor of Chemical Engineering
Graduate Officer, Department of Chemical Engineering

Accepted by: ....
Nicolas Hadjiconstantinou
Professor of Mechanical Engineering
Co-Director, Center for Computational Science & Engineering

Nonsmooth Distillation Models Robust to Convergence Errors: Numerical Methods and Topological Aspects

by

Suzane Martins Cavalcanti

Submitted to the Department of Chemical Engineering and Center for Computational Science & Engineering on January 12, 2024 in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY IN CHEMICAL ENGINEERING AND COMPUTATION

ABSTRACT

Distillation is the most widely used (yet highly energy-intensive) industrial separation method and one of the most well-studied chemical engineering processes. However, engineers still often encounter distillation simulation errors while using state-of-the-art process software such as Aspen Plus and HYSYS. These errors preclude one from converging flowsheets with recycle streams and from successfully utilizing rigorous process optimization methods, which are both essential tasks in designing more energy-efficient and economically viable processes. In this thesis we address these challenges by developing nonsmooth (i.e., non-differentiable) distillation models and equation-solving methods that are robust to a wide range of convergence errors. As demonstrated by our results, nonsmooth functions are a powerful tool due to their ability to automatically switch between different terms, which allows us to describe and adapt to different modes of behavior of a system using a single model.

To investigate the “dry column” errors often encountered in Aspen Plus we developed a nonsmooth version of the MESH model, which can be solved with Newton-type methods using exact generalized derivatives obtained with automatic differentiation techniques. This model allows us to simulate distillation columns in which one or more stages operate with a single phase, either superheated vapor or subcooled liquid. By developing continuation methods to simulate the nonsmooth MESH model, we discovered a new class of degenerate bifurcations in distillation columns which are generally observed regardless of the mixture or parameter being varied. These bifurcations are characterized by infinitely-many, multiple steady states with dry/vaporless stages, and happen at the so-called critical parameter value associated with the first flow rate in the column reaching zero.

In order to describe the topological structure of these bifurcation curves in a rigorous fashion, we proved a piecewise-differentiable (PC^r) Rank Theorem that allows us to characterize nonsmooth curves and surfaces as PC^r manifolds, according to the theoretical framework introduced in this thesis. We also generalized a previous Lipschitz Rank Theorem and applied it to define Lipschitz embedded submanifolds. Further, we developed sufficient and practically verifiable conditions, in terms of the B-subdifferential generalized derivative, that can be applied to the PC^r MESH model function to theoretically predict the geometric behavior of its level sets that we observed numerically.

The nonsmooth MESH model overcomes dry column errors for specifications that lead to a feasible state with dry/vaporless stages. To address convergence failure due to column specifications being infeasible, which in general is unpredictable prior to simulation, we developed a second class of nonsmooth, adaptive distillation models. Our modeling strategies return a feasible solution even when one or two specifications are infeasible, by automatically resetting the latter to ensure that all flow rates are within their imposed lower and upper bounds. Additionally, we developed a nonsmooth version of the inside-out algorithm to converge these nonsmooth models reliably from an *ab initio* starting point, even for highly non-ideal mixtures. With a series of test cases, we demonstrate that our distillation modeling methods outperform Aspen Plus due to their ability to converge both individual columns and flowsheets with recycle under infeasible or near-infeasible specifications, non-ideal thermodynamics, and poor initial guesses.

Thesis Supervisor: Paul I. Barton

Title: Lammot du Pont Professor of Chemical Engineering

Acknowledgments

I can still remember how dreamlike it felt to receive a call from professor Kristala Prather informing of my acceptance into the MIT ChemE PhD program. As this chapter of my life comes to a conclusion, recalling my experience at MIT feels just as, if not more, surreal. I am grateful to many people for supporting me, directly and indirectly, throughout this long and life-changing journey.

I have been fortunate to have had Paul Barton as my PhD advisor. Your detailed technical feedback and your interest in my work were invaluable and extremely motivating. I also enjoyed the occasional group meetings dedicated to footage of your travels. I cannot thank you enough for giving me the intellectual freedom to pursue topics that interested me, including some "rabbit holes" that sometimes (but not always) proved to be time-worthy. Further, I will always be grateful for your support and understanding during my personal struggles. I am proud of the work we have accomplished in this thesis.

I am thankful also to my other thesis committee members, Richard Braatz and William Green, for their meaningful feedback and career advice. Throughout our committee meetings, your questions instigated me to think both more deeply and more broadly about my research. I would like to acknowledge my main funding source, HighEFF (Centre for Environment-friendly Energy Research, 257632, Research Council of Norway), as well as my MathWorks Engineering and Exxon-MIT Energy Fellowships. I am grateful to Truls Gundersen for supervising the HighEFF partnership with MIT, and for our meetings throughout the years in which he demonstrated interest in my work and provided useful feedback.

During my time at MIT I have had the pleasure of meeting many amazing people whose brilliance inspired me to go further than I thought was possible. I'm grateful to my research group mates at PSEL, past and present, for all the fruitful discussions and for being there when I needed a distraction from work. In particular, I'd like to thank Harry Watson and Peter Stechlinski for all of their help in getting me started with nonsmooth analysis, and Matias Vikse for making me laugh. I could not have gotten through the workload and stress of the first semester classes without my brilliant and fun ChemE PhD classmates. I will always cherish the friends I made during that semester, including but not limited to Kevin Silmore, Sharon Lin, Kindle Williams and Caroline Nielsen. Caroline, it was awesome having you also as a group mate to go through math classes and research activities together. Rohan Kadambi, Priyanka Raghavan and Jing Ying, I am glad that we became friends through TAing 10.34. Rohan, I had a lot of fun nerding out over 10.34 recitations and homework with you. I have been fortunate to have made great friends also outside of ChemE. Sam Raymond, thank you for being my gym buddy, fellow LOTR nerd, and for being there for me during extremely hard times. Afra Ansaria, I am blessed to call you my friend. Thank you for all our deep conversations and fun times, and for listening to me talk about my PhD hardships. Matt Holsey, thank you for distracting me from work and for encouraging me during the last steps of my PhD.

I would like to thank Unicamp professors Marisa Beppu, Flavio Vasconcelos da Silva and Roger Zemp for their support in submitting a successful application to the MIT PhD program. I would also like to offer my appreciation to all of the teachers and professors that I learned and drew inspiration from throughout my life.

I am forever grateful for the emotional support from my family members, especially my parents. I could not have made it this far without your lifelong sponsorship. Thank you for believing in me, for remaining hopeful when I was not able to, and for the countless phone calls in which you supported me from far away.

Above all, I am thankful to God for leading me through this journey from beginning to end. He is the one behind the very fabric of reality; my job and privilege has been simply to discover a few of its beautiful hidden patterns. I dedicate this thesis to Jesus, who is the author of my existence, and who brought me back to finish the work that He had started through me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1. Convergence errors in distillation simulation

Distillation is the most widely used industrial separation method. However, it is very-energy intensive, being responsible for as much as 30 % of the total energy use in the industry [38] and for 90-95 % of separation energy [1]. In order to develop more energy-efficient distillation processes, we need an accurate model that can be solved reliably under all the varying input conditions encountered during flowsheet convergence and optimization. This is crucial in complex, integrated and energy-intensive processes such as oxycombustion [77], in which cryogenic air distillation supplies nearly-pure oxygen to a power plant so as to facilitate CO_2 purification post-combustion. However, existing process models, simulation and optimization methods are still not realistic, reliable and versatile enough to ensure optimal plant design, which is a necessary step towards economical viability of cleaner energy initiatives.

In the context of distillation, commercial software such as Aspen Plus have well-known failure issues related to dry and vaporless stages, because they are restricted to the smooth MESH model which assumes both liquid and vapor are always present in every stage. Unfortunately, dry column errors are not the only way in which distillation simulation software fails to converge to a physically valid solution. Aspen Plus does not provide any insight on the origin of the more generic convergence errors commonly encountered within distillation simulation, or on how the input specifications might be changed to allow the model to converge. In general, selecting a set of feasible column inputs is not an obvious task; that is especially evident for product purity specifications, which might exhibit a quite narrow (and unpredictable) range of allowed values.

Now we temporarily take a step back from distillation simulation to ask a much broader question: why does process simulation software fail to converge? Before attempting to answer that, we need to clarify a few key concepts. When dealing with steady-state processes, simulation entails choosing a set of input specifications and then predicting what steady state the process should attain, if any, for that set of specifications. To make this prediction without conducting experiments, first we have to come up with physical laws (e.g., mass and energy balances, thermodynamic laws, reaction kinetics) to describe the process. Then we try to find a steady state, described in terms of a vector of values $\mathbf{x}_s \in \mathbb{R}^n$ for the process variables, that satisfies these laws. Moreover, the variable values $\mathbf{x}_s$ should be physically realizable. If such an $\mathbf{x}_s$ exists for a set of specifications, then we say that the latter are *feasible*. Our choice of physical laws might have considerable impact on the applicability of our model in predicting the steady state reached by real process equipment. However, experimental validation of a given modeling paradigm is outside of the scope of this thesis and as such will have no bearing on our concept of feasibility.

The next question is, how do we determine if a given set of specifications is feasible or infeasible? First we must translate the process physical laws, to the best of our ability, into a system of nonlinear algebraic equations $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, where in general $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ describes the same number of equations as the number of variables. We then employ equation-solving methods in an attempt to find a mathematical solution $\mathbf{x}^*$ of $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ that also corresponds to a physically valid steady-state $\mathbf{x}_s$. If we succeed in doing so, we say that our simulation converged to a solution and we conclude the specifications are feasible. Unfortunately, that is not always the outcome of trying to solve a model $\mathbf{f}(\mathbf{x}) = \mathbf{0}$; we may find a mathematical model solution $\mathbf{x}^*$ in which process variables assume non-physical values, or no solution at all. However, being unable to find a valid model solution does not guarantee that no physically valid steady state $\mathbf{x}_s$ exists.

Unless the specification values themselves are out of their own physical bounds (e.g., specifying a temperature of -10 K in a flash vessel or a negative reflux ratio in a distillation column), determining if a set of specifications is infeasible is a hard task which cannot be achieved without (repeatedly) simulating the process. The best we can do is formulate our model $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ thoroughly enough so that it enforces the necessary physical laws under all possible modes of behavior of the process. Nonsmooth (i.e., non-differentiable) functions are particularly well-suited for such a task, given their ability to automatically switch between different terms. Once we have a suitable model we can gradually vary the specification values, in a process called parametric continuation, and compute the model solutions until we eventually arrive at a specification value for which these solutions cease to exist. In that case, if our model is thorough enough we can conclude that any specification values beyond the said value are infeasible.

In summary, our simulation might fail to converge to a valid steady state $\mathbf{x}_s$ for a given set of specifications due to three possible scenarios:

- (**Scenario 1**) A physically valid solution $\mathbf{x}_s$ exists but it is not a mathematical solution of the model, and as such cannot be found by any equation-solving methods.
- (**Scenario 2**) No physically valid solution $\mathbf{x}_s$ exists, in which case the specifications are referred to as *infeasible*.
- (**Scenario 3**) A physically valid solution $\mathbf{x}_s$ exists and it is also a mathematical solution of the model, but the equation-solving methods employed are unable to find it.

With this general framework in mind, we are now ready to give an overview of the main mechanisms behind convergence errors in distillation simulation, and how the models and algorithms developed in this thesis can overcome these errors.

First, we address the well-known “dry column” convergence errors within distillation simulation. The standard MESH model, which is employed in Aspen Plus’ RadFrac unit model, does not encompass solutions where one or more stages are dry/vaporless and operating outside of equilibrium (i.e., with superheated vapor or subcooled liquid). On the other hand, our nonsmooth MESH model developed in Chapter 3 overcomes this limitation due to its ability to enforce different equations depending on the mode of operation of the system. Our model allows us to demonstrate that some specification values, despite leading to dry column errors in Aspen Plus, are actually feasible and generate valid solutions with dry/vaporless stages outside of equilibrium. As such, we can overcome dry column errors that happen within the context of Scenario 1. With our modeling strategy we have also been able to discover novel bifurcations in dry/vaporless columns, which exhibit an infinite number of multiple steady states.

By developing special continuation methods for our nonsmooth MESH model, we have verified that a significant range of specification values, though seemingly feasible at first glance, can be infeasible due to liquid and/or vapor flow rates tending to become negative. As could be expected, Aspen Plus exhibits the same dry column errors for these infeasible specification values. Unfortunately, the nonsmooth MESH model cannot avoid these Scenario 2 convergence errors either.

Surprisingly, Chapter 4 demonstrates that nonsmooth functions allow us to avoid convergence errors and still obtain a useful simulation output even for infeasible specifications, without having any prior knowledge about the latter. Our nonsmooth adaptive models are capable of overcoming dry column errors and also of automatically resetting one or more specifications if they happen to be infeasible, due to the flow rates becoming either negative or unbounded above. This way, we address a broad class of convergence errors within distillation simulation which are due to Scenario 2.

Simultaneous convergence of the standard MESH equations can be unreliable and highly dependent on a good initial guess, especially with complex non-ideal systems. To prevent numerical convergence failure due to Scenario 3 as well as dry column and infeasibility errors, we develop a nonsmooth version of the inside-out algorithm [14] in Chapter 5 to converge our nonsmooth adaptive model equations. The model employs the standard outer loop structure of the inside-out algorithm, while the inner loop (in the format proposed by Russell [75]) was restructured to employ new iteration variables, which allow convergence to MESH solutions with dry and vaporless stages and do not require heuristic scaling factors. Moreover, the new inner loop converges the specification equations from the nonsmooth adaptive modeling strategy of choice.

1.2. The topology of level sets of nonsmooth functions

Nonsmooth systems of equations $\mathbf{f}(\mathbf{x}) = \mathbf{c}$ are a powerful modeling tool to describe complex systems with multiple physical modes of behavior. For instance, piecewise-differentiable (PC^r) functions (see Section 2.1.2) have been employed to model multi-stream heat exchangers with phase change [86] with a single equation solving task, which cannot be achieved with other modeling approaches in the literature. Moreover, Chapters 3, 4 and 5 demonstrate the usefulness of using nonsmooth distillation models both to obtain new types of feasible solutions and to automatically reset parameters that may turn out to be infeasible.

In this thesis, we are interested in characterizing the local structure of each level set $\mathbf{f}^{-1}(\mathbf{c}) \subset \mathbb{R}^n$ of a nonsmooth function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$, that is, the set of all solutions $\mathbf{x}$ to the system of equations $\mathbf{f}(\mathbf{x}) = \mathbf{c}$. By nonsmooth we mean either locally Lipschitz continuous or PC^r. Our main motivation for researching this topic stems from the unusual behavior of the bifurcations of our nonsmooth MESH distillation model, as described in Chapter 3. In general, the topological structure of the level sets $\mathbf{f}^{-1}(\mathbf{c})$ of a nonsmooth model determines the existence, uniqueness and behavior of the model solutions under variation of one or more parameters. As another example application, the feasible set of Mathematical Programs with Complementarity Constraints (MPCCs) of the form

$$\begin{aligned}&\min f(\mathbf{x}) \qquad\qquad (1.1)\\ &\text{s.t. } \mathbf{F}_1(\mathbf{x})^{\mathrm{T}}\mathbf{F}_2(\mathbf{x}) = 0, \quad \mathbf{F}_1(\mathbf{x}), \mathbf{F}_2(\mathbf{x}) \geq \mathbf{0},\end{aligned}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $\mathbf{F}_1, \mathbf{F}_2 : \mathbb{R}^n \to \mathbb{R}^m$ are C^1, corresponds to the zero-level set of the PC^1 function

$$\mathbf{g}(\mathbf{x}) = \min\left(\mathbf{F}_1(\mathbf{x}), \mathbf{F}_2(\mathbf{x})\right), \quad \mathbf{g} : \mathbb{R}^n \to \mathbb{R}^m. \qquad (1.2)$$

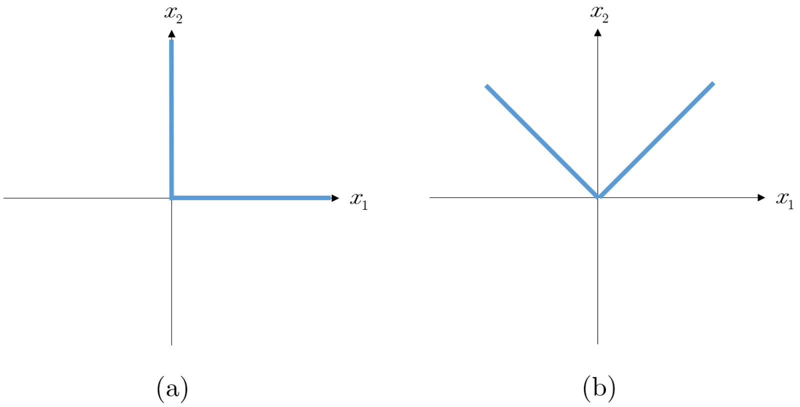
The stability of the feasible set under perturbation of the functions $\mathbf{F}_1, \mathbf{F}_2$ is described in terms of the topological structure of $\mathbf{g}^{-1}(\mathbf{0})$ (e.g., see [43]).

The case $m \leq n$ is usually analyzed in terms of Implicit Function Theorems. If the generalized derivatives of $\mathbf{f}$ at $\mathbf{x}_0$ are "non-degenerate" (in some specific sense) with respect to the first m variables x_i, then we obtain Lipschitz and PC^r Implicit Function Theorems by applying the corresponding Inverse Function Theorems (see Section 6.2.1) to the function

$$\mathbf{h} : \mathbb{R}^n \to \mathbb{R}^n, \quad \mathbf{h}(\mathbf{x}) = (\mathbf{f}(\mathbf{x}), x_{m+1}, \ldots, x_n) \qquad (1.3)$$

at $\mathbf{x}_0$. In turn, this allows us to conclude $\mathbf{f}^{-1}(\mathbf{c})$ is locally the graph of a nonsmooth function $\mathbf{y} : \mathbb{R}^{n-m} \to \mathbb{R}^m$. Of course, requiring the non-degenerate variables to be exactly the first ones is too restrictive, so the next idea is to allow for a reordering or permutation of the coordinates x_i before applying the Implicit Function Theorem. This approach always works in the smooth C^r case, in the sense that whenever $\mathbf{f}^{-1}(\mathbf{c}) \subset \mathbb{R}^n$ is a smooth manifold it corresponds, within coordinate permutation, to the graph of an implicit C^r function. However, this permutation strategy fails even with extremely simple nonsmooth functions.

Consider the zero level set of the piecewise-linear function $\mathbf{f}(x_1, x_2) = \min(x_1, x_2)$ around $\mathbf{x}_0 = \mathbf{0}$, depicted in Figure 1.1(a). It behaves as a 1-dimensional manifold that is nonsmooth, yet neither coordinate x_i corresponds to an implicit function of the other around the origin. Nevertheless, $\mathbf{f}^{-1}(0)$ can be transformed into the graph of the absolute value function, depicted in Figure 1.1(b), by a simple rotation of the axes. This leads us towards the more general concept of nonsmooth submersions (Definition 6.3.3), i.e., functions whose level sets are graphs only after being transformed by a homeomorphism $\mathbf{g}_1$ of the same type as $\mathbf{f}$. Even though this idea allows us to characterize level sets of a wider range of functions, we must turn to Rank Theorems to assess the most general case with arbitrary m, n. In this context, two homeomorphisms $\mathbf{g}_1, \mathbf{g}_2$ might be needed to transform both the coordinates x_i and the function components f_j, respectively.



(a) (b)

Figure 1.1: (a) Zero level set of $\mathbf{f}(x_1, x_2) = \min(x_1, x_2)$, (b) graph of $x_2 = |x_1|$.

The (smooth) Rank Theorem, a traditional result in differential topology, is based on the Inverse Function Theorem. In Chapter 6 we will present a Lipschitz Rank Theorem that generalizes a previous result [6], and the first Rank Theorem for PC^r functions. When the appropriate conditions are satisfied by the (generalized) derivative(s) of $\mathbf{f}$ to characterize the latter as a constant rank function, Rank Theorems allow us to express a given level set $\mathbf{f}^{-1}(\mathbf{c}) \subset \mathbb{R}^n$ locally as the graph of an "implicit" function, within a homeomorphic transformation of the same class as $\mathbf{f}$.

Despite the relevance of nonsmooth level sets to many applications, there is a lack of precise topological notions to describe them. Moreover, one could wish to analyze the topology of abstract level sets or high-dimensional surfaces, which are not subsets of Euclidean space $\mathbb{R}^n$. For these purposes, we need to go beyond Rank Theorems and consider the concept of abstract manifolds, which are sets locally homeomorphic to some Euclidean space of dimension k. Smooth manifolds are a standard concept in differential topology (e.g., see [54]). Abstract Lipschitz manifolds have been previously defined in the literature [74, 58, 67] according to the standard differential topology framework. However, to the best of our knowledge, our definition of PC^r manifolds presented in Chapter 7 is the first well-defined concept for piecewise-differentiable manifolds. This is likely due to the fact that the usual definition of a "piecewise-differentiable" function used in the field of topology [89] involves being able to subdivide the domain into simplexes where the function is smooth. Such functions are not closed under composition, thus they do not constitute a pseudogroup that could generate a valid manifold definition, as opposed to PC^r functions.

Based on the Rank Theorems we develop in Chapter 6, in Chapter 7 we are able to state Level Set Theorems for functions between manifolds, which show that the level sets of constant rank PC^r and Lipschitz functions are embedded submanifolds of the domain manifold. While these results provide the most general conditions under which we can characterize the topological structure of abstract nonsmooth level sets, they can also be applied to our nonsmooth MESH model. As such, in Chapter 7 we are able to theoretically validate which properties of this distillation model give rise to the observed geometric behavior of its bifurcation curves.

Chapter 2

Background

2.1. Nonsmooth Analysis

The directional derivative $d\mathbf{f}_{\mathbf{x}_0} : \mathbb{R}^n \to \mathbb{R}^m$ of a function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ at $\mathbf{x}_0 \in \mathbb{R}^n$ is defined as

$$d\mathbf{f}_{\mathbf{x}_0}(\mathbf{d}) = \mathbf{f}'(\mathbf{x}_0; \mathbf{d}) = \lim_{\alpha \to 0^+} \frac{\mathbf{f}(\mathbf{x}_0 + \alpha \mathbf{d}) - \mathbf{f}(\mathbf{x}_0)}{\alpha} \tag{2.1}$$

if the limit exists $\forall \mathbf{d} \in \mathbb{R}^n$. The Jacobian matrix of $\mathbf{f}$ at $\mathbf{x}_0$, if defined, is denoted $\mathbf{Jf}(\mathbf{x}_0)$.

2.1.1. Locally Lipschitz continuous functions

Let $U \subset \mathbb{R}^n$ be open. A function $\mathbf{f} : U \to \mathbb{R}^m$ is said to be Lipschitz continuous on $U' \subset U$ if $\exists L \in \mathbb{R}$ such that

$$||\mathbf{f}(\mathbf{x}_2) - \mathbf{f}(\mathbf{x}_1)|| \leq L||\mathbf{x}_2 - \mathbf{x}_1||, \quad \forall\ \mathbf{x}_1, \mathbf{x}_2 \in U'. \tag{2.2}$$

$\mathbf{f}$ is said to be: a) locally Lipschitz continuous at $\mathbf{x} \in U$ if there exists a δ-neighborhood $N_\delta(\mathbf{x})$ on which $\mathbf{f}$ is Lipschitz continuous; b) a locally Lipschitz function if it is locally Lipschitz continuous at every point of its domain U. For example, the function $f(x) = x^2$ is locally Lipschitz at every $x \in \mathbb{R}$ but not Lipschitz on $\mathbb{R}$.

According to Rademacher's Theorem, a locally Lipschitz function is differentiable almost everywhere in the Lebesgue measure sense [22]. The two main types of set-valued generalized derivatives of locally Lipschitz functions are the B-subdifferential and the Clarke Jacobian, where the latter is the convex hull of the former. We can also define projections of these sets with respect to only a subset of the variables, as presented in Definition 2.1.1.

Definition 2.1.1 (B-subdifferential and Clarke Jacobian). Let $U \subset \mathbb{R}^n$ be open, $\mathbf{f} : U \to \mathbb{R}^m$ be a locally Lipschitz function at $\mathbf{x}_0 \in U$, and $\Omega_{\mathbf{f}} \subset U$ be the measure-zero set where $\mathbf{f}$ is not differentiable. The B-subdifferential of $\mathbf{f}$ at $\mathbf{x}_0$ is defined as

$$\partial^B \mathbf{f}(\mathbf{x}_0) = \left\{ \lim_{i \to \infty} \mathbf{Jf}(\mathbf{x}_i) : \lim_{i \to \infty} \mathbf{x}_i = \mathbf{x}_0 \text{ and } \mathbf{x}_i \in U \setminus \Omega_{\mathbf{f}} \right\} \tag{2.3}$$

and the Clarke Jacobian of $\mathbf{f}$ at $\mathbf{x}_0$ is defined as

$$\partial \mathbf{f}(\mathbf{x}_0) = \operatorname{conv}\left\{ \partial^B \mathbf{f}(\mathbf{x}_0) \right\} = \operatorname{conv} \left\{ \lim_{i \to \infty} \mathbf{Jf}(\mathbf{x}_i) : \lim_{i \to \infty} \mathbf{x}_i = \mathbf{x}_0 \text{ and } \mathbf{x}_i \in U \setminus S \right\}, \tag{2.4}$$

where $S \subset U$ is any measure zero set containing $\Omega_{\mathbf{f}}$ (see Theorem 4 in [83]).

Given $k \leq n$, the projections of $\partial^B \mathbf{f}(\mathbf{x}_0)$ with respect to the first k variables and to the last $n - k$ variables at $\mathbf{x}_0$ are defined respectively as

$$\boldsymbol{\pi}_k^B \mathbf{f}(\mathbf{x}_0) = \left\{ \mathbf{M} \in \mathbb{R}^{m \times k} : \begin{bmatrix} \mathbf{M} & \mathbf{N} \end{bmatrix} \in \partial^B \mathbf{f}(\mathbf{x}_0) \text{ for some } \mathbf{N} \in \mathbb{R}^{m \times n-k} \right\}, \tag{2.5}$$

$$\boldsymbol{\rho}_{n-k}^B \mathbf{f}(\mathbf{x}_0) = \left\{ \mathbf{N} \in \mathbb{R}^{m \times (n-k)} : \begin{bmatrix} \mathbf{M} & \mathbf{N} \end{bmatrix} \in \partial^B \mathbf{f}(\mathbf{x}_0) \text{ for some } \mathbf{M} \in \mathbb{R}^{m \times k} \right\}. \tag{2.6}$$

The projections $\boldsymbol{\pi}_k \mathbf{f}(\mathbf{x}_0)$ and $\boldsymbol{\rho}_{n-k} \mathbf{f}(\mathbf{x}_0)$ of $\partial \mathbf{f}(\mathbf{x}_0)$ are defined analogously.

Therefore, $\partial^B \mathbf{f}(\mathbf{x}_0)$ can be seen as the set of all "limiting Jacobian matrices" that $\mathbf{f}$ might exhibit as we approach $\mathbf{x}_0$. For instance, the B-subdifferential for the absolute value function at the origin contains two elements, -1 and 1.

The concept of Clarke regularity plays a role in Clarke's Inverse Function Theorem (see Section 2.2).

Definition 2.1.2 (Clarke regularity). Let $U \subset \mathbb{R}^n$ be open, $\mathbf{f} : U \to \mathbb{R}^m$ be a Lipschitz function at $\mathbf{x}_0 \in U$, and $m \leq n$. We say $\mathbf{f}$ is Clarke regular with respect to the first m variables at $\mathbf{x}_0$ if all matrices in $\boldsymbol{\pi}_m \mathbf{f}(\mathbf{x}_0)$ are invertible. If $m = n$, we simply say $\mathbf{f}$ is Clarke regular at $\mathbf{x}_0$.

In general the Clarke Jacobian of a composition of locally Lipschitz functions does not satisfy the chain rule. However, the following proposition provides a special case in which the latter is satisfied, which will be useful within Chapter 6.

Proposition 2.1.3. Let $U \subset \mathbb{R}^n$ be open, $\mathbf{f} : U \to \mathbb{R}^m$ be locally Lipschitz, and $\mathbf{g}_1 : \mathbb{R}^n \to \mathbb{R}^n,\ \mathbf{g}_1(\mathbf{x}) = \mathbf{P}_1\mathbf{x}$ and $\mathbf{g}_2 : \mathbb{R}^m \to \mathbb{R}^m,\ \mathbf{g}_2(\mathbf{x}) = \mathbf{P}_2\mathbf{x}$ be linear homeomorphisms. Define the Lipschitz function $\mathbf{F} = \mathbf{g}_2 \circ \mathbf{f} \circ \mathbf{g}_1 : V \to \mathbb{R}^m$, where $V = {\mathbf{g}_1}^{-1}(U) \subset \mathbb{R}^n$ is open. Then

$$\partial \mathbf{F}(\mathbf{x}) = \mathbf{P}_2 \partial \mathbf{f}(\mathbf{g}_1(\mathbf{x}))\mathbf{P}_1, \quad \forall \mathbf{x} \in V. \tag{2.7}$$

Proof. Let $\Omega_{\mathbf{F}} \subset V$ and $\Omega_{\mathbf{f}} \subset U$ be the measure zero sets where $\mathbf{F}$ and $\mathbf{f}$ are not differentiable. Since $\mathbf{g}_1, \mathbf{g}_2$ are invertible, $\mathbf{f} = {\mathbf{g}_2}^{-1} \circ \mathbf{F} \circ {\mathbf{g}_1}^{-1} : U \to \mathbb{R}^m$. Therefore, $\mathbf{x}_i \in V \setminus \Omega_{\mathbf{F}}$ if and only if $\mathbf{y}_i = \mathbf{g}_1(\mathbf{x}_i) \in U \setminus \Omega_{\mathbf{f}}$. We can apply the Chain Rule to $\mathbf{F}$ at $\mathbf{x}_i \in V \setminus \Omega_{\mathbf{F}}$ to yield $\mathbf{JF}(\mathbf{x}_i) = \mathbf{P}_2\mathbf{Jf}(\mathbf{g}_1(\mathbf{x}_i))\mathbf{P}_1$. Moreover, from continuity of both $\mathbf{g}_1$ and ${\mathbf{g}_1}^{-1}$, $\mathbf{x}_i \to \mathbf{x}$ if and only if $\mathbf{y}_i = \mathbf{g}_1(\mathbf{x}_i) \to \mathbf{g}_1(\mathbf{x})$. Then, given $\mathbf{x} \in V$,

$$\begin{aligned}
\partial^B \mathbf{F}(\mathbf{x}) &= \left\{ \lim_{i\to\infty} \mathbf{JF}(\mathbf{x}_i) :\ \mathbf{x}_i \to \mathbf{x},\ \mathbf{x}_i \in V \setminus \Omega_{\mathbf{F}} \right\} && (2.8)\\
&= \left\{ \lim_{i\to\infty} \mathbf{P}_2\mathbf{Jf}(\mathbf{y}_i)\mathbf{P}_1 :\ \mathbf{y}_i \to \mathbf{g}_1(\mathbf{x}),\ \mathbf{y}_i \in U \setminus \Omega_{\mathbf{f}} \right\} = \mathbf{P}_2 \partial^B \mathbf{f}(\mathbf{g}_1(\mathbf{x}))\mathbf{P}_1, && (2.9)
\end{aligned}$$

and

$$\begin{aligned}\partial \mathbf{F}(\mathbf{x}) &= \left\{ \sum_i \lambda_i \mathbf{M}_i : \ \mathbf{M}_i \in \partial^B \mathbf{F}(\mathbf{x}), \lambda_i \geq 0, \sum_i \lambda_i = 1 \right\} \qquad (2.10)\\ &= \left\{ \mathbf{P}_2 \Big(\sum_i \lambda_i \mathbf{N}_i \Big) \mathbf{P}_1 : \ \mathbf{N}_i \in \partial^B \mathbf{f}(\mathbf{g}_1(\mathbf{x})), \lambda_i \geq 0, \sum_i \lambda_i = 1 \right\} = \mathbf{P}_2 \partial \mathbf{f}(\mathbf{g}_1(\mathbf{x}))\mathbf{P}_1.\end{aligned}$$

□

2.1.2. Piecewise-differentiable (PC^r) functions

Nonsmooth models for most chemical engineering processes, including distillation systems as analyzed in this thesis and multistream heat exchangers [85], can be formulated in terms of piecewise-smooth (PC^∞) functions. In this section we define piecewise-differentiable (PC^r) functions as established by Scholtes in [78], and summarize the main properties that will be relevant to prove the PC^r Rank Theorem in Chapter 6.

Definition 2.1.4 (**PC^r functions**)**.** Let $U_0 \subset \mathbb{R}^n$ be open and $r \in \{1, 2, \ldots, \infty\}$. A function $\mathbf{f} : U_0 \to \mathbb{R}^m$ is said to be PC^r at $\mathbf{x}_0 \in U_0$ if there exist a neighborhood $U \subset U_0$ of $\mathbf{x}_0$ and finitely many C^r selection functions $\mathbf{f}_{(1)}, \ldots, \mathbf{f}_{(k)} : U \to \mathbb{R}^m$ such that $\mathbf{f}$ is continuous on U and $\mathbf{f}(\mathbf{x}) \in \{\mathbf{f}_{(1)}(\mathbf{x}), \ldots, \mathbf{f}_{(k)}(\mathbf{x})\}$ for every $\mathbf{x} \in U$. $\mathbf{f}$ is said to be a PC^r function if it is PC^r at every point of U_0. Moreover, if the C^r selection functions are linear [affine], we say $\mathbf{f}$ is piecewise-linear [piecewise-affine].

As illustrated in Figure 2.1, conceptually the domain of $\mathbf{f}$ around $\mathbf{x}^0$ can be subdivided into regions where $\mathbf{f}$ is equal to a C^r selection function $\mathbf{f}_{(i)}$.

Piecewise-differentiable (PC^r) functions are locally Lipschitz (Corollary 4.1.1 in [78]), while the converse is not true in general. For instance, the Euclidean norm is locally Lipschitz but can only be expressed at the origin using infinitely many selection functions and is thus not PC^r. Examples of PC^r functions include max, min, abs, and C^r functions. PC^r functions are semismooth (see [27] for a definition of the latter). As locally Lipschitz functions, PC^r functions can only fail to have a well-defined derivative on a "small" set $\Omega_\mathbf{f}$ (i.e., with Lebesgue measure zero). In Figure 2.1, $\Omega_\mathbf{f}$ could correspond, at most, to the boundaries between regions.

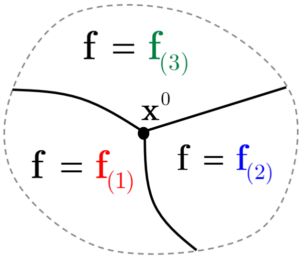

$\mathbf{f} = \mathbf{f}_{(3)}$
$\mathbf{x}^0$
$\mathbf{f} = \mathbf{f}_{(1)}$
$\mathbf{f} = \mathbf{f}_{(2)}$

Figure 2.1: A possible representation of the domain of a PC^r function $\mathbf{f}$.

Given a set of selection functions $\mathbf{f}_{(1)}, \ldots, \mathbf{f}_{(k)}$, we can find a subset of indices $I^e_\mathbf{f}(\mathbf{x}_0) \subset \{1, \ldots, k\}$ such that $\mathbf{x}_0 \in \mathrm{cl}(\mathrm{int}(U_i))$ for every $i \in I^e_\mathbf{f}(\mathbf{x}_0)$, where $U_i = \{\mathbf{x} \in U : \mathbf{f}(\mathbf{x}) = \mathbf{f}_{(i)}(\mathbf{x})\}$. The functions $\{\mathbf{f}_{(i)} : i \in I^e_\mathbf{f}(\mathbf{x}_0)\}$ are said to be essentially active at $\mathbf{x}_0$, and they form a set of selection functions for $\mathbf{f}$ on a potentially smaller neighborhood of $\mathbf{x}_0$ (see proof of Proposition 4.1.1 in [78]). Moreover, they can be used to characterize the B-subdifferential and directional derivative of $\mathbf{f}$ (see [78]):

$$\partial^B \mathbf{f}(\mathbf{x}_0) = \left\{ \mathbf{J}\mathbf{f}_{(i)}(\mathbf{x}_0) : i \in I^{\mathrm{e}}_\mathbf{f}(\mathbf{x}_0) \right\}, \tag{2.11}$$

$$d\mathbf{f}_{\mathbf{x}_0}(\mathbf{d}) \in \left\{ \mathbf{J}\mathbf{f}_{(i)}(\mathbf{x}_0)\mathbf{d} : i \in I^{\mathrm{e}}_\mathbf{f}(\mathbf{x}_0) \right\} = \left\{ \mathbf{M}\mathbf{d} : \mathbf{M} \in \partial^B \mathbf{f}(\mathbf{x}_0) \right\}, \tag{2.12}$$

where $d\mathbf{f}_{\mathbf{x}_0} : \mathbb{R}^n \to \mathbb{R}^n$ is a piecewise-linear function whose B-subdifferential at any $\mathbf{d} \in \mathbb{R}^n$ is a subset of $\partial^B \mathbf{f}(\mathbf{x}_0)$.

As the next proposition demonstrates, PC^r functions are closed under composition, unlike piecewise-differentiable functions in the sense of Whitehead [89].

Proposition 2.1.5 (Composition of PC^r functions). Let $U \subset \mathbb{R}^n, V \subset \mathbb{R}^m$ be open sets. If $\mathbf{f} : U \to \mathbb{R}^m$ is PC^r at $\mathbf{x}_0 \in U$ and $\mathbf{g} : V \to \mathbb{R}^p$ is PC^r at $\mathbf{f}(\mathbf{x}_0) \in V$, then $\mathbf{g} \circ \mathbf{f}$ is

PC^r at $\mathbf{x}_0$ and

$$\partial^B(\mathbf{g} \circ \mathbf{f})(\mathbf{x}_0) \subset \left\{\mathbf{M}_1\mathbf{M}_2 : \mathbf{M}_1 \in \partial^B\mathbf{g}(\mathbf{f}(\mathbf{x}_0)),\ \mathbf{M}_2 \in \partial^B\mathbf{f}(\mathbf{x}_0)\right\}. \tag{2.13}$$

Proof.

If $\{\mathbf{f}_{(i)} : U' \to \mathbb{R}^m :\ i \in \{1, \ldots, k\}\}$ and $\{\mathbf{g}_{(j)} : V' \to \mathbb{R}^p :\ j \in \{1, \ldots, q\}\}$ are selection functions for $\mathbf{f}$ at $\mathbf{x}_0$ and for $\mathbf{g}$ at $\mathbf{f}(\mathbf{x}_0)$, then we can shrink the neighborhoods U', V' such that $\left\{\mathbf{f}_{(i)} : U' \to \mathbb{R}^m :\ i \in I^{\mathrm{e}}_{\mathbf{f}}(\mathbf{x}_0)\right\}$ and $\left\{\mathbf{g}_{(j)} : V' \to \mathbb{R}^p :\ j \in I^{\mathrm{e}}_{\mathbf{g}}(\mathbf{f}(\mathbf{x}_0)\right\}$ are selection functions for $\mathbf{f}$ and $\mathbf{g}$. Then $\mathbf{g} \circ \mathbf{f}$ is continuous by composition and it admits the C^r selection functions $\left\{\mathbf{g}_{(j)} \circ \mathbf{f}_{(i)} :\ i \in I^{\mathrm{e}}_{\mathbf{f}}(\mathbf{x}_0), j \in I^{\mathrm{e}}_{\mathbf{g}}(\mathbf{f}(\mathbf{x}_0))\right\}$. Applying the Chain Rule to each $\mathbf{g}_{(j)} \circ \mathbf{f}_{(i)}$,

$$\partial^B(\mathbf{g} \circ \mathbf{f})(\mathbf{x}_0) \subset \left\{\mathbf{J}\mathbf{g}_{(j)}(\mathbf{f}(\mathbf{x}_0))\mathbf{J}\mathbf{f}_{(i)}(\mathbf{x}_0) : i \in I^{\mathrm{e}}_{\mathbf{f}}(\mathbf{x}_0),\ j \in I^{\mathrm{e}}_{\mathbf{g}}(\mathbf{f}(\mathbf{x}_0))\right\}. \tag{2.14}$$

□

The following concepts relate to the PC^r Inverse Function Theorem (see Section 2.2).

Definition 2.1.6 (**Coherent and complete coherent orientation**)**.** Let $U \subset \mathbb{R}^n$ be open, $\mathbf{f} : U \to \mathbb{R}^m$ be a PC^r function at $\mathbf{x}_0 \in U$, and $m \leq n$. We say $\mathbf{f}$ is coherently oriented with respect to the first m variables at $\mathbf{x}_0$ if all matrices in $\boldsymbol{\pi}^B_m\mathbf{f}(\mathbf{x}_0)$ have the same non-zero determinant sign. We say $\mathbf{f}$ is completely coherently oriented with respect to the first m variables at $\mathbf{x}_0$ if all matrices in the set

$$\left\{\mathbf{M} \in \mathbb{R}^{m \times m} :\ \text{each } i\text{-th row of } \mathbf{M} \text{ equals the } i\text{-th row of some } \mathbf{N} \in \boldsymbol{\pi}^B_m\mathbf{f}(\mathbf{x}_0)\right\} \tag{2.15}$$

(i.e., the set of row-by-row permutations of $\boldsymbol{\pi}^B_m\mathbf{f}(\mathbf{x}_0)$) have the same non-zero determinant sign. If $m = n$, we simply say $\mathbf{f}$ is (completely) coherently oriented at $\mathbf{x}_0$.

2.1.3. Automatic LD-differentiation

Lexicographically smooth (L-smooth) functions [64] are a subclass of locally Lipschitz functions for which high-order directional derivatives are well defined. More precisely, $\mathbf{f}$ is L-smooth at $\mathbf{x}^0 \in X$ if the following sequence of functions is well-defined for any directions matrix $\mathbf{M} := [\mathbf{m}_1 \ ... \ \mathbf{m}_k] \in \mathbb{R}^{n\times k}$, $k \in \mathbb{N}$:

$$\mathbf{f}^{(0)}_{\mathbf{x}^0,\mathbf{M}} : \mathbb{R}^n \to \mathbb{R}^m \ \ : \mathbf{d} \mapsto \mathbf{f}'(\mathbf{x}^0, \mathbf{d}), \qquad \mathbf{f}^{(1)}_{\mathbf{x}^0,\mathbf{M}} : \mathbb{R}^n \to \mathbb{R}^m \ \ : \mathbf{d} \mapsto [\mathbf{f}^{(0)}_{\mathbf{x}^0,\mathbf{M}}]'(\mathbf{m}_1, \mathbf{d}), \ \ \ldots \ \ ,$$
$$\mathbf{f}^{(k)}_{\mathbf{x}^0,\mathbf{M}} : \mathbb{R}^n \to \mathbb{R}^m \ \ : \mathbf{d} \mapsto [\mathbf{f}^{(k-1)}_{\mathbf{x}^0,\mathbf{M}}]'(\mathbf{m}_k, \mathbf{d}).$$

The class of L-smooth functions is closed under composition and includes C^r, PC^r and convex functions such as the Euclidean norm.

The lexicographic directional derivative (LD-derivative), recently introduced by Khan and Barton [45, 46], is a computationally relevant generalized derivative for L-smooth functions. The LD-derivative of $\mathbf{f}$ at $\mathbf{x}^0$ in the directions $\mathbf{M} \in \mathbb{R}^{n\times k}$ is uniquely defined as

$$\mathbf{f}'(\mathbf{x}^0; \mathbf{M}) := \left[\mathbf{f}^{(0)}_{\mathbf{x}^0,\mathbf{M}}(\mathbf{m}_1) \quad \mathbf{f}^{(1)}_{\mathbf{x}^0,\mathbf{M}}(\mathbf{m}_2) \quad \ldots \quad \mathbf{f}^{(k-1)}_{\mathbf{x}^0,\mathbf{M}}(\mathbf{m}_k)\right]. \tag{2.16}$$

As demonstrated in [46, 8], analytical expressions for the LD-derivative of elementary L-smooth functions (such as abs, max, min, norms) can be derived. For instance, for $f(x) = \text{abs}(x)$, the directions matrix is a vector $\mathbf{m} \in \mathbb{R}^{1\times k}$, and

$$f'(x; \mathbf{m}) = \text{fsign}(x, \mathbf{m}^{\text{T}}) \cdot \mathbf{m}, \tag{2.17}$$

where $\text{fsign}(x, \mathbf{m}^{\text{T}})$ returns the sign (1 or -1) of the first non-zero element of the concatenated vector $(x, \mathbf{m}^{\text{T}})$ or zero if the latter is the zero vector.

When the chosen directions matrix $\mathbf{M}$ is square and invertible (e.g., the identity matrix $\mathbf{I}_n$), the LD-derivative can be used to obtain an element of the plenary Jacobian of $\mathbf{f}$ at

$\mathbf{x}^0$, denoted here by $\mathbf{A}$ [45]:

$$\mathbf{f}'(\mathbf{x}^0;\mathbf{M}) = \mathbf{A}\cdot\mathbf{M}. \tag{2.18}$$

By definition, an element $\mathbf{A}$ of the plenary Jacobian is indistinguishable from an element of the Clarke Jacobian in any matrix-vector product, which is usually all that is required within equation solving methods (see Section 2.3); if f is scalar-valued, $\mathbf{A} \in \partial f(\mathbf{x})$. Additionally, if $\mathbf{f}$ is PC^r, which is the case for the distillation models presented in this thesis, $\mathbf{A}$ is specifically an element of the B-subdifferential [46].

For directionally differentiable nonsmooth functions (such as the PC^r class), one might naively think that the LD-derivative in the coordinate directions (i.e., using $\mathbf{M} = \mathbf{I}_n$) would yield the same result as simply concatenating the directional derivatives in the coordinate directions. However, that is not necessarily the case, i.e., in general,

$$\mathbf{f}'(\mathbf{x}^0;\mathbf{I}_n) \neq \left[\mathbf{f}'(\mathbf{x}^0;\mathbf{e}_1)\ldots\mathbf{f}'(\mathbf{x}^0;\mathbf{e}_n)\right], \tag{2.19}$$

unless $\mathbf{f}$ is differentiable at $\mathbf{x}^0$. Additionally, the set of non-differentiable points $Z_{\mathbf{f}}$, where the two constructions potentially differ, is reachable within finite-precision arithmetic. This was demonstrated numerically by case studies in [85, 8].

The distinctive feature of the LD-derivative is that a strict chain rule applies, with $\mathbf{M}$ not necessarily invertible or square, which allows for the use of automatic differentiation (AD) to compute exact LD-derivatives of L-smooth factorable (L-factorable) functions. Conventional AD [32] associates with each elemental smooth function (such as +, -, cos(x)) not only its evaluation but also the simultaneous computation of its analytical derivative, which can be implemented via operator overloading (e.g., see [63]). Application of the chain rule allows the computation of exact derivatives for differentiable factorable functions, i.e., functions that can be computed from a finite number of operations, fixed a priori, with elemental smooth functions on a computer – this excludes "if-else" statements and "while" loops. By including the LD-differentiation rules for elemental L-smooth operators, Khan and Barton [46] extended AD to include L-factorable functions. In particular, this is the first method that can be used to obtain exact B-subdifferential elements for PC^r functions.

2.2. Inverse Function Theorems

Since PC^r functions are closed under composition and include the identity, the set of open subsets of $\mathbb{R}^n$ together with the set of PC^r functions defines the PC^r category. The next definition characterizes homeomorphisms in terms of a given generic category or "class" of functions $\mathcal{G}$; in this thesis we will be considering $\mathcal{G} = C^r, PC^r$, locally Lipschitz.

Definition 2.2.1 (**$\mathcal{G}$ homeomorphisms**)**.** Let $\mathcal{G}$ represent a category of functions defined on open subsets of $\mathbb{R}^n$. A function $\mathbf{f} : U \to V$ between open sets $U, V \subset \mathbb{R}^n$ is said to be a $\mathcal{G}$ homeomorphism if it is a homeomorphism and both $\mathbf{f}$ and $\mathbf{f}^{-1}$ are $\mathcal{G}$ functions. We say $\mathbf{f}$ is a local $\mathcal{G}$ homeomorphism at $\mathbf{x}_0 \in U$ if there exist (open) neighborhoods $A \subset U$ of $\mathbf{x}_0$ and $B \subset V$ of $\mathbf{f}(\mathbf{x}_0)$ such that $\mathbf{f}|_A : A \to B$ is a $\mathcal{G}$ homeomorphism.

Inverse Function Theorems are used as a basis to prove the Implicit Function and the Rank Theorems. The Inverse Function Theorem "version" that we decide to use, with its particular conditions, will indirectly give rise to the conditions of these other "offspring" theorems. It is thus preferable to use an "if and only if" (iff) Inverse Function Theorem, which states necessary and sufficient conditions for the existence of a local homeomorphism, instead of an "if" theorem stating only sufficient conditions. As presented below, iff Inverse Theorems exist for all the function classes we are interested in: C^r, PC^r, and locally Lipschitz continuous functions.

Most commonly, the C^r Inverse Function Theorem is presented as an "if" theorem only. However, its converse is essentially immediate, so we present it in its "iff" version below.

Theorem 2.2.2 (**C^r "iff" Inverse Function Theorem**)**.** Let $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ be a C^r function. $\mathbf{f}$ is a local C^r homeomorphism at $\mathbf{x}_0 \in \mathbb{R}^n$ if and only if $\mathbf{Jf}(\mathbf{x}_0) \in \mathbb{R}^{n \times n}$ is invertible.

The "iff" version of the Inverse Theorem for locally Lipschitz continuous functions, which is stated in terms of the generalized Thibault derivative

$$\Delta\mathbf{f}(\mathbf{x}_0; \mathbf{d}) = \left\{ \mathbf{z} \in \mathbb{R}^m : \ \mathbf{z} = \lim_{k \to \infty} \frac{\mathbf{f}(\mathbf{x}_k + \lambda_k \mathbf{d}) - \mathbf{f}(\mathbf{x}_k)}{\lambda_k}, \ \{\mathbf{x}_k\} \to \mathbf{x}_0, \ \{\lambda_k\} \to 0^+ \right\}, \tag{2.20}$$

was introduced in Theorem 1.1 of [52].

Theorem 2.2.3 (**Kummer's Lipschitz "iff" Inverse Function Theorem**)**.** Let $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ be a locally Lipschitz function. $\mathbf{f}$ is a local Lipschitz homeomorphism at $\mathbf{x}_0 \in \mathbb{R}^n$ if and only if $\mathbf{0} \notin \Delta\mathbf{f}(\mathbf{x}_0; \mathbf{d})$ for every non-zero $\mathbf{d} \in \mathbb{R}^n$.

However, the set $\Delta\mathbf{f}(\mathbf{x}_0; \mathbf{d})$ is more abstract and much less used than the Clarke Jacobian $\partial\mathbf{f}(\mathbf{x}_0)$ for Lipschitz functions. We can use the latter generalized derivative set to express Clarke's Inverse Function Theorem for Lipschitz functions (Theorem 7.1.1 in [22]), at the cost of it being an "if" theorem only. Indeed, Clarke's condition is not necessary even in the case of piecewise-linear functions, as demonstrated in Example 2.2 of [52].

Theorem 2.2.4 (**Clarke's Lipschitz "if" Inverse Function Theorem**)**.** Let $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ be a locally Lipschitz function. $\mathbf{f}$ is a local Lipschitz homeomorphism at $\mathbf{x}_0 \in \mathbb{R}^n$ if $\mathbf{f}$ is Clarke regular at $\mathbf{x}_0$ (i.e., all matrices in $\partial\mathbf{f}(\mathbf{x}_0)$ are invertible).

The PC^r "iff" Inverse Function Theorem was first presented in Theorem 5 of [72], with three equivalent sets of necessary and sufficient conditions. We present this theorem below with the most relevant and concrete set of conditions.

Theorem 2.2.5. **(PC^r "iff" Inverse Function Theorem)** Let $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ be a PC^r function. $\mathbf{f}$ is a local PC^r homeomorphism at $\mathbf{x}_0 \in \mathbb{R}^n$ if and only if $\mathbf{f}$ is coherently oriented at $\mathbf{x}_0$ and $d\mathbf{f}_{\mathbf{x}_0} : \mathbb{R}^n \to \mathbb{R}^n$ is invertible.

The above theorem transforms the task of judging local invertibility of the PC^r function $\mathbf{f}$ at $\mathbf{x}_0$ into that of judging global invertibility of the piecewise-linear (PL) function $d\mathbf{f}_{\mathbf{x}_0}$. However, the latter is still a hard task for which no "iff" theorem exists. According to Corollary 19 from [72], a piecewise-affine (PA) function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ is a (global) PA homeomorphism if it is (everywhere) completely coherently oriented (see Definition 2.1.6). Given that $\partial^B(d\mathbf{f}_{\mathbf{x}_0})(\mathbf{d}) \subset \partial^B\mathbf{f}(\mathbf{x}_0)$ for every $\mathbf{d} \in \mathbb{R}^n$ (see Proposition 4.1.3 in [78]), if $\mathbf{f}$ is completely coherently oriented at $\mathbf{x}_0$ then $d\mathbf{f}_{\mathbf{x}_0}$ is completely coherently oriented everywhere and is thus invertible. This result gives rise to the following "if" Inverse Theorem for PC^r functions.

Theorem 2.2.6. **(PC^r "if" Inverse Function Theorem)** Let $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ be a PC^r function. $\mathbf{f}$ is a local PC^r homeomorphism at $\mathbf{x}_0 \in \mathbb{R}^n$ if $\mathbf{f}$ is completely coherently oriented at $\mathbf{x}_0$.

Since a PC^r function is a local PC^r homeomorphism at $\mathbf{x}_0$ if and only if it is a local Lipschitz homeomorphism at $\mathbf{x}_0$ (Proposition 4.2.1 in [78]), for PC^r functions the necessary and sufficient PC^r conditions are equivalent to Kummer's condition. The advantage of the former is that they utilize the B-subdifferential, a finite set with a concrete representation for PC^r functions in terms of essentially active C^r functions. As with Lipschitz functions, Clarke regularity is a sufficient but not necessary condition.

Though not directly related to Inverse Function Theorems, the concept of "slices" of open subsets of $\mathbb{R}^n$ (e.g., see [54]) will be used to formulate parts (b) and (c) of the Rank Theorem 6.4.2.

Definition 2.2.7 (Slices of open sets). Let $U \subset \mathbb{R}^n$ be open and $1 \leq k \leq n$ be an integer. A k-dimensional slice, or k-slice, of U is any subset of the form

$$U' = \left\{(x_1, \ldots, x_n) \in U : \; x_{i_1} = c_1, x_{i_2} = c_2, \ldots, x_{i_{n-k}} = c_{n-k}\right\}, \tag{2.21}$$

where $\mathbf{c} = (c_1, \ldots, c_{n-k}) \in \mathbb{R}^{n-k}$ is a constant, and $i_1, \ldots, i_{n-k} \in \{1, \ldots, n\}$ are increasing indices.

Note that the k-slice of U described above is the intersection of U with the k-dimensional hyperplane defined by fixing the $n-k$ coordinates $i_1, \ldots, i_{n-k}$ in $\mathbb{R}^n$ at the constant values $c_1, \ldots, c_{n-k}$.

Proposition 2.2.8 (Properties of k-slices). Let $U \subset \mathbb{R}^n$ be open, $1 \leq k \leq n$ be an integer, and $U' \subset \mathbb{R}^n$ be a k-slice of U represented without loss of generality as

$$U' = \{(x_1, \ldots, x_n) \in U : \; x_{k+1} = c_1, \ldots, x_n = c_{n-k}\}. \tag{2.22}$$

Then:

1) The projection of U' onto the first k coordinates, $\boldsymbol{\rho}^n_k(U')$, is an open subset of $\mathbb{R}^k$.

2) U' is homeomorphic to $\boldsymbol{\rho}^n_k(U')$.

3) $\boldsymbol{\rho}^n_k|_{U'} : U' \to \boldsymbol{\rho}^n_k(U')$ is an extended linear homeomorphism.

Proof. 1) Let $\mathbf{x} \in \boldsymbol{\rho}^n_k(U')$. Then $(\mathbf{x}, \mathbf{c}) = (\mathbf{x}, c_1, \ldots, c_{n-k}) \in U$. Since $U \subset \mathbb{R}^n$ is open, there exists an open cube $C_1 \times C_2 \subset U$ containing $(\mathbf{x}, \mathbf{c})$, where $C_1 \subset \mathbb{R}^k$, $C_2 \subset \mathbb{R}^{n-k}$ are open cubes. Then $C_1 \subset \boldsymbol{\rho}^n_k(U')$ is an open set in $\mathbb{R}^k$ containing $\mathbf{x}$. Since $\mathbf{x}$ was arbitrary, we conclude $\boldsymbol{\rho}^n_k(U') \subset \mathbb{R}^k$ is open.

2) The projection $\boldsymbol{\rho}^n_k : \mathbb{R}^n \to \mathbb{R}^k$ is a continuous and linear function, therefore its restriction to the open subset $U' \subset \mathbb{R}^k$, $\boldsymbol{\rho}^n_k|_{U'} : U' \to \boldsymbol{\rho}^n_k(U')$, is also continuous and linear. Define the inclusion $\mathbf{i}^k_n : \mathbb{R}^k \to \mathbb{R}^n$ as

$$\mathbf{i}^k_n(x_1, \ldots, x_k) = (x_1, \ldots, x_k, c_1, \ldots, c_{n-k}), \tag{2.23}$$

which is clearly continuous and linear. Then we can see that its restriction $\mathbf{i}_n^k|_{\boldsymbol{\rho}_k^n(U')}$: $\boldsymbol{\rho}_k^n(U') \to U'$, which remains continuous, is the inverse of $\boldsymbol{\rho}_k^n|_{U'} : U' \to \boldsymbol{\rho}_k^n(U')$.

3) $\boldsymbol{\rho}_k^n|_{U'} : U' \to \boldsymbol{\rho}_k^n(U')$ is linear, and its inverse has the global linear extension $\mathbf{i}_n^k$: $\mathbb{R}^k \to \mathbb{R}^n$. □

In other words, every k-slice of an open subset of $\mathbb{R}^n$ is linearly homeomorphic to an open subset of $\mathbb{R}^k$, namely, its projection onto the k non-fixed coordinates. The slice $U' \subset \mathbb{R}^n$ is not open, but $\boldsymbol{\rho}_k^n(U') \subset \mathbb{R}^k$ is.

The following projection and inclusion mappings will be relevant to express and prove the Rank Theorems in Chapter 6.

Definition 2.2.9. Let n, m be integers greater than or equal to 1 with $n \geq m$.

Define $\boldsymbol{\rho}_m^n : \mathbb{R}^n \to \mathbb{R}^m$ as the projection from $\mathbb{R}^n$ onto the first m coordinates,

$$\boldsymbol{\rho}_m^n(x_1, \ldots, x_m, x_{m+1}, \ldots, x_n) = (x_1, \ldots, x_m), \tag{2.24}$$

$\boldsymbol{\pi}_{n-m}^n : \mathbb{R}^n \to \mathbb{R}^{n-m}$, as the projection from $\mathbb{R}^n$ onto the last $n - m$ coordinates,

$$\boldsymbol{\pi}_{n-m}^n(x_1, \ldots, x_m, x_{m+1}, \ldots, x_n) = (x_{m+1}, \ldots, x_n), \tag{2.25}$$

and $\boldsymbol{\iota}_n^m : \mathbb{R}^m \to \mathbb{R}^n$ as the inclusion of $\mathbb{R}^m$ as the first coordinates of $\mathbb{R}^n$,

$$\boldsymbol{\iota}_n^m(x_1, \ldots, x_m) = (x_1, \ldots, x_m, 0, \ldots, 0). \tag{2.26}$$

2.3. Nonlinear equation solving methods

The standard format for a system of nonlinear equations is

$$\mathbf{f}(\mathbf{x}) = \mathbf{0}, \tag{2.27}$$

where $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ and the number of equations m is not necessarily equal to number of variables n. We say that the system (2.27) is square if $n = m$, and non-square otherwise.

2.3.1. The Newton method

The standard Newton method can be used to solve Equation 2.27 for a smooth function $\mathbf{f}$ with $n = m$. We start from a current solution estimate $\mathbf{x}^k$ and generate the next iterate $\mathbf{x}^{k+1}$ by solving the linear system

$$\mathbf{Jf}(\mathbf{x}^k)(\mathbf{x}^{k+1} - \mathbf{x}^k) = -\mathbf{f}(\mathbf{x}^k). \tag{2.28}$$

If $\left\|\mathbf{f}(\mathbf{x}^{k+1})\right\|$ is smaller than a desired tolerance $\epsilon > 0$, we take $\mathbf{x}^{k+1}$ as an approximate solution of Equation 2.27. Otherwise, we set $k \leftarrow k+1$ and repeat this process. The Newton method has Q-quadratic convergence in a neighborhood of a solution $\mathbf{x}^*$ of Equation 2.27 if $\mathbf{f}$ is a C^2 function, provided $\mathbf{Jf}(\mathbf{x}^*)$ is invertible (e.g., see Theorem 8.6.5 in [9]).

2.3.2. The semismooth Newton method

The semismooth Newton method [69] naturally extends the Newton method to locally Lipschitz functions by using an element of the Clarke Jacobian instead of $\mathbf{Jf}(\mathbf{x}^k)$ in Equation 2.28:

$$\mathbf{G}(\mathbf{x}^k)(\mathbf{x}^{k+1} - \mathbf{x}^k) = -\mathbf{f}(\mathbf{x}^k), \qquad \mathbf{G}(\mathbf{x}^k) \in \partial\mathbf{f}(\mathbf{x}^k). \tag{2.29}$$

If $\partial\mathbf{f}(\mathbf{x}^*)$ contains no singular matrices at a solution $\mathbf{x}^*$ of Equation 2.27 and $\mathbf{f}$ is semismooth (strongly semismooth), then the semismooth Newton Method exhibits Q-superlinear (Q-quadratic) convergence in a neighborhood of $\mathbf{x}^*$. More importantly for the nonsmooth models considered in this thesis, convergence is Q-quadratic if $\mathbf{f}$ is PC^r and $\mathbf{G}(\mathbf{x}^k) \in \partial^B\mathbf{f}(\mathbf{x}^k)$, provided that $\partial^B\mathbf{f}(\mathbf{x}^*)$ contains no singular matrices [48]. Since the B-subdifferential of a PC^r function contains the finitely many "limiting" Jacobians of the C^r essentially active selection functions, the non-singularity condition on $\partial^B\mathbf{f}(\mathbf{x}^*)$

is much looser and more easily verifiable than that on $\partial \mathbf{f}(\mathbf{x}^*)$. A simple example of that is the absolute function $f(x) = |x|$, for which $\partial^B f(0) = \{-1, 1\}$ does not contain zero while $\partial f(0) = [-1, 1]$ does. B-subdifferential elements of PC^r functions can be computed exactly using the algorithm of Khan and Barton [46].

However, neither the standard nor the semismooth Newton methods can be applied if $m \neq n$, or if $m = n$ but $\mathbf{Jf}(\mathbf{x}^k)$ or $\mathbf{G}(\mathbf{x}^k)$ is singular (or very ill-conditioned) at any given iteration k.

2.3.3. The linear programming Newton method

The linear programming Newton (LP-Newton) method [28] can be used to solve constrained and potentially non-square equation systems of the form

$$\begin{aligned} &\mathbf{f}(\mathbf{x}) = \mathbf{0} \\ &\text{s.t.} \quad \mathbf{x} \in X, \end{aligned} \tag{2.30}$$

where $\mathbf{f}$ is locally Lipschitz continuous and X is closed and given by a set of polyhedral bounds. Starting from a point $\mathbf{x}^k$, the next iterate $\mathbf{x}^{k+1}$ is obtained as the $\mathbf{x}$ part of the solution of the following linear program:

$$\begin{aligned} \min_{\mathbf{x},\gamma} \ &\gamma \\ \text{s.t.} \ &\left\|\mathbf{f}(\mathbf{x}^k) + \mathbf{G}(\mathbf{x}^k)(\mathbf{x} - \mathbf{x}^k)\right\|_\infty \le \gamma \left\|\mathbf{f}(\mathbf{x}^k)\right\|_\infty^2, \\ &\left\|\mathbf{x} - \mathbf{x}^k\right\|_\infty \le \gamma \left\|\mathbf{f}(\mathbf{x}^k)\right\|_\infty, \\ &\mathbf{x} \in X, \end{aligned} \tag{2.31}$$

where $\mathbf{G}(\mathbf{x}^k)$ is a suitable substitute for $\mathbf{Jf}(\mathbf{x}^k)$ when $\mathbf{f}$ is nonsmooth, e.g., $\mathbf{G}(\mathbf{x}^k) \in \partial \mathbf{f}(\mathbf{x}^k)$ or $\mathbf{G}(\mathbf{x}^k) \in \partial^B \mathbf{f}(\mathbf{x}^k)$. In constrast to the semismooth Newton method, the LP-Newton step remains well-defined when $\mathbf{G}(\mathbf{x}^k)$ is singular or non-square. In [28], Facchinei et al. showed that the method exhibits Q-quadratic convergence in the neighborhood of a solution $\mathbf{x}^*$ of Equation 2.30 under certain conditions, which are not easily verifiable in general. In particular, Q-quadratic convergence holds when we use $\mathbf{G}(\mathbf{x}^k) \in \partial^B \mathbf{f}(\mathbf{x}^k)$ in (2.31) if all matrices in $\partial^B \mathbf{f}(\mathbf{x}^*)$ have full column rank n and an additional condition (Condition 2 in [28]) is satisfied (see Corollary 2 in [28]). The conditions under which Q-quadratic convergence is guaranteed may be satisfied for some systems with non-isolated solutions, as illustrated in [28] for a KKT system example.

The LP-Newton method can also be equipped with a backtracking line search to ensure global convergence under certain assumptions [30]. To implement the latter with chosen constants $\theta, \sigma \in (0,1)$, at each iteration k we compute the LP-Newton step $\mathbf{d}^k = \mathbf{x}^{k+1} - \mathbf{x}^k$ by solving (2.31) and then we evaluate

$$\Delta(\mathbf{x}^k) = -\left\|\mathbf{f}(\mathbf{x}^k)\right\| \left(1 - \gamma^k \left\|\mathbf{f}(\mathbf{x}^k)\right\|\right).$$

Starting with $\alpha = 1$ we recursively reset $\alpha \leftarrow \theta\alpha$, if needed, until the expression

$$\left\|\mathbf{f}(\mathbf{x}^k + \alpha\mathbf{d}^k)\right\| \leq \left\|\mathbf{f}(\mathbf{x}^k)\right\| + \sigma\alpha\Delta(\mathbf{x}^k)$$

is satisfied. Then, the next iterate is computed as $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha\mathbf{d}^k$.

In our simulations we have used $\theta = 0.6$ and $\sigma = 0.1$.

2.3.4. The pseudoinverse Newton method

The Newton method step for smooth functions (Equation 2.28) is only well defined when $n = m$ and $\mathbf{Jf}(\mathbf{x}^k)$ is invertible; therefore, it is equivalent to

$$\mathbf{x}^{k+1} := \mathbf{x}^k - \mathbf{Jf}(\mathbf{x}^k)^{-1}\,\mathbf{f}(\mathbf{x}^k). \tag{2.32}$$

If $\mathbf{Jf}(\mathbf{x}^k)$ is singular or if $m \neq n$ we can use the pseudoinverse (i.e., Moore-Penrose inverse) of the Jacobian matrix in place of its inverse:

$$\mathbf{x}^{k+1} := \mathbf{x}^{k} - \mathbf{Jf}(\mathbf{x}^{k})^{\dagger}\,\mathbf{f}(\mathbf{x}^{k}). \tag{2.33}$$

This defines what we refer to as the pseudoinverse Newton method. The pseudoinverse of a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ is the unique matrix $\mathbf{A}^{\dagger} \in \mathbb{R}^{n\times m}$ that satisfies the following four relationships:

$$\mathbf{A}^{\dagger}\mathbf{A}\mathbf{A}^{\dagger} = \mathbf{A}, \tag{2.34}$$

$$\mathbf{A}\mathbf{A}^{\dagger}\mathbf{A} = \mathbf{A}^{\dagger}, \tag{2.35}$$

$$(\mathbf{A}^{\dagger}\mathbf{A})^{*} = \mathbf{A}^{\dagger}\mathbf{A}, \tag{2.36}$$

$$(\mathbf{A}\mathbf{A}^{\dagger})^{*} = \mathbf{A}\mathbf{A}^{\dagger}, \tag{2.37}$$

where $\mathbf{A}^{*}$ is the conjugate transpose of $\mathbf{A}$. If $\mathbf{A}$ is invertible then $\mathbf{A}^{\dagger} = \mathbf{A}^{-1}$. A matrix $\mathbf{X} \in \mathbb{R}^{n\times m}$ is said to be an outer inverse of $\mathbf{A}$ if Equation 2.34 is satisfied; outer inverses are not unique in general. We refer the reader to [12] for a detailed treatment of the pseudoinverse, outer inverse, and other types of generalized inverses.

Among the many properties of the pseudoinverse, a particularly useful one is the fact that $\mathbf{x}^{*} = \mathbf{A}^{\dagger}\mathbf{b}$ provides the "best (potentially) approximate solution" of the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ [68]. That is, $\mathbf{x}$ gives a least squares solution of minimum norm:

$$\mathbf{A}^{\dagger}\mathbf{b} \;\in\; \arg\min_{\mathbf{x}} \left\{ \|\mathbf{x}\| :\; \mathbf{x} \in \arg\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\| \right\}. \tag{2.38}$$

Therefore, in the pseudoinverse Newton method step (Equation 2.33), $\mathbf{x}^{k+1}$ is a minimum-norm point that minimizes the squared norm of the local linearization of $\mathbf{f}$ at $\mathbf{x}^{k}$, that is, of $h(\mathbf{x}) = \mathbf{f}(\mathbf{x}^{k}) + \mathbf{Jf}(\mathbf{x}^{k})(\mathbf{x} - \mathbf{x}^{k})$. In [57] Levin and Ben-Israel developed conditions under which this method converges Q-quadratically to a point $\bar{\mathbf{x}}$ that satisfies

$$\mathbf{Jf}(\bar{\mathbf{x}})^{\dagger}\,\mathbf{f}(\bar{\mathbf{x}}) = \mathbf{0}; \tag{2.39}$$

this is equivalent to $\bar{\mathbf{x}}$ being a stationary point of $g(\mathbf{x}) = \|\mathbf{f}(\mathbf{x})\|^2$, given that the null spaces of $\mathbf{A}^{\dagger}$ and $\mathbf{A}^{\mathrm{T}}$ coincide for every matrix $\mathbf{A}$ (note also that $\mathbf{A}$ and $\mathbf{A}^{\dagger}$ have the same rank) [12]. We can guarantee that $\bar{\mathbf{x}}$ is a solution of $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ if $\mathbf{Jf}(\bar{\mathbf{x}})$ has full row rank m. The conditions for Q-quadratic convergence involve $\mathbf{f}$ being C^1, its Jacobian matrix $\mathbf{Jf}(\mathbf{x})$ being locally Lipschitz continuous, and the pseudoinverse matrix of $\mathbf{Jf}(\mathbf{x})$ remaining bounded (in a specific sense) around $\bar{\mathbf{x}}$. We point out that the convergence theorem stated in [56] considers, first and foremost, the more general method in which we use an outer inverse of $\mathbf{Jf}(\mathbf{x}^k)$ instead of its pseudoinverse.

2.3.5. The pseudoinverse semismooth Newton method

Within the semismooth Newton method for solving a locally Lipschitz system $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, we can use a generalized inverse of the generalized derivative element instead of the latter. When we choose to use the pseudoinverse, this constitutes what we hereby refer to as the pseudoinverse semismooth Newton method, whose algorithmic map is defined as

$$\mathbf{x}^{k+1} := \mathbf{x}^k - \mathbf{G}(\mathbf{x}^k)^{\dagger}\,\mathbf{f}(\mathbf{x}^k), \qquad \mathbf{G}(\mathbf{x}^k) \in \partial\mathbf{f}(\mathbf{x}^k). \tag{2.40}$$

Convergence theorems for the semismooth Newton method using generalized inverses are limited. In [23], Dorsch et al. denote the pseudoinverse semismooth Newton method for the case when $\mathbf{G}(\mathbf{x}^k)$ is full rank as the nonsmooth projection method (NPM). The latter is employed in [23] to solve for Fritz-John points of generalized Nash equilibrium problems, which can be described by an underdetermined system of nonsmooth equations. As reported by Herrich [35], Dorsch et al. claim that the method can exhibit Q-quadratic convergence for this specific type of problem; however, the cited proof to back this claim, presented in [20], demonstrates linear convergence only.

In [21], Chen et al. consider the more general method in which we use an outer inverse $\mathbf{V}^k$ of $\mathbf{G}(\mathbf{x}^k) \in \partial^B \mathbf{f}(\mathbf{x}^k)$ in Equation 2.40 instead of $\mathbf{G}(\mathbf{x}^k)^\dagger$, where $\mathbf{f}$ is locally Lipschitz. In their Theorem 4.3, the authors show that the method converges to a solution $\bar{\mathbf{x}}$ of $\Gamma \mathbf{f}(\mathbf{x}) = \mathbf{0}$, where Γ is some $n \times m$ matrix, if we are able to guarantee that the null space of every outer inverse $\mathbf{V}^k$ that we use coincides with that of Γ, and that the $\mathbf{V}^k$ satisfy two boundedness conditions. Since the pseudoinverse is an outer inverse, with this result we can conclude that if $\mathbf{G}(\mathbf{x}^k) \in \partial^B \mathbf{f}(\mathbf{x}^k)$ is always full row rank and if $\mathbf{G}(\mathbf{x}^k)^\dagger$ satisfies the local boundedness conditions, then $\mathbf{G}(\mathbf{x}^k)^\dagger$ is always full column rank and the pseudoinverse semismooth Newton method converges to a solution of $\mathbf{f}(\mathbf{x}) = \mathbf{0}$. If $\mathbf{f}$ is semismooth, then the convergence rate is superlinear under the assumptions of Theorem 4.3 in [21]. When $\mathbf{G}(\mathbf{x}^k)$ is not guaranteed to be full row rank, the null space condition on the outer inverse is quite restrictive and not practically verifiable in general, except perhaps for certain piecewise-affine functions (see Example 2 in [21]).

To the best of our knowledge, there are no convergence rate theorems for the pseudoinverse Newton method (or its outer inverse version) using $\mathbf{G}(\mathbf{x}^k) \in \partial^B \mathbf{f}(\mathbf{x}^k)$ that are specific to PC^r functions. Due to the special properties of the latter, one could conjecture that quadratic convergence might hold under suitable conditions.

2.3.6. Fixed-point methods

Systems of nonlinear equations of the form

$$\mathbf{f}(\mathbf{x}) = \mathbf{x}, \quad \mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n, \tag{2.41}$$

are referred to as fixed-point problems, given that a solution $\mathbf{x}$ of the equation system above is called a fixed point of the function $\mathbf{f}$.

We can choose to state the fixed-point problem above in standard format $\mathbf{F}(\mathbf{x}) =$

$\mathbf{f}(\mathbf{x}) - \mathbf{x} = \mathbf{0}$, and then employ the equation solving methods presented in the previous sections. However, in general it is advantageous to employ specific methods that can explore the special structure of fixed-point problems. Moreover, most fixed-point methods do not rely on derivative information, which can be a great advantage when dealing with nonsmooth systems whose generalized derivatives require implicit function calculations.

The algorithmic map of a generic fixed-point method can be represented as

$$\mathbf{x}^{k+1} \coloneqq \mathbf{g}\left(\mathbf{x}^k, \mathbf{x}^{k-1}, ..., \mathbf{f}^k, \mathbf{f}^{k-1}, ...\right), \tag{2.42}$$

where $\mathbf{f}^k = \mathbf{f}(\mathbf{x}^k)$.

Direct substitution

The simplest fixed-point method is direct substitution, in which we update our initial guess $\mathbf{x}^0$ iteratively through the algorithmic map

$$\mathbf{x}^{k+1} \coloneqq \mathbf{f}(\mathbf{x}^k). \tag{2.43}$$

However, this method is prone to oscillations and can only be shown to achieve a linear convergence rate [65].

Wegstein's method

Perhaps the second most famous fixed-point method within process simulation is that of Wegstein. For a real-valued function of a single variable $f : \mathbb{R} \to \mathbb{R}$, its update formula is

$$x^{k+1} \coloneqq (1 - q_k)\, f(x^k) + q_k\, x^k, \tag{2.44}$$

where

$$q_k = \frac{a_k}{a_k - 1}, \quad a_k = \frac{f(x^k) - f(x^{k-1})}{x^k - x^{k-1}}. \tag{2.45}$$

In the single-variable case, Wegstein's method can be shown to achieve an almost quadratic (i.e., 1.6) convergence rate, and it may induce convergence in problems for which direct substitution diverges [34]. In the multivariable case $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$, the coefficients q_k are determined for each variable x_i independently. Therefore, in this case the method does not retain the same convergence properties as the single-variable case. According to Aspen Plus' user guide [4], Wegstein's method is considered the quickest and most reliable method for converging tear streams in process flowsheets, though admittedly it can fail when variables are strongly coupled. However, [87] demonstrated in Watson et al. that Anderson acceleration, though not traditionally used in process systems engineering, exhibits superior convergence performance than Wegstein's method for single-stage flash calculations.

Anderson acceleration

Starting from $\mathbf{x}^0$, the Anderson acceleration update step has the general format

$$\mathbf{x}^{k+1} := \alpha_0 \, \mathbf{f}(\mathbf{x}^{k-m_k}) + \alpha_1 \, \mathbf{f}(\mathbf{x}^{k-m_k+1}) + \ldots + \alpha_{m_k} \, \mathbf{f}(\mathbf{x}^k), \tag{2.46}$$

where $m_k = \min(m, k)$ and m is the memory parameter. The weights α_j must be updated at each iteration k, and the procedure for computing them determines the type of Anderson acceleration algorithm. In the original algorithm [3], which is denoted Anderson acceleration Type II in [91], the weights α_j correspond to the solution of the following constrained least squares problem:

$$\begin{aligned} \min_{\alpha_j} \; & \left\| \sum_{j=0}^{m_k} \alpha_j \, \mathbf{g}(\mathbf{x}^{k-m_k+j}) \right\|_2^2 \\ \text{s.t.} \; & \sum_{j=0}^{m_k} \alpha_k = 1, \end{aligned} \tag{2.47}$$

where $\mathbf{g}(\mathbf{x}) = \mathbf{x} - \mathbf{f}(\mathbf{x})$. That is, the α_j minimize a linear combination of the errors of the previous $m_k + 1$ iterations. In [91], the authors show this method can be considered to perform a Broyden Type II update of the inverse $\mathbf{H}^k$ of an approximate Jacobian of $\mathbf{g}$.

That is, the Anderson acceleration Type I step can be expressed as

$$\mathbf{x}^{k+1} := \mathbf{x}^k - \mathbf{H}^k \mathbf{g}(\mathbf{x}^k), \quad \mathbf{H}^k = \mathbf{I} + (\mathbf{S}^k - \mathbf{Y}^k)((\mathbf{Y}^k)^{\mathrm{T}}\mathbf{Y}^k)^{-1}(\mathbf{Y}^k)^{\mathrm{T}} \tag{2.48}$$

when $\mathbf{Y}^k$ has full column rank, where $\mathbf{S}^k = [\mathbf{s}^{k-m_k}, \ldots, \mathbf{s}^{k-1}]$, $\mathbf{Y}^k = [\mathbf{y}^{k-m_k}, \ldots, \mathbf{y}^{k-1}]$, $\mathbf{s}^j = \mathbf{x}^{j+1} - \mathbf{x}^j$, and $\mathbf{y}^j = \mathbf{g}^{j+1} - \mathbf{g}^j$.

On the other hand, the so-called Anderson acceleration Type I algorithm [29] is based on a Broyden Type I update of an approximate Jacobian of $\mathbf{g}$. The method's step is given by

$$\mathbf{x}^{k+1} := \mathbf{x}^k - (\mathbf{B}^k)^{-1}\mathbf{g}(\mathbf{x}^k), \quad \mathbf{B}^k = \mathbf{I} + (\mathbf{Y}^k - \mathbf{S}^k)((\mathbf{S}^k)^{\mathrm{T}}\mathbf{S}^k)^{-1}(\mathbf{S}^k)^{\mathrm{T}} \tag{2.49}$$

when $\mathbf{S}^k$ has full column rank.

In [91], Zhang et al. develop a stabilized version of the Anderson acceleration Type I method by including Powell regularization, restart checking, and safeguarding steps. The algorithm is shown to be globally convergent without any assumptions on $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ other than it being non-expansive, i.e., $\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})\|_2 \leq \|\mathbf{x} - \mathbf{y}\|_2$ for all $\mathbf{x}, \mathbf{y}$, and the solution set of $\mathbf{f}(\mathbf{x}) = \mathbf{x}$ being non-empty. Therefore, the method is also applicable to nonsmooth functions.

In Chapter 5 we make use of the MATLAB implementation of the Anderson acceleration algorithm of Zhang et al. [91], which was made available by the authors at https://github.com/cvxgrp/nonexp_global_aa1 under the option "aa1-safe". However, we have had to modify their implementation within our simulations in Chapter 5. Specifically, we had to rewrite a right matrix division operation using the pseudoinverse to avoid the algorithm running into errors under singular matrices. We chose to use the same set of hyperparameters employed in [91], i.e., $\bar{\theta} = 0.01$, $\tau = 0.001$, $D = 10^6$, $\epsilon = 10^{-6}$, and memory $m = 5$.

Chapter 3

A nonsmooth modeling strategy to simulate dry and vaporless distillation columns

Many process systems, such as distillation columns and other equipment with phase change, exhibit multiple modes of physical behavior that can be described by non-differentiable (i.e., nonsmooth) models. In this chapter, we introduce a nonsmooth model for steady-state multistage distillation that can describe columns with dry and/or vaporless stages reliably. The model consists of a system of nonsmooth MESH and specification equations, without inequality or complementarity constraints, that can be directly solved with the semismooth Newton method using automatically computed generalized derivatives. With a modified version of pseudo-arclength continuation, we have been able to observe several novel types of bifurcations in dry and/or vaporless distillation column models. Many of these bifurcations exhibit degenerate behavior with an infinite number of steady states for certain critical input specifications, and occur in general multistage distillation systems regardless of the mixture components or thermodynamic models chosen. We present case studies drawn from the literature and analyze the occurrence and behavior of the bifurcations with respect to several types of column configurations, involving ideal stages, stage efficiencies, pressure gradients, tray heat transfer, multiple feeds, and side products. The associated bifurcation curves are inherently nonsmooth and can be described mathematically by the concept of piecewise-differentiable manifolds introduced in Chapter 7.

3.1. Introduction

Trayed columns are still prevalent in unit operations for two-phase contact, such as absorption, stripping and distillation [79], whereas packed column simulation also commonly employs an equivalent number of stages. Though the rate-based modeling approach [51] describes the complex transport phenomena in multistage columns much more realistically than the efficiency/equilibrium stage approach, the former relies on empirical correlations for hydrodynamic, heat and mass transfer parameters, which depend on tray geometry and column configuration [80]. Therefore, equilibrium stage models are still invaluable for the preliminary stages of process design, when detailed column specifications are not established yet. This simpler modeling approach is also widely used in industrial practice because it requires less computational effort, and condenses all deviations from ideal mass transfer behavior into a single parameter, the stage efficiency.

The efficiency/equilibrium stage approach to steady-state simulation of multistage separations employs the MESH (Mass balance, Equilibrium, Summation and energy balance, where H stands for enthalpy) equations, which assume vapor-liquid equilibrium exists at the conditions of each stage. In the rate-based approach, vapor-liquid equilibrium is also enforced at the phase interface between the bulk vapor and liquid phases. However, certain process specifications can lead to a steady state in which the exiting liquid or vapor phase is absent from one or more stages. In a dry/vaporless stage, the remaining vapor/liquid outlet stream can be superheated/subcooled; under these conditions, vapor-liquid equilibrium no longer exists and consequently both the MESH-equation and rate-based models are no longer valid. This gives rise to the often-experienced “dry column” simulation errors in commercial process software, such as Aspen Plus’ [2] RadFrac multistage column model. For these “problematic” process specifications, it is widely known that RadFrac’s equilibrium-stage model aborts all calculations and exhibits a severe error message stating that stages “dried up” of liquid and/or vapor. In addition, we have found that RadFrac’s rate-based model (previously called RateFrac) also fails to converge and prompts a general error message, without detailing its cause.

One might argue that the absence of a valid model to simulate distillation columns with dry/vaporless stages is irrelevant, since such steady-state solutions correspond to extreme and undesirable operating conditions. However, given a certain set of process specifications, we cannot predict a priori which phases will be present within each stage in the column. When current distillation software is unable to find a vapor-liquid equilibrium solution, the user is left with the complicated task of changing specifications by trial-and-error until the model can converge, which is especially challenging within a flowsheet with several interconnected equipment and recycles. Additionally, process specifications are iteratively changed outside user control in sequential-modular simulation of flowsheets with recycle streams, design specifications, and in process optimization; therefore, the solution algorithms might stray into dry/vaporless conditions and fail to converge.

Without a suitable model, we cannot answer a very fundamental question: what is the steady-state behavior of columns with dry/vaporless stages? In order to obtain these steady states, we must change the model equations that describe each stage to reflect which phases (vapor-liquid, vapor only, or liquid only) are present at the solution; however, we have no knowledge of the latter prior to simulation. It is possible to create a single model that automatically “switches” between describing equations and selects the correct ones, at the cost of introducing nonsmooth (i.e., non-differentiable) behavior and requiring more advanced mathematical tools not present in commercial software.

Previous work on steady-state simulation of dry/vaporless distillation columns, using MESH-based models, dates back to the 1990s and is limited to two other papers [16, 31]. In the first paper [16], the equilibrium relationship for each stage is relaxed in dry/vaporless regimes by introducing inequalities in terms of a slack variable. In order to address the inequalities, the original task of simulating the process is transformed into the optimization problem of minimizing the slack variable. In subsequent work [31], the KKT conditions for a similar optimization formulation are used to create a model with complementarity constraints. The latter are rewritten as nonsmooth equations in terms of the max operator, and the model must be solved iteratively as a series of smooth-approximation problems. However, in both papers, only limited simulation results with dry/vaporless columns are reported, corresponding to very few sets of column specifications. As demonstrated in this chapter, these do not give the full picture of *how* the vapor/liquid "drying" process occurs within the column.

To this date, all other subsequent papers that address modeling of dry/vaporless distillation regimes [53, 70, 71, 25, 26, 93] have considered flowsheet optimization only – the type of problem where complementarities are ostensibly easier to handle mathematically. However, all the aforementioned approaches rely either on a series of equation-solving problems or on optimization algorithms even when only a single simulation is needed. This increases computational effort, and introduces nonphysical variables and parameters that need to be heuristically tuned for each process flowsheet.

On the other hand, recent advances in the automatic evaluation of generalized derivatives [46] have opened up the possibility of creating explicitly-nonsmooth algebraic models that can be directly solved with Newton-like methods. By introducing a single nonsmooth equation in terms of the mid function, which returns the median of its three arguments, Watson et al. [85, 87] have successfully reformulated the phase equilibrium problem for a single stage, in order to perform flash calculations and model multistream heat exchangers with phase change. With an analogous approach, Sahlodin et al. [76] proposed a nonsmooth dynamic model for multistage distillation columns, formulated in terms of liquid and vapor molar holdups.

In this work, we extend the explicitly-nonsmooth modeling strategy to steady-state distillation simulation by proposing a nonsmooth MESH model, which remains valid regardless of the phases present in each stage. Using this compact equation-based modeling strategy, and by developing a nonsmooth version of the pseudo-arclength continuation method [44], we have been able to observe infinitely many steady states with dry/vaporless stages in distillation column models. This degenerate behavior occurs for certain critical input specifications independently of the particular mixture being separated or the thermodynamic models used, and persists even when different column configurations are specified.

Bifurcations, or changes in the number of steady-state solutions, have been previously observed in multistage distillation column simulation with smooth models [59, 10, 55, 42, 18, 49] and also confirmed experimentally [47, 61]. Most cases analyzed involve homogeneous azeotropic distillation systems with at least three components [59, 10, 55], although bifurcations have also been observed in binary distillation [42] and Petlyuk columns [18]. In the majority of cases, the curve of steady-state solutions contains 2 turning points forming a hysteresis curve, and therefore a total number of 3 steady states exist for parameter values in between the turning points. An extended hysteresis curve, with 4 turning points yielding up to 5 steady states, has also been reported within azeotropic distillation [49]. In addition, hysteresis behavior is often responsible for the more familiar occurrence of multiple steady states in exothermic chemical reactors. On the other hand, a Hopf bifurcation, with the corresponding appearance of a limit cycle, has also been observed in association with a hysteresis curve for a ternary azeotropic column [55]. However, the occurrence of multiple steady states in distillation simulation might depend, in some instances, on the thermodynamic model used [59, 11].

Interestingly, Bekiaris et al. [10] presented the theoretical possibility of infinitely many steady states for homogeneous azeotropic distillation with a simplified analysis, which considered infinite reflux, infinitely many trays and constant molar overflow. However, to the best of our knowledge, this degeneracy of steady states has not been observed in distillation or other process systems described by more realistic models. Moreover, a distinctive feature of the degenerate bifurcations introduced in this chapter is that they involve nonsmooth behavior.

In the following sections of this chapter, we first discuss the conceptual challenges in describing dry and vaporless equilibrium stages, and present existing nonsmooth modeling strategies and simulation methods. We then describe our nonsmooth MESH model and the numerical continuation strategy developed to trace the curves of infinitely many steady states, in view of the mathematical concept of piecewise-differentiable extrinsic manifolds introduced in Chapter 7. Next, we conduct detailed parameter continuations in two case studies from the literature [16, 31] and vary several types of column specifications, in order to describe and analyze the degenerate and non-degenerate bifurcations that occur in dry/vaporless distillation columns. Finally, we present a summary of the novel, nonsmooth bifurcations and conclude this chapter with remarks on our contributions and future lines of work.

3.2. The issues with dry and vaporless stages

Consider an equilibrium stage at steady state depicted in Figure 3.1, which could represent either a flash vessel or, in a simplified analysis, one of the stages within a column. Let F, L and V be the total inlet, outlet liquid, and outlet vapor molar flow rates, respectively, and $\mathbf{z}$, $\mathbf{x}$ and $\mathbf{y}$ the vectors with mole fractions of the N_c components for each respective stream. The system of MESH equations that models the stage, which assumes that outlet liquid and vapor are in equilibrium, is:

$$F = L + V, \tag{3.1}$$

$$Fz_i = Lx_i + Vy_i, \quad i = 1, \ldots, N_c, \tag{3.2}$$

$$Fh_F + Q = Lh_L + Vh_V, \tag{3.3}$$

$$y_i = K_i\ x_i, \quad i = 1, \ldots, N_c, \tag{3.4}$$

$$\textstyle\sum_{i=1}^{N_c} y_i - \sum_{i=1}^{N_c} x_i = 0, \tag{3.5}$$

where i is the index for a specific component, h_j is the molar enthalpy of stream j, T and P are the stage temperature and pressure, Q is the heat transfer rate to the stage, and $K_i \equiv K_i(T, P, \mathbf{x}, \mathbf{y})$ is the equilibrium ratio for component i. Note that the single summation equation in Equation 3.5 indirectly enforces both the liquid and vapor phase mole fractions to sum to one, since the mole balances for all the components are included together with the overall mole balance.

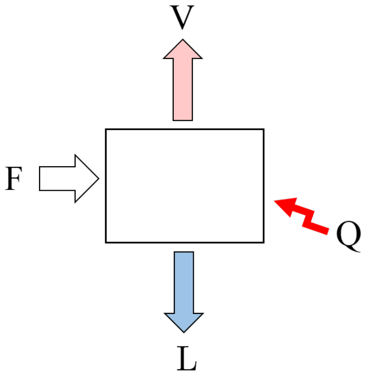


Figure 3.1: A single-stage flash vessel.

3.2.1. Dry and vaporless phase regimes

We define a stage at steady state to be dry if its total outlet liquid flow rate is equal to zero ($L = 0$). Analogously, a stage without a vapor outlet stream is said to be vaporless ($V = 0$). This way, we can characterize the following possible phase regimes for each stage:

- **Phase Regime I**: a stage with vapor and liquid outlets in equilibrium with each other;
- **Phase Regime II**: (a) a dry stage with a dew-point vapor outlet, (b) a vaporless stage with a bubble-point liquid outlet;
- **Phase Regime III**: (a) a dry stage with a superheated vapor outlet, (b) a vaporless stage with a subcooled liquid outlet.

Each set of feasible input parameters, in the correct number to fix the necessary degrees of freedom, may yield a steady state in a certain phase regime. For instance, consider a PT-flash vessel for which all feed conditions are specified. Figure 3.2 presents a schematic view of the input parameter space in terms of the specified temperature T and pressure P, with the resulting phase regimes at steady state. Note that Phase Regimes II correspond to the nonlinear boundaries between the regions for Phase Regimes I and III.

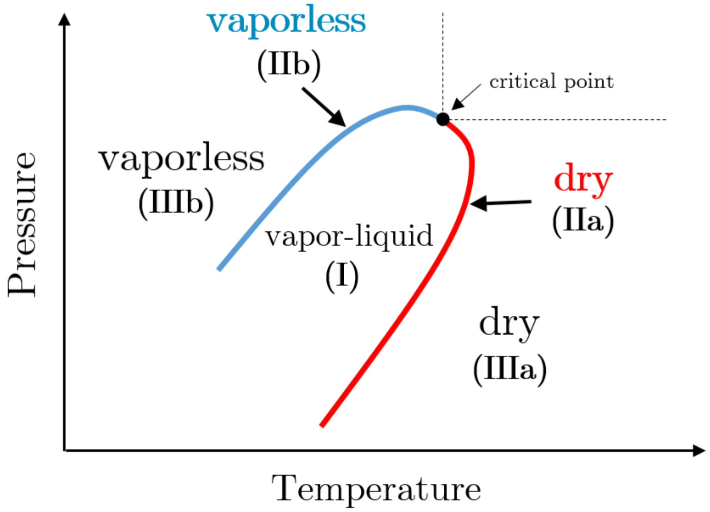


Figure 3.2: Phase regimes at the solution for each temperature-pressure pair in a PT-flash.

3.2.2. The MESH equations are not valid in Phase Regimes III

A robust model must encompass all possible modes of behavior of the system, and yield the correct steady state for any set of feasible input specifications. This means that variables characterizing the state of every possible stream in a process, such as compositions and temperatures, must always be included and solved for within the model variables. The mole fractions of an absent liquid or vapor stream are examples of fictitious variables; they bear no physical meaning but can still be computed using the model equations, Equations 3.1-3.5, as long as the correlations used to evaluate K_i and the phase enthalpies remain well-defined.

However, suppose that a given set of input specifications leads to a steady state in which the stage operates in Phase Regimes IIIa or IIIb. It can be shown from the KKT conditions for minimization of the Gibbs free energy [76] that, in such a steady state, fictitious mole fractions computed with Equation 3.4 sum to *less than* one. For instance, for a vaporless steady state with subcooled liquid, $\sum_{i=1}^{N_c} y_i = \sum_{i=1}^{N_c} K_i x_i < 1$. Since the MESH Equations 3.1-3.5 always enforce the mole fractions of both phases to sum to one, they cannot yield the correct steady-state solution. Instead, we obtain a unique but non-physical MESH solution in which the flow rate of the absent phase is negative; in the previous example, $V < 0$. Therefore, the MESH equations are a valid model to describe Phase Regimes I and II, but not Phase Regimes III.

3.2.3. Valid equations for the dry and vaporless phase regimes

We can propose modified systems of equations to model dry or vaporless stages, both in Phase Regimes II and III, depending on how we formulate and compute the fictitious mole fractions. However, these models are not valid in Phase Regime I (vapor-liquid equilibrium). Any fictitious mole fraction formulation yields the same solutions in terms of physical variables, but the convergence properties of the equation system can be affected by the formulation chosen.

3.2.3.1. Formulation 1:

In this strategy, fictitious mole fractions are computed from the unchanged equilibrium relationship and are not required to sum to one. The system of model equations for a dry stage (Phase Regimes IIa and IIIa) consists of Equations 3.1-3.4 and

$$L = 0, \tag{3.6}$$

which intuitively replaces the summation equation, Equation 3.5. Analogously, the model equations for a vaporless stage (Phase Regimes IIb and IIIb) consist of Equations 3.1-3.4 and

$$V = 0. \tag{3.7}$$

3.2.3.2. Formulation 2:

In this approach, Equations 3.1-3.3 and the summation Equation 3.5 are maintained, while the equilibrium relationship in Equation 3.4 is relaxed for all components by introducing a non-physical variable β for each stage:

$$y_i = \beta K_i \; x_i, \quad i = 1, \ldots, N_c. \tag{3.8}$$

This allows the fictitious mole fractions to sum to one but requires an additional model equation to be included: Equation 3.6 for a dry stage, Equation 3.7 for a vaporless stage, and $\beta = 1$ for a stage with vapor-liquid equilibrium. Note that, from the previously mentioned result for minimization of the Gibbs free energy, we must have $\beta \leq 1$ for a dry stage, and $\beta \geq 1$ for a vaporless stage.

3.2.4. The phase regime cannot be predicted prior to simulation

Since predicting the exact distribution of regimes within parameter space is a complex task, the mode of behavior corresponding to a given set of input parameters is usually not known before simulating the system. On the other hand, we must choose an equation system and its associated mode of behavior to simulate the process. A naive way to approach this conundrum is by trial-and-error, attempting each system of model equations until one of them converges to a valid solution. While this seems feasible in the case of a single-stage flash, for which only 3 such models exist, it is not practical for a multistage column. In the latter case, *each stage* has 3 possible sets of describing equations. The overall number of possible model equations for the column is equal to 3^N, scaling exponentially with the number of stages N. Process simulation software such as Aspen Plus and HYSYS consider only the MESH distillation model, in which all stages are assumed to be in Phase Regime I (vapor-liquid); no dry or vaporless models are included.

Instead, it is possible to create a single model that remains valid in all possible phase regimes, automatically switching between the equations for each stage and enforcing the correct ones without prior knowledge of the regime at the solution. However, this can only be achieved by introducing nonsmooth or non-differentiable behavior (e.g., the complementarity constraint and explicitly nonsmooth strategies described below) or even discrete variables (e.g., generalized disjunctive programming [33]).

3.3. Nonsmooth modeling approaches

3.3.1. Complementarity constraints

Modeling of equilibrium stages with complementarity constraints is due to Biegler and collaborators [31]. In their strategy, Formulation 2 is chosen to define fictitious mole fractions. In order to encompass both the dry and vaporless equation systems, other two non-physical slack variables, s_V and s_L, must be added for each stage, aside from β. The overall model consists of Equations 3.1-3.3, 3.5, 3.8, and the additional relationships

$$\beta = 1 - s_L + s_V, \tag{3.9}$$

$$0 \leq L \perp s_L \geq 0, \tag{3.10}$$

$$0 \leq V \perp s_V \geq 0. \tag{3.11}$$

A complementarity constraint $0 \leq a \perp b \geq 0$ forces at least one of the variables a, b to be zero and both to be non-negative; it can be expressed by the smooth equation $ab = 0$ together with the inequalities $a, b \geq 0$. Equivalently, a complementarity constraint can be reformulated as a single nonsmooth equation that is non-differentiable (at least) at the origin, such as $\min(a, b) = 0$, $a = \max(0, a - b)$, or the Fischer-Burmeister equation $\sqrt{a^2 + b^2} - (a + b) = 0$.

In order to avoid handling inequality constraints within equation solving, Gopal and Biegler [31] implement distillation simulation with dry and vaporless stages by solving a series of smoothing approximations to the max reformulation $a = \max(0, a - b)$. In subsequent work, Biegler and collaborators [53, 70, 25] incorporate the complementarity constraints into nonlinear programs for distillation optimization. The current strategy [26, 93] is to include these constraints in the form of exact penalty terms $\rho \mathbf{a}^{\mathrm{T}} \mathbf{b}$ in the objective function, with the parameter ρ needing to be tuned for each problem at hand. In both simulation and optimization settings, the complementarity constraint approach introduces artificial variables and parameters that need to be tuned, initialized and updated, and does not allow for simulation with direct equation solving, creating the need to solve a series of problems in addition to the original one. Moreover, the infinitely many steady states described in this chapter have never been obtained or presented within this modeling strategy, perhaps due to difficulties in performing the necessary continuation methods when complementarity constraints are present.

3.3.2. Explicitly nonsmooth equations

Non-differentiable functions, such as the absolute value, min and max, can be explicitly used to create a single system of nonsmooth algebraic equations, without inequality constraints, that is a valid model for all system behaviors. This concise approach neither introduces non-physical variables nor increases problem size. As detailed below, recent developments enable us to compute generalized derivatives for these models and to use direct nonsmooth equation-solving methods for process simulation.

In the explicitly nonsmooth model proposed by Watson and Barton [85], Formulation 1 is used to define the fictitious mole fractions. Equation 3.5 is replaced with

$$\operatorname{mid}\left(\frac{V}{F}\ ,\ \sum_{i=1}^{N_c} x_i - \sum_{i=1}^{N_c} y_i\ ,\ \frac{V}{F} - 1\right) = 0, \tag{3.12}$$

where the piecewise-smooth function mid returns the median of its three arguments. Equivalently, the third argument can be substituted by $-\frac{L}{F}$. The denominator F in the first and third arguments acts simply as a scaling factor so that all three arguments have a similar order of magnitude, and can therefore be substituted by any other positive constant. With the mid function, we can include 3 different model equations, respectively Equations 3.7, 3.5 and 3.6, in a single one. The correct expression is automatically satisfied (i.e., becomes the median) according to the phase regime: vaporless $\left(\frac{V}{F} = 0\right)$, vapor-liquid $\left(\sum_{i=1}^{N_c} x_i - \sum_{i=1}^{N_c} y_i = 0\right)$, and dry $\left(\frac{V}{F} - 1 = -\frac{L}{F} = 0\right)$. Equation 3.12 is potentially nondifferentiable at points where two of the arguments are equal. For instance, in Phase Regimes II, one of the flow rates is zero and the summation relationship in Equation 3.5 is still satisfied. The explicitly nonsmooth approach has been successfully applied to perform flowsheet flash calculations [87], and model multi-stream heat exchangers with [85] and without [86] phase change.

The explicitly nonsmooth strategy can also accommodate Formulation 2 with a clear advantage over the complementarity constraint approach, since no slack variables or inequality constraints are introduced. In this case, the system of piecewise-smooth equations consists of Equations 3.1-3.3, 3.5, 3.8 and the additional relationship

$$\operatorname{mid}\left(\frac{V}{F}\ ,\ \beta - 1\ ,\ \frac{V}{F} - 1\right) = 0. \tag{3.13}$$

Alternatively, this extra equation associated with the extra variable β in Formulation 2 can be further eliminated by making use of the identity $\beta \equiv \frac{\sum_{i=1}^{N_c} y_i}{\sum_{i=1}^{N_c} K_i x_i}$. One way to do so, as presented by Watson et al. [88], is to maintain Equations 3.1-3.3, 3.5 and replace the N_c equilibrium relationships in Equation 3.4 with

$$y_i \textstyle\sum_{i=1}^{N_c} K_i x_i = K_i \cdot x_i \sum_{i=1}^{N_c} y_i, \quad i = 2, \ldots, N_c, \tag{3.14}$$

$$\operatorname{mid}\left(\frac{V}{F}\ , \textstyle\sum_{i=1}^{N_c} y_i - \sum_{i=1}^{N_c} K_i x_i\ ;\ \displaystyle\frac{V}{F} - 1\right) = 0. \tag{3.15}$$

Watson et al. [88] recommend choosing the most volatile component to be left out from Equations 3.14 in order to improve numerical conditioning, although any choice of component $i = 1$ is valid.

3.4. The proposed nonsmooth MESH model

Consider a steady state-distillation column with N stages, numbered from top to bottom, separating a mixture with N_c components. For each generic stage j, as depicted in Figure 3.3, we propose a modified system of nonsmooth MESH equations:

$$L_{j-1} + V_{j+1} + F_j - (L_j + W_{L,j}) - (V_j + W_{V,j}) = 0, \tag{3.16}$$

$$x_{i,j-1}L_{j-1} + y_{i,j+1}V_{j+1} + z_{i,j}F_j - x_{i,j}(L_j + W_{L,j}) - y_{i,j}(V_j + W_{V,j}) = 0, \quad i = 1, \ldots, N_c, \tag{3.17}$$

$$h_{j-1}^L L_{j-1} + h_{j+1}^V V_{j+1} + h_j^F F_j - h_j^L(L_j + W_{L,j}) - h_j^V(V_j + W_{V,j}) + Q_j = 0, \tag{3.18}$$

$$y_{i,j} - K_{i,j} x_{i,j} = 0, \quad i = 1, \ldots, N_c, \tag{3.19}$$

$$\text{mid}\left(\frac{V_j + W_{V,j}}{F_s} \ , \ \sum_{i=1}^{N_c} x_{i,j} - \sum_{i=1}^{N_c} y_{i,j} \ , \ \frac{-(L_j + W_{L,j})}{F_s} \right) = 0, \tag{3.20}$$

where V_j and L_j are the liquid and vapor molar flow rates leaving stage j, with the respective mole fractions $y_{i,j}$, $x_{i,j}$ of component i; F_j and $z_{i,j}$ are the molar flow rate and mole fractions of the feed stream to stage j; $W_{V,j}$ and $W_{L,j}$ are the flow rates of vapor and liquid side products withdrawn from the stage; h_j^V and h_j^L are the molar enthalpies of the outlet vapor and liquid phases, $K_{i,j}$ is the equilibrium ratio for component i, Q_j is the heat transfer rate to the stage, and F_s is the sum of the feed flow rates to all stages. As illustrated in Figure 3.3, some of the streams are absent in the first and last stages. As opposed to Chapters 4 and 5, in which we potentially reset or relax one or more column specifications in case they happen to be infeasible, in this chapter we (attempt to) enforce all user-chosen column specifications strictly.

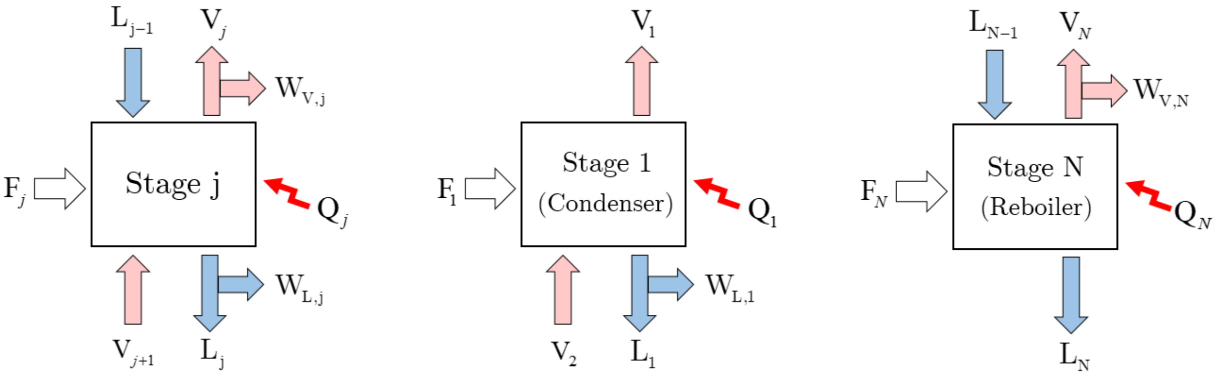


Figure 3.3: An intermediate stage j (left), the condenser (center), and the reboiler (right) in a distillation column.

3.4.1. The mid equation

Two modifications are introduced into the first and third arguments of the original mid equation (Equation 3.12) for a single-stage flash. Firstly, the numerators of these two arguments, which represent the overall vapor and liquid outlets of the stage, now include the side product stream flow rates $W_{V,j}$ and $W_{L,j}$, respectively. Secondly, instead of the overall inlet flow rate $L_{j-1} + V_{j+1} + F_j$ particular to each stage j, F_s is used in the denominators as a constant scaling factor for all stages.

Here, we note that the mid equation not only relaxes the summation equation in Phase Regimes III, but also automatically bounds the *total* outlet flow rates $(V_j + W_{V,j})$ and $(L_j + W_{L,j})$ to be non-negative. To see why that is, consider a vaporless stage: in case $(V_j + W_{V,j})$ assumes a negative value, both the first and third arguments of the mid function are strictly negative and therefore the median cannot equal zero to satisfy the equation. However, unlike the single-stage flash case, the individual flow rates L_j or V_j are not guaranteed to be non-negative at the solution if a liquid or vapor side product is present, respectively.

The above equations employ Formulation 1 to define the fictitious mole fractions, but the other two forms of Formulation 2 can equivalently be used by making the necessary modifications previously described. When using Equation 3.13, one extra variable β_j must be included for each stage.

3.4.2. The condenser

The total distillate flow rate D is given by

$$D = W_{L,1} + V_1, \tag{3.21}$$

and the reflux ratio is defined as $R = L_1/D$. The vapor distillate fraction $\theta = V_1/(W_{L,1} + V_1)$ for the condenser, ranging from 0 to 1, must be specified with an additional equation.

For a partial condenser $(0 < \theta \leq 1)$, the mid equation (Equation 3.20) is maintained. For a total condenser $(\theta = 0)$, since $V_1 = 0$ is constant, the mid equation is replaced with

$$\sum_{i=1}^{N_c} x_{i,1} - y_{i,1} = 0 \tag{3.22}$$

to ensure a bubble-point outlet liquid stream.

3.4.3. Stage efficiencies

Equations 3.16-3.20 define an ideal stage in which vapor-liquid mass transfer happens to its full extent, with outlet vapor and liquid mole fractions related through the equilibrium relationship in Equation 3.19. Instead, less-than-ideal mass transfer in a real stage j can be approximately described by introducing stage efficiencies $\eta_{i,j}$ for each component i. If $\eta_{i,j}^{M}$ represents the Murphree vapor phase efficiency, Equation 3.19 is replaced with

$$(y_{i,j} - y_{i,j+1}) - \eta_{i,j}^{M}(K_{i,j}\ x_{i,j} - y_{i,j+1}) = 0. \tag{3.23}$$

If the vaporization efficiency $\eta_{i,j}^{V}$ is specified instead, Equation 3.19 is replaced with

$$y_{i,j} - \eta_{i,j}^{V} K_{i,j} x_{i,j} = 0. \tag{3.24}$$

3.4.4. Side products

Aside from the main top and bottom products with flow rates D and L_N, respectively, vapor and/or liquid side products can also be withdrawn from intermediate stages ($2 \leq j \leq N-1$). Most formulations of the MESH equations in the literature are defined in terms of withdrawal ratios, such as $W_{L,j}/L_j$ or $1 + W_{L,j}/L_j$. However, these ratios become undefined for dry or vaporless stages, and therefore it is essential to choose the withdrawal flow rates $W_{V,j}$, $W_{L,j}$ as the variables in our model.

For stages without side products, $W_{L,j}$ and $W_{V,j}$ are set to zero. When a vapor or liquid side product is present at an intermediate stage, a corresponding specification equation must be included, usually in terms of either the withdrawal ratio or the side product flow rate.

3.4.5. Withdrawal ratio specification

In a non-dry stage, a desired value for the withdrawal ratio $R_{L_j} = W_{L,j}/L_j$ can be enforced by adding a specification equation in the form

$$W_{L,j} - R_{L_j} L_j = 0. \tag{3.25}$$

This way, $W_{L,j}$ is enforced to zero for a dry stage despite the withdrawal ratio itself becoming undefined, which reflects the physical behavior of a splitter valve. An analogous equation is included for a desired vapor withdrawal ratio $R_{V_j} = W_{V,j}/V_j$.

3.4.6. Flow rate specification

In order to enforce a desired value $W_{L,j,\text{spec}}$ for the liquid side product flow rate in stage j, the following nonsmooth specification is included:

$$\min\left(L_j,\ -|W_{L,j} - W_{L,j,\text{spec}}|\ \right) = 0. \tag{3.26}$$

This equation enforces the specified value for $W_{L,j}$ and simultaneously bounds L_j to be ≥ 0. An analogous equation is included for a desired vapor side product flow rate $W_{V,j,\text{spec}}$, bounding V_j to be ≥ 0.

3.4.7. Specifying the degrees of freedom

When we specify the number N of stages, all feed stream conditions, all stage pressures, the heat duties for intermediate stages (commonly set to 0), and all side product ratios or flow rates for intermediate stages, two degrees of freedom remain for a distillation column. In the standard MESH model, these are fixed directly by two specification equations. For instance, desired values R_spec and D_spec for the reflux ratio and distillate flow rate are specified, respectively, by the equations

$$R - R_{\text{spec}} = 0, \tag{3.27}$$

$$D - D_{\text{spec}} = 0. \tag{3.28}$$

However, if $0 \leq \theta < 1$, a nonzero liquid distillate flow rate $W_{L,1}$ is present and the condenser equations presented so far cannot guarantee a non-negative reflux flow rate L_1. To correct that, we must modify one of these two specification equations and create a formulation analogous to Equation 3.26. For instance, the distillate flow rate specification becomes

$$\min\left(L_1,\ -|D - D_{\text{spec}}|\ \right) = 0. \tag{3.29}$$

3.4.8. Model simulation and parameter continuation

Our nonsmooth MESH model is valid for all possible combinations of liquid-only, vapor-only and liquid-vapor phase regimes in each stage. Moreover, the mathematical behavior of the model reflects the physical behavior of the system: all flow rates are automatically enforced to be greater than or equal to zero, and therefore any solution obtained with our model is physically valid in that regard. Another distinctive feature is that infeasible input parameter values are also mathematically infeasible, and in this case the model has no solution.

The total set of n model equations is represented by the nonsmooth nonlinear system

$$\mathbf{f}(\mathbf{x}, \lambda) = \mathbf{0}, \tag{3.30}$$

where $\mathbf{x} \in \mathbb{R}^n$ represents the n model variables that are solved for, $\lambda \in \mathbb{R}$ represents a single input parameter while all other degrees of freedom remain fixed, and $\mathbf{f} : \mathbb{R}^{n+1} \to \mathbb{R}^n$ is piecewise-smooth (PC^∞).

In this work, we wish to analyze how the steady-state solutions $\mathbf{x}$ change as we vary the parameter λ. Specifically, we say that a bifurcation occurs at a parameter value λ^* when there is a change in the number of solutions $\mathbf{x}$ for each λ. Many concepts from bifurcation theory for dynamical systems can be applied to analyze this problem; the only caveat is that no dynamic or stability considerations can be made if $\mathbf{x}'(t) \neq \mathbf{f}(\mathbf{x}, \lambda)$, which is the case for the nonsmooth steady-state MESH model. In a differential-algebraic dynamic model for a distillation column, the differential equations express the time derivatives of the molar and enthalpy holdups of each stage, which are not present as variables in steady-state MESH models.

Bifurcations can be identified with continuation methods, which are responsible for the numerical approximation of the solution set

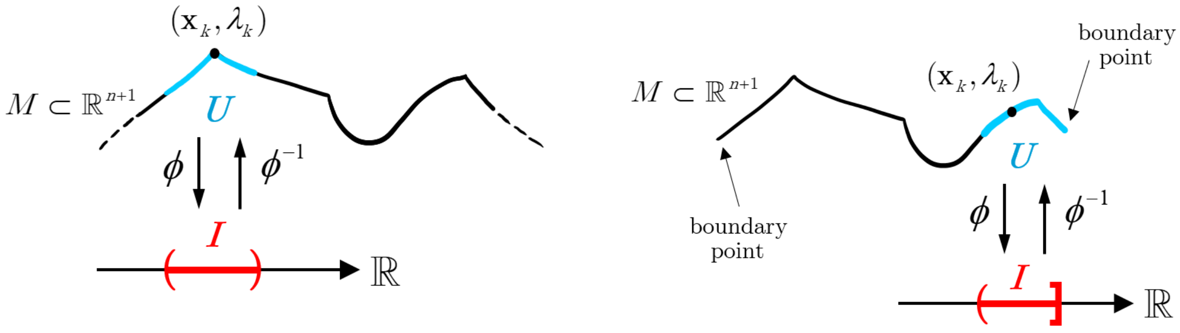
$$M = \{(\mathbf{x}, \lambda) \in \mathbb{R}^{n+1} : \mathbf{f}(\mathbf{x}, \lambda) = \mathbf{0}\}. \tag{3.31}$$

If the limiting partial Jacobians of $\mathbf{f}$ with respect to $\mathbf{x}$ (represented by $\mathbf{J}_{\mathbf{x}}\mathbf{f}_{(i)}(\mathbf{x}, \lambda)$) remain invertible, we can perform a simple parameter continuation, in which we fix λ and solve for $\mathbf{x}$ using the semismooth Newton method (see Section 2.3.2). The limiting partial Jacobians, used to compute the Newton step according to Equation 2.28, are obtained exactly with the automatic differentiation algorithm of Khan and Barton [46].

However, if the limiting partial Jacobians with respect to $\mathbf{x}$ become singular, a bifurcation is likely to be present and Newton-type methods fail in solving for $\mathbf{x}$ directly. In such cases, as introduced in this chapter, we can employ a nonsmooth version of pseudo-arclength continuation to trace solution points $(\mathbf{x}, \lambda)$, as long as the solution set remains a 1-dimensional PC^r manifold.

3.5. PC^r manifolds

Definition 3.5.1 below, which is a specific instance of Definition 7.5.1, gives the precise notion of a piecewise-differentiable (PC^r) manifold as introduced in Chapter 7.


$(\mathbf{x}_k, \lambda_k)$
$M \subset \mathbb{R}^{n+1}$
U
ϕ
ϕ^{-1}
I
$\mathbb{R}$
boundary point
$M \subset \mathbb{R}^{n+1}$
$(\mathbf{x}_k, \lambda_k)$
boundary point
U
ϕ
ϕ^{-1}
I
$\mathbb{R}$

(a) (b)

Figure 3.4: (a) A 1-dimensional PC^r manifold; (b) a 1-dimensional PC^r manifold with 2 boundary points.

Definition 3.5.1 (Extrinsic PC^r manifold). An extrinsic PC^r manifold (with boundary) is a set $M \subset \mathbb{R}^n$ endowed with the subspace topology such that for every point $\mathbf{x} \in M$ there exists a neighborhood $U \subset M$ of $\mathbf{x}$, an open subset $V \subset \mathbb{R}^k$ $[V \subset \mathbb{H}^k]$ for some $k \leq n$ which is called the dimension of M at $\mathbf{x}$, and an extended PC^r homeomorphism $\boldsymbol{\phi} : U \to V$ (see Definition 7.2.1 in Chapter 7).

Now, suppose the solution set $M \subset \mathbb{R}^{n+1}$ to $\mathbf{f}(\mathbf{x}, \lambda) = \mathbf{0}$ is a 1-dimensional PC^r manifold (Figure 3.4a). This means that, on an open neighborhood $U \subset M$ of every solution point $(\mathbf{x}_k, \lambda_k) \in M$, points in the solution set can be expressed as a function of a single parameter $v \in \mathbb{R}$,

$$(\mathbf{x}, \lambda) = \phi^{-1}(v), \tag{3.32}$$

where $\phi^{-1} : I \to U$ is a PC^r homeomorphism and $I \subset \mathbb{R}$ is an open interval. The PC^r functions ϕ^{-1} and ϕ are called a local parametrization and a local coordinate map, respectively. On the other hand, if M is a 1-dimensional PC^r manifold with boundary (Figure 3.4b), then I might also be a half-closed interval, and any point $(\mathbf{x}, \lambda)$ corresponding to the closed endpoint of such an interval I is called a boundary point.

3.6. Pseudo-arclength continuation methods

3.6.1. Smooth systems

The pseudo-arclength continuation method, developed by Keller [44] for smooth functions, can be used to trace the solution set $M \subset \mathbb{R}^{n+1}$ to $\mathbf{f}(\mathbf{x}, \lambda) = \mathbf{0}$ when $\mathbf{f} : \mathbb{R}^{n+1} \to \mathbb{R}^n$ is a C^2 function that satisfies the regularity condition, i.e., its Jacobian matrix $\mathbf{Jf}(\mathbf{x}, \lambda) \in \mathbb{R}^{n \times (n+1)}$ is full row rank at every solution point $(\mathbf{x}, \lambda)$. When this assumption holds, the solution set M is a 1-dimensional C^2 manifold and constitutes a single solution branch. Examples of such behavior include a turning point (Figure 3.5a), two turning points forming a hysteresis curve (Figure 3.5b) and a hysteresis or cusp point (Figure 3.5c), at which $\mathbf{J_x f}(\mathbf{x}, \lambda)$ is singular but $\mathbf{Jf}(\mathbf{x}, \lambda)$ remains full row rank. In contrast, $\mathbf{Jf}(\mathbf{x}, \lambda)$ is rank deficient at a pitchfork bifurcation point (Figure 3.5d), where two solution branches intersect.

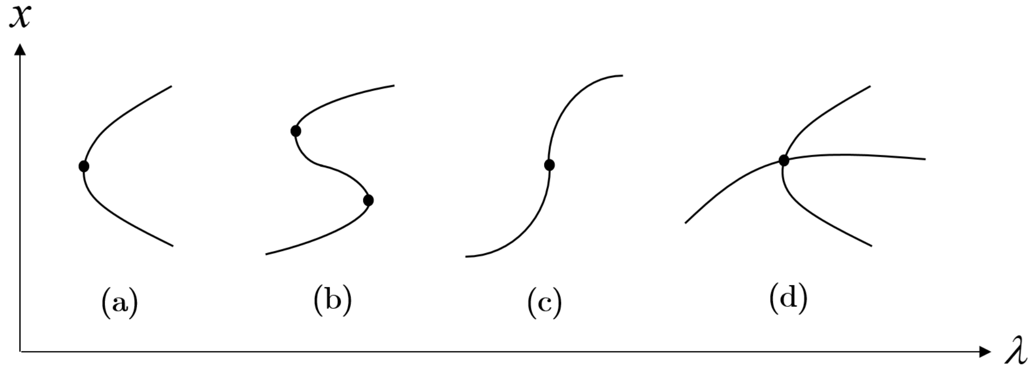


Figure 3.5: (a) Turning point; (b) Hysteresis curve (2 turning points); (c) Hysteresis point; (d) Pitchfork bifurcation point.

Starting from a known solution $(\mathbf{x}_k, \lambda_k)$ of Equation 3.30, the next point $(\mathbf{x}_{k+1}, \lambda_{k+1})$ on the solution branch is obtained in three steps, as schematically illustrated in Figure 3.6.

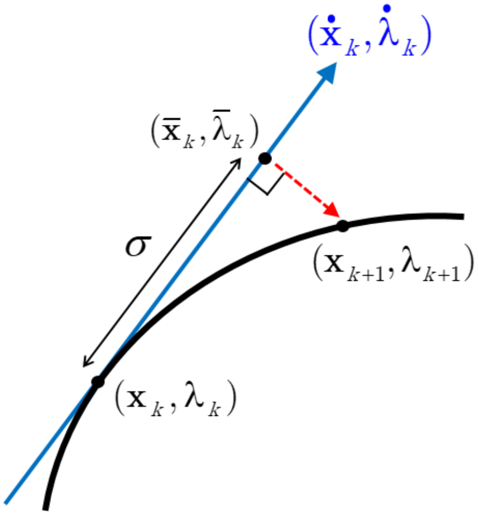

$(\dot{\mathbf{x}}_k, \dot{\lambda}_k)$
$(\bar{\mathbf{x}}_k, \bar{\lambda}_k)$
σ
$(\mathbf{x}_{k+1}, \lambda_{k+1})$
$(\mathbf{x}_k, \lambda_k)$

Figure 3.6: The pseudo-arclength continuation method.

3.6.1.1. Step 1: Obtain the unit tangent direction

A unit tangent vector $(\dot{\mathbf{x}}_k, \dot{\lambda}_k)$ to the solution branch at $(\mathbf{x}_k, \lambda_k)$ is obtained from the 1-dimensional null space of $\mathbf{Jf}(\mathbf{x}_k, \lambda_k)$. Its direction is chosen such that the continuation process moves in the same direction along the solution branch, which, according to Keller [44], can be done by enforcing a positive inner dot product between the current and previous tangent vectors:

$$(\dot{\mathbf{x}}_k)^{\mathrm{T}}\dot{\mathbf{x}}_{k-1} + \dot{\lambda}_k\dot{\lambda}_{k-1} > 0. \tag{3.33}$$

3.6.1.2. Step 2: Take a predictive step along the tangent direction

While arc length corresponds to the distance between any two points along the actual solution branch, pseudo-arclength is locally defined with respect to each point $(\mathbf{x}_k, \lambda_k)$ and corresponds to the distance traveled in the tangent direction determined by $(\dot{\mathbf{x}}_k, \dot{\lambda}_k)$. Starting from the current point $(\mathbf{x}_k, \lambda_k)$, we take a pseudo-arclength step of size σ to generate the point

$$(\bar{\mathbf{x}}_k, \bar{\lambda}_k) = (\mathbf{x}_k, \lambda_k) + \sigma(\dot{\mathbf{x}}_k, \dot{\lambda}_k), \tag{3.34}$$

which is an initial guess (or an Euler predictor) for the next point on the solution branch.

3.6.1.3. Step 3: Make an orthogonal correction

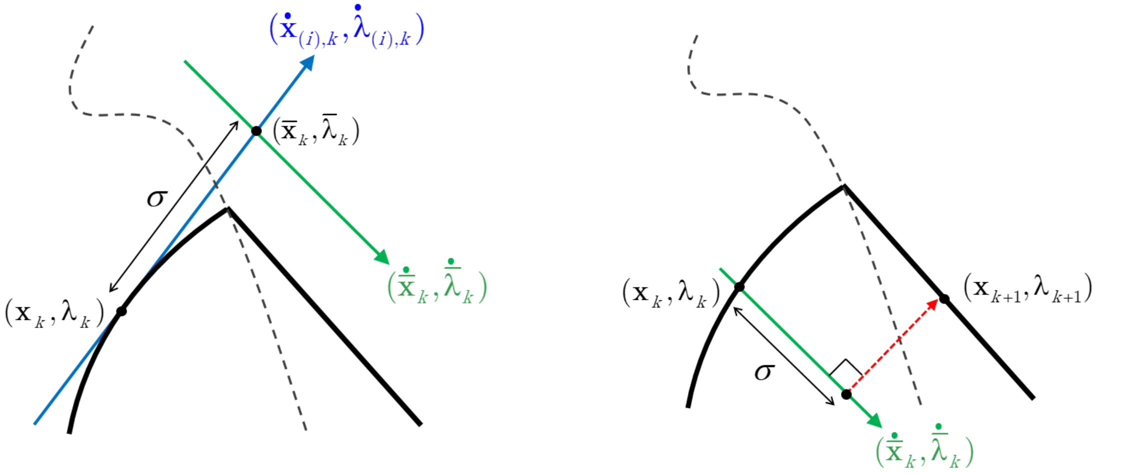
The next point $(\mathbf{x}_{k+1}, \lambda_{k+1})$ on the solution branch corresponds to the solution of the following augmented nonlinear system:

$$\mathbf{h}(\mathbf{x}, \lambda) = \begin{pmatrix} \mathbf{f}(\mathbf{x}, \lambda) \\ (\dot{\mathbf{x}}_k)^{\mathrm{T}} \cdot (\mathbf{x} - \mathbf{x}_k) + \dot{\lambda}_k \cdot (\lambda - \lambda_k) - \sigma \end{pmatrix} = \mathbf{0}. \tag{3.35}$$

This system can readily be solved with Newton's method, since its Jacobian matrix

$$\mathbf{Jh}(\mathbf{x}, \lambda) = \begin{pmatrix} \mathbf{Jf}(\mathbf{x}, \lambda) \\ \dot{\mathbf{x}}_k \quad \dot{\lambda}_k \end{pmatrix} \tag{3.36}$$

is guaranteed to remain invertible for $\sigma > 0$ small enough, and the predictor point $(\bar{\mathbf{x}}_k, \bar{\lambda}_k)$ is used as the initial guess. The first n equations in Equation 3.35 ensure that the next point lies on the solution manifold within numerical precision, and thus no integration errors are incurred. On the one hand, this allows for an adaptive step size strategy, in which σ can be increased by a suitable percentage whenever the Newton correction converges, and decreased otherwise until convergence is reestablished. On the other hand, in this work we chose instead to trace the solution branch in terms of its actual arc length; this can be achieved by keeping σ small enough so that the distance between consecutive solution points becomes numerically indistinguishable, within a 0.1% relative tolerance, from their pseudo-arclength distance. Finally, the last equation in Equation 3.35 geometrically enforces the next point to belong to a plane that is orthogonal to the tangent direction, and situated at an orthogonal distance σ from the current point $(\mathbf{x}_k, \lambda_k)$.


$(\dot{\mathbf{x}}_{(i),k}, \dot{\lambda}_{(i),k})$
$(\bar{\mathbf{x}}_k, \bar{\lambda}_k)$
σ
$(\mathbf{x}_k, \lambda_k)$
$(\dot{\bar{\mathbf{x}}}_k, \dot{\bar{\lambda}}_k)$
$(\mathbf{x}_k, \lambda_k)$
$(\mathbf{x}_{k+1}, \lambda_{k+1})$
σ
$(\dot{\bar{\mathbf{x}}}_k, \dot{\bar{\lambda}}_k)$

(a) Detecting a nonsmooth boundary.

(b) Updating the unit vector and orthogonal correction step.

Figure 3.7: PC^r pseudo-arclength continuation.

3.6.2. Nonsmooth PC^r systems

In this work, we extend the pseudo-arclength continuation method to PC^r functions $\mathbf{f} : \mathbb{R}^{n+1} \to \mathbb{R}^n$ for which the solution set to $\mathbf{f}(\mathbf{x}, \lambda) = \mathbf{0}$ is a 1-dimensional PC^r manifold. This means that the solution branch can only fail to be differentiable at isolated points, for which two distinct limiting tangent directions exist. We further assume that only two distinct limiting Jacobians $\mathbf{Jf}_{(i)}(\mathbf{x}, \lambda)$ can exist at the non-differentiable points; this assumption is satisfied by the nonsmooth MESH model. The modifications introduced into each of the three steps of the smooth pseudo-arclength method are described below and depicted in Figure 3.7.

3.6.2.1. Step 1: Obtain a limiting unit tangent direction

A limiting Jacobian matrix at the current point, $\mathbf{Jf}_{(i)}(\mathbf{x}_k, \lambda_k)$, is computed exactly with the method of Khan and Barton [46], and its 1-dimensional null space yields a limiting unit tangent vector $(\dot{\mathbf{x}}_{(i),k}, \dot{\lambda}_{(i),k})$ to the PC^r solution branch at $(\mathbf{x}_k, \lambda_k)$. We have found that requiring a positive dot product between subsequent limiting tangent vectors (Equation 3.33) is not a valid strategy in general to ensure the correct direction, since these pairs of vectors are often orthogonal. Instead, we have resorted to problem-specific information; for instance, for a drying column, the direction is chosen so as to decrease the liquid flow rates in the column.

3.6.2.2. Step 2: Take a predictive step, detect nonsmooth boundaries and update the direction

An Euler predictor step is taken with Equation 3.34 using $(\dot{\mathbf{x}}_{(i),k}, \dot{\lambda}_{(i),k})$, and the active selection function for the PC^r equation system is monitored. If the latter function changes, we know that the method has crossed a nonsmooth boundary in the domain of $\mathbf{f}$. We have found that convergence of Step 3 is unlikely in this case, since the orthogonal hyperplane corresponding to the limiting tangent direction on one side of the boundary might not intersect the solution branch on the other side. To address this problem in our case studies with the nosmooth MESH model, we compute a limiting Jacobian matrix $\mathbf{Jf}_{(j)}(\bar{\mathbf{x}}_k, \bar{\lambda}_k)$ at the predictor point. Its null space yields an updated unit vector $(\dot{\bar{\mathbf{x}}}_k, \dot{\bar{\lambda}}_k)$ (Figure 3.7a) that substitutes $(\dot{\mathbf{x}}_{(i),k}, \dot{\lambda}_{(i),k})$ and provides a different orthogonal hyperplane, which we have found to be more likely to intercept the solution manifold. The Euler predictor step is recomputed and we proceed to Step 3 with the updated unit vector $(\dot{\bar{\mathbf{x}}}_k, \dot{\bar{\lambda}}_k)$ (Figure 3.7b).

3.6.2.3. Step 3: Make a nonsmooth orthogonal correction

The augmented nonlinear system in Equation 3.35, which is now PC^r, is solved with the semismooth Newton method to generate the next point $(\mathbf{x}_{k+1}, \lambda_{k+1})$.

3.7. Bifurcations in dry/vaporless distillation columns

In this section, we present detailed parameter continuations for the two case studies previously considered in the relevant literature [16, 31] and describe new types of bifurcations observed in dry and/or vaporless multistage distillation columns, in light of the concept of PC^r manifolds proposed in Chapter 7. Several of these nonsmooth bifurcations exhibit degenerate behavior, with the occurrence of infinitely many steady-state solutions at certain input parameter values.

3.7.1. Case study 1: ideal binary mixture

A bubble-point liquid stream with 70% benzene, 30% toluene (% mol) is fed to Stage 6 of a column with $N = 27$ ideal stages, and the vapor and liquid phases are described by ideal thermodynamics (Raoult's Law). The intermediate stages are adiabatic and a linear pressure profile is specified, ranging from the total condenser (Stage 1) at 1.05 bar to the reboiler (Stage N) at 1.2 bar. The distillate-to-feed ratio is fixed at $D/F = 0.5$, and the feed flow rate is chosen as $F = 100$ mol/s. The only change made to the original case study [16, 31] is that we have switched the feed from Stage 20 to 6 to create clearer plots and illustrations of the drying process. For all case studies in this chapter, parameters for the K value and phase enthalpy correlations are retrieved from the Aspen Plus V10 database [2]. The nonsmooth model equations, and the nonsmooth equation-solving and pseudo-arclength continuation methods previously described were implemented in Matlab.

We fix the remaining degree of freedom in this system by specifying the reflux ratio R, which represents a single parameter λ on which the nonsmooth MESH model equations (represented by Equation 3.30) depend. Figure 3.8 illustrates how the steady-state solutions of the model vary as functions of R, in terms of the liquid flow rates coming out of the stages above the feed. When the value of R is high enough, all stages operate with vapor-liquid equilibrium (Phase Regime I); therefore, the original and nonsmooth MESH models have the same unique solution, which varies smoothly with respect to R.

In general, decreasing the reflux ratio causes the vapor and liquid flow rates throughout the column to diminish. The value of R at which one or more flow rates first become zero is denoted here as the critical reflux ratio R_{cr}. Similarly, we can define the critical value λ_{cr} for a general parameter λ. We avoid the "minimum reflux ratio" nomenclature used

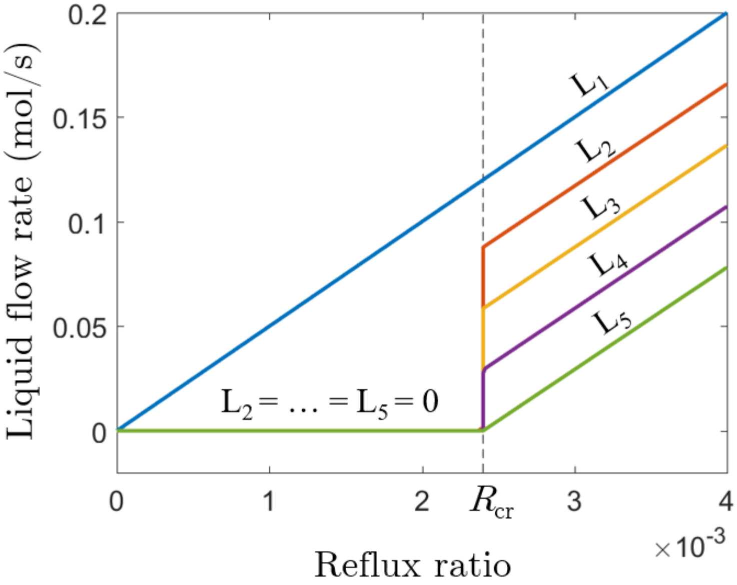


Figure 3.8: Above-feed liquid flow rates as functions of the reflux ratio R.

by Bullard and Biegler [16] because, as evidenced by Figure 3.8, it might be possible for the column to operate below R_{cr} while still satisfying all process specifications. Moreover, this phenomenon is not to be confused with the Underwood concept of the minimum reflux ratio to perform a separation, which corresponds to infinitely many stages.

For this particular case study, it is the liquid flow rate L_5 directly above the feed stage that disappears as we approach the critical reflux ratio $R_{\text{cr}} \approx 0.0024$ from above. We call the system state corresponding to $R \to R_{\text{cr}}^+$ the *upper critical solution*. In this state, Stage 5 is the only dry stage but it still operates in Phase Regime IIa, and thus the standard MESH equations are still satisfied. However, the MESH model yields a unique but non-physical solution for $R < R_{\text{cr}}$, with the above-feed liquid flow rates assuming negative values. On the other hand, the nonsmooth MESH model remains physically valid and reveals an unexpected behavior at R_{cr}, where a continuum of infinitely many steady states exist instead of a unique solution. This is evidenced by the vertical lines in the graphs of L_2, L_3 and L_4 at R_{cr} in Figure 3.8. For this reason, the overall model solution is discontinuous with respect to the reflux ratio at R_{cr}: the *lower critical solution* corresponding to $R \to R_{\text{cr}}^-$ is different from the upper critical solution ($R \to R_{\text{cr}}^+$).

The non-uniqueness of solutions at R_{cr} gives rise to singular limiting partial Jacobians of the model equations $\mathbf{f}(\mathbf{x}, R) = \mathbf{0}$ with respect to $\mathbf{x}$, and therefore the semismooth Newton method fails at or near R_{cr}. In order to obtain these infinitely many steady states, which range from the lower to the upper critical solutions, we had to develop the previously described nonsmooth version of the pseudo-arclength continuation method. By taking small enough continuation steps, we can trace the solutions in terms of arc length as a substitute parameter, which corresponds to the distance traveled along the solution curve in $(\mathbf{x}, R)$ space. Geometrically, arc length acts as an extra coordinate that allows us to "move" perpendicularly to the paper at the vertical line $R = R_{\text{cr}}$. Moreover, since a unique steady-state solution exists at each value of arc length, the latter constitutes a more adequate parameter than R to describe the overall set of solutions.

Figure 3.9 portrays the complete curve of solutions in terms of its arc length starting from $R = 0$, with the corresponding values of reflux ratio represented by the juxtaposed R axis. The vertical line $R = R_{\text{cr}}$ from Figure 3.8 is expanded horizontally in terms of arc length in Figure 3.9 to reveal an overall solution curve that is continuous, but non-differentiable at several points between (and including) the upper and lower critical solutions. The state of the distillation column is schematically represented in Figure 3.9 for all the nonsmooth points, which correspond to when each stage first becomes dry.

Exactly at the upper critical solution ($R \to R_{\text{cr}}^{+}$), Stage 5 is dry in Phase Regime IIa and V_5 is a dew-point vapor. As we continue tracing the solutions towards smaller values of arc length, L_4 starts decreasing in the same amount that V_5 increases, keeping Stage 5 in mass balance. Concurrently, T_5 increases and makes the vapor V_5 progressively more superheated, putting Stage 5 into Phase Regime IIIa. This is a completely local process, in which only the variables directly associated with Stage 5 change. When L_4 finally reaches zero, we arrive at the next nonsmooth point represented in Figure 3.9, and further decreasing of arc length initiates this same local process around Stage 4. This way, as we traverse the solutions from $R \to R_{\text{cr}}^{+}$ to $R \to R_{\text{cr}}^{-}$, the above-feed stages 2-5

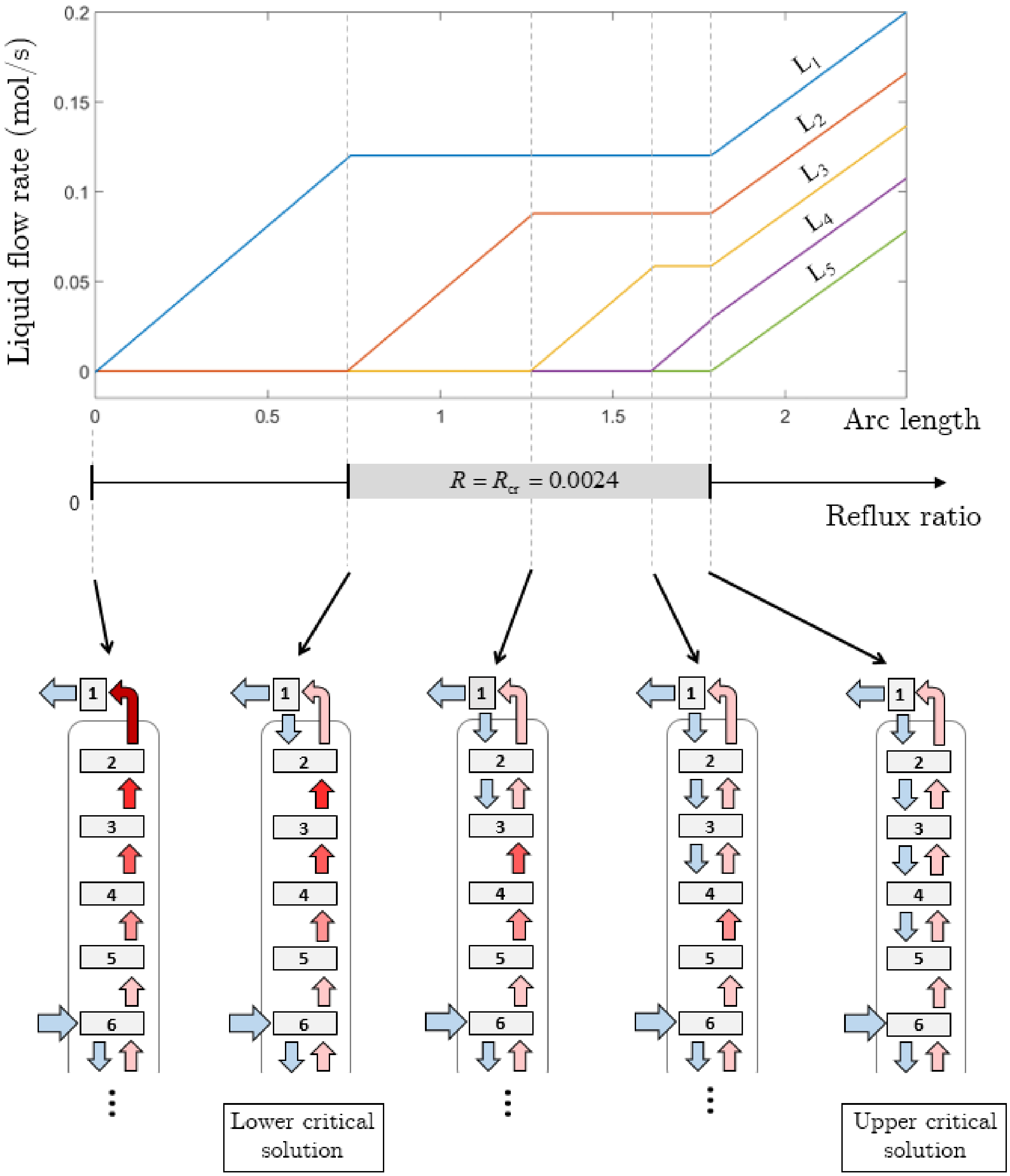


Figure 3.9: Above-feed liquid flow rates as functions of the arc length of the solution curve, with schematic representations of the column at non-differentiable points.

become dry one at a time, sequentially from bottom to top. For $R < R_{\text{cr}}$, decreasing the reflux ratio leads to smaller values of L_1, until the latter reaches zero at $R = 0$. However, the condenser does not become dry because of the specified liquid distillate output $W_{L,1}$.

Darker shades of red in Figure 3.9 represent the degree of “superheating” of the vapor streams, which increases as we go up the column due to the pressure drop in each stage.

For negative values of the reflux ratio ($R < 0$), the standard MESH model continues to have a unique mathematical solution, which is nevertheless not physically valid because $L_1 = RD$ and several other liquid flow rates become negative. On the other hand, the nonsmooth MESH model has no mathematical solution for $R < 0$, since its equations bound L_1 to be non-negative, and therefore reflects the behavior of the physical system.

3.7.1.1. The bifurcations at $R = R_{\mathbf{cr}}$ and $R = 0$

Recall the representation in Equation 3.30 of the nonsmooth MESH model equations as depending on a single parameter λ, which here corresponds to R. We can say that a 1 - ∞ - 1 bifurcation exists at R_{cr}, since the number of model solutions for each value of R change from 1 for $R < R_{\mathrm{cr}}$, to infinitely many at $R = R_{\mathrm{cr}}$, and back to 1 for $R > R_{\mathrm{cr}}$. Similarly, we observe a 0 - 1 - 1 bifurcation at $R = 0$, where the solution branch ends abruptly. We can represent the essential aspects of both of these bifurcations in Figure 3.10 by plotting just the liquid flow rate L_4 against R. However, the behavior of the overall solution set cannot be represented by every one of the variables; for instance, the graph of L_5 does not reveal the occurrence of the bifurcation at R_{cr}. Moreover, note that the intermediate nonsmooth points between the upper and lower critical solutions cannot be observed in terms of L_4 only in Figure 3.10.

We can also observe a bifurcation of type 1 - ∞ - 1 in very simple smooth systems. For instance, for $f(x, \lambda) = \lambda x = 0$, there is a unique solution $x = 0$ for $\lambda \neq 0$ but the whole real line $x = \mathbb{R}$ of solutions at $\lambda^* = 0$. What makes the bifurcation at $R = R_{\mathrm{cr}}$ unique and novel is that the overall solution set remains a 1-dimensional and connected PC^r manifold, as illustrated in Figure 3.10. Moreover, the set of infinitely many solutions at R_{cr} is bounded, and in this case consists of a 1-dimensional and connected PC^r manifold with two boundary points, the upper and lower critical solutions. Intuitively, this type of

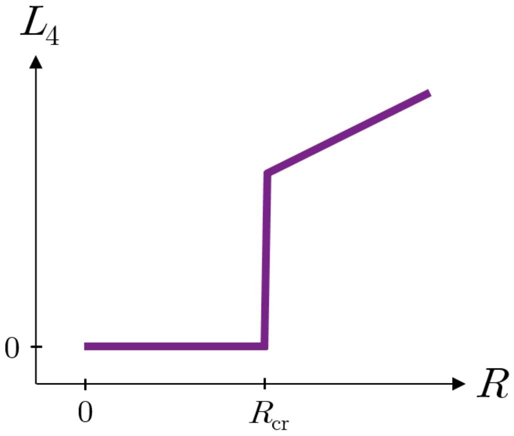


Figure 3.10: The 0 - 1 - 1 and 1 - ∞ - 1 bifurcations at $R = 0$ and $R = R_{\mathrm{cr}}$, respectively, in terms of L_4.

bifurcation can be thought of as a *hysteresis region*: a hysteresis point (Figure 3.5c) that has been "stretched" vertically into a whole segment of constant $R = R_{\mathrm{cr}}$. Note how this differs from the smooth hysteresis curve (Figure 3.5b) observed in most other distillation systems with bifurcations, which are modeled with the (smooth) MESH equations.

Mathematically, for each of the infinitely many solutions at $R = R_{\mathrm{cr}}$, at least one of the limiting partial Jacobian matrices $\mathbf{J_x f}_{(i)}(\mathbf{x}, R_{\mathrm{cr}})$ with respect to the variables $\mathbf{x}$ is singular, with a rank 1 deficiency. For the nonsmooth MESH model of a distillation column, these singularities arise from complex interactions between the model equations of types M, E, S and H of several stages. Intuitively, the singularities are associated with an extra degree of freedom for the model equations, which appears only at R_{cr} and makes the system momentarily underdetermined.

On the other hand, the limiting partial Jacobian matrices $\mathbf{J_x f}_{(i)}(\mathbf{x}, R)$ at $R = 0$ remain invertible despite the occurrence of a bifurcation, and therefore the semismooth Newton method can be used to solve for $\mathbf{x}$ directly at or around $R = 0$. In this case, singular matrices are only present in the Clarke Jacobian set of $\mathbf{f}$, which corresponds to the set of all convex combinations of the limiting Jacobian matrices $\mathbf{J_x f}_{(i)}(\mathbf{x}, 0)$.

3.7.1.2. The choice of parameter and the bifurcation at $B = B_{\text{cr}}$

We can analyze how the solutions of the nonsmooth MESH model change with respect to any other parameter λ. If we choose to specify and vary the boilup ratio $B = V_N/L_N$ instead of R, we also arrive at a critical boilup ratio B_{cr} at which the first flow rate in the column becomes zero. Figure 3.11 presents the above-feed liquid flow rates for Case Study 1 in terms of B, with $B_{\text{cr}} \approx 1.0108$. Even though the overall solution set remains the same, its representation in terms of B gives rise to a different bifurcation at B_{cr} of type 0 - $/\infty$ - 1; here, $/\infty$ indicates that the set of infinitely many solutions at the bifurcation parameter B_{cr} is bounded and ends abruptly on one end. The upper critical solution solution ($R \to R_{\text{cr}}^{+}$) now corresponds to $B \to B_{\text{cr}}^{+}$, and the solution for $R = 0$ would correspond to approaching B_{cr} from below. The essential aspects of this bifurcation can be described by the graph of L_4 versus B in Figure 3.12; the solution set at B_{cr} is bounded and consists of a 1-dimensional PC^r manifold with two boundary points.

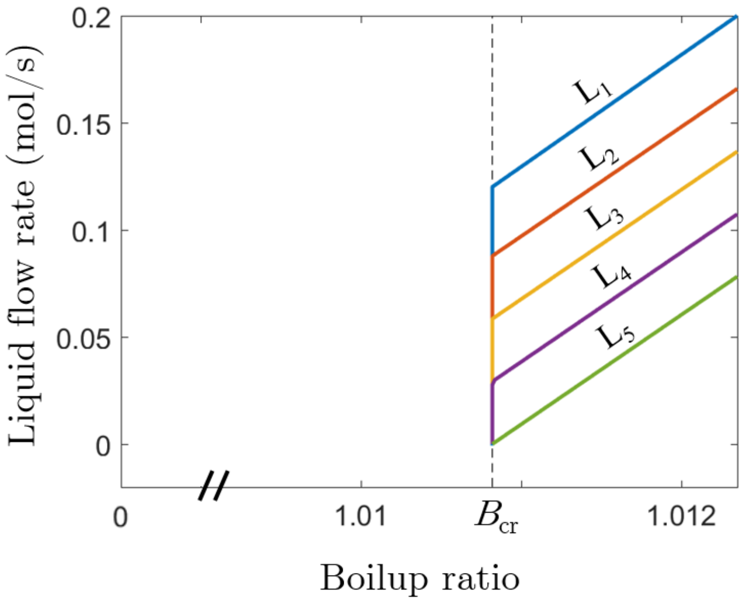


Figure 3.11: Above-feed liquid flow rates as functions of the boilup ratio B.

Since no solutions exist below the critical boilup ratio, B_{cr} represents a positive lower bound for the parameter that cannot be predicted prior to simulation. This happens

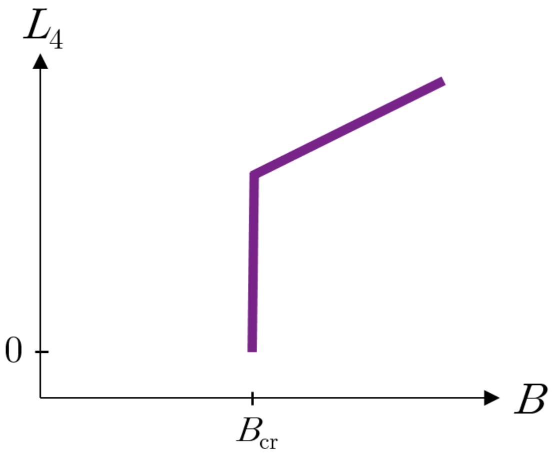


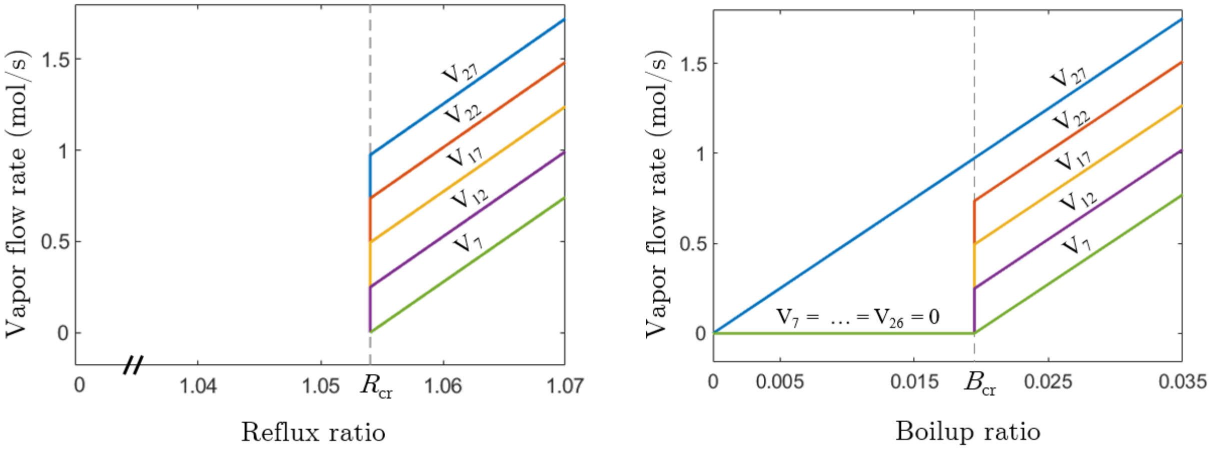
Figure 3.12: The 0 - /∞ - 1 bifurcation at B_{cr} in terms of L_4.

because, once L_5 becomes zero at the upper critical solution, the distillation column is split into two halves (Stages 1-4 and Stages 5-27). Changes in the upper half, such as decreases in the value of R, can no longer impact the lower half, whose describing variables (including $B = B_{\mathrm{cr}}$) remain the same for $0 \leq R \leq R_{\mathrm{cr}}$.

3.7.1.3. Vapor feed

Now, consider the same specifications in Case Study 1 except that the feed stream is a dew-point vapor, directly introduced into Stage 6. Figure 3.13 presents some of the vapor flow rates in the below-feed column section as functions of either R or B. In this case, it is the *vapor phase* that disappears, but only in the stages below the feed. We can observe a behavior that is mostly analogous to the dry column case, except that the roles of R and B are switched; the critical points for this system are $R_{\mathrm{cr}} \approx 1.054$ and $B_{\mathrm{cr}} \approx 0.0195$. The below-feed stages become sequentially vaporless from top to bottom, starting with $V_7 = 0$ at $B \to B_{\mathrm{cr}}^{+}$, then $V_{26} = 0$ at $B \to B_{\mathrm{cr}}^{-}$, and finally with the reboiler "turning off" and becoming vaporless ($V_{27} = 0$) at $B \to 0^{+}$. However, the solution curve behavior at $B = 0$ for a vaporless column is qualitatively different than that of a dry column at $R = 0$.

No solutions exist for $B < 0$, since the mid equation (Equation 3.20) for the reboiler



(a)

(b)

Figure 3.13: Below-feed vapor flow rates as functions of (a) the reflux ratio R; (b) the boilup ratio B.

ensures $L_N > 0$. However, at $B = 0$, the reboiler is vaporless and thus the liquid bottoms product L_N is mathematically allowed to become subcooled. This generates infinitely many solutions associated with a negative reboiler heat duty Q_N, and gives rise to a 0 - ∞ - 1 bifurcation both at $B = 0$ and at $R = R_{\text{cr}}$. This type of bifurcation differs from the 0 - /∞ - 1 bifurcation depicted in Figure 3.12 because the solution set is now unbounded at the bifurcation parameter, and consists of a connected 1-dimensional PC^r manifold with a single boundary point (the upper critical solution). The essential aspects of the 0 - ∞ - 1 bifurcations at $B = 0$ and at $R = R_{\text{cr}}$ can only be represented in terms of some of the reboiler variables, such as the graph of the reboiler temperature T_N versus the boilup ratio B at $B = 0$ in Figure 3.14.

Note that a negative reboiler heat duty is physically realizable in terms of heat exchange. However, we can choose to eliminate these infinitely many solutions mathematically by bounding Q_N to be positive, which can be attained by employing a nonsmooth equation analogous to Equation 3.29.

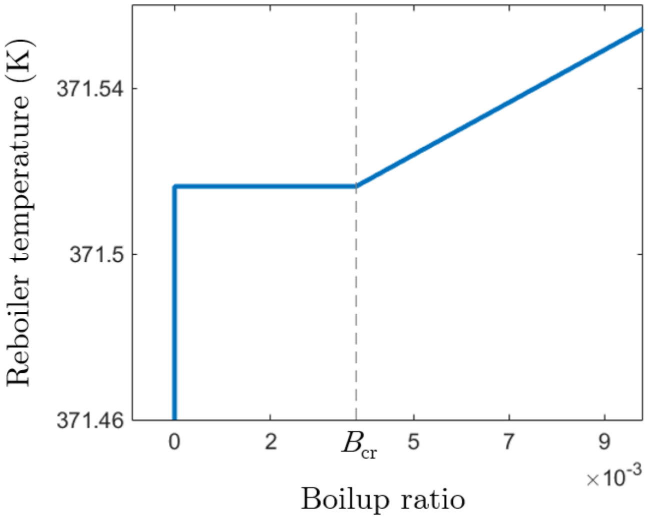


Figure 3.14: Reboiler temperature T_N as a function of the boilup ratio B.

3.7.2. Case Study 2: 5-component non-ideal mixture

A bubble-point liquid stream composed of 15% methanol, 40% acetone, 5% methyl acetate, 20% benzene and 20% chloroform (% mol) is fed to Stage 7 of a column with $N = 19$ ideal stages. The UNIQUAC activity model is used for the liquid phase, and the Hayden-O'Connell correlation is used to compute the second virial coefficients that model the vapor phase fugacity. The feed-to-distillate ratio is fixed at $D/F = 0.3$, the intermediate stages are adiabatic, and the linear pressure profile ranges from 1.1 bar at the reboiler to 1.015 bar at the total condenser. This system was also considered in the same papers previously mentioned [16, 31].

Due to its low feed concentration, methyl acetate reaches very small vapor phase mole fractions in the above-feed stages, to the point that its fictitious liquid mole fractions defined via Formulation 1 (Equations 3.19, 3.20) would have to be negative in the dry stages. This precludes convergence of the model equations, since the K value correlations contain logarithmic terms that would become undefined. Instead, Formulation 2 can be used to trace all model solutions, with the variable β present either explicitly through Equation 3.13 (Formulation 2a) or implicitly through Equations 3.14 and 3.15 (Formulation 2b). In both cases, it is β that changes to reflect deviations from liquid-vapor equilibrium, with the fictitious liquid mole fractions remaining approximately constant. We note that convergence with Formulation 2a is much more robust, which comes at the price of adding an extra β_j variable to each stage j. Formulation 2b can become numerically unstable depending on the component chosen as $i = 1$ in Equation 3.14, including when the most volatile component (acetone) is chosen, and thus smaller continuation steps must be used.

Using either form of Formulation 2, we can observe the same qualitative behavior in this 5-component system as seen in the binary Case Study 1, with stages becoming dry (for a liquid feed) or vaporless (for a vapor feed) through the same previously detailed bifurcations. Therefore, for the sake of brevity, we omit the plots and report only the critical parameter values: $R_{\text{cr}} \approx 0.0013$ and $B_{\text{cr}} \approx 0.4382$ when the feed is a bubble-point liquid, and $R_{\text{cr}} \approx 2.2976$ and $B_{\text{cr}} \approx 0.0068$ when the feed is a dew-point vapor.

3.8. Analysis of the bifurcations

The bifurcations that occur at the critical parameter values represent the transition of some of the column stages from Phase Regime I (vapor-liquid) into Phase Regimes III (superheated vapor or subcooled liquid), and are intrinsic to cascades of equilibrium stages. In this section, we use the original Case Study 1 (with a liquid feed) as a basis for comparison and introduce several modifications into the column configuration and types of specifications, in order to analyze which factors can influence and give rise to these bifurcations.

3.8.1. Type of modeling approach

Any modeling approach capable of enforcing the necessary MESH-based physical laws, with equilibrium relationships only between the phases that are actually present, will invariably lead to the same steady-state solutions and bifurcations described in this chapter. In this sense, all three different formulations using the explicitly nonsmooth, PC^{∞} function mid that we have employed (Equations 3.12, 3.13, and 3.14-3.15) are equally valid. Equivalently, we could have used the complementarity constraint modeling approach represented by Equations 3.9-3.11, by first rewriting them with explicitly nonsmooth equations (e.g. the Fischer-Burmeister formulation) and then employing our pseudo-arclength continuation method to yield the same steady-state solutions. However, this approach would be more costly, given that it includes the additional variables s_V, s_L and β for each stage.

3.8.2. The critical parameter value

The critical parameter value λ_{cr} is particular to each system and cannot be predicted or computed with the MESH equations in general, since the location of the first flow rate(s) to equal zero cannot be predicted. However, λ_{cr} and the upper critical solution can be readily computed by replacing the specification equation for the parameter in the standard MESH model,

$$\lambda - \lambda_{\mathrm{user}} = 0, \tag{3.37}$$

with the explicitly nonsmooth equation

$$\min\Big(\min_j L_j, \min_j V_j\Big) = 0. \tag{3.38}$$

3.8.3. Mixture components and thermodynamic models

Our numerical experiments have shown that all bifurcations presented in this chapter are independent of the number and identity of components, and of the thermodynamic model used, which was also illustrated by Case Study 2. For instance, employing the Peng-Robinson equation of state in the original Case Study 1 maintains the same bifurcation behaviors, while the critical parameter values change slightly ($R_{\mathrm{cr}} \approx 0.00230$, $B_{\mathrm{cr}} \approx 1.007$). The only issue to consider, as discussed in Case Study 2, is that certain fictitious mole fraction formulations might preclude convergence of the model equations at or below the critical parameter value, depending on the mixture.

3.8.4. Pressure gradients and tray heat loss

Mathematically, the degenerate bifurcations at the critical parameter values will only occur if "external driving forces" are included in the column specifications, in the form of a pressure gradient and/or external heat transfer in the trays (i.e., intermediate stages). Without at least one of these imposing forces, the vapor and liquid streams cannot drive themselves out of saturation and thus the column stages cannot reach Phase Regimes III. In such cases, there's a unique solution at the critical parameter value, corresponding to when one or more stages first become dry/vaporless but remain in Phase Regimes II.

To demonstrate that, consider Case Study 1, now with a uniform column pressure of 1 bar. As Figure 3.15 illustrates, the stages above the feed can only become dry simultaneously, with a unique critical solution at $R = R_{\text{cr}} = 0$. In terms of the boilup ratio, this unique solution corresponds to $B_{\text{cr}} \approx 0.986$. Further decreasing of either R or B would necessarily lead to negative liquid flow rates, which is not allowed by the nonsmooth MESH model; as a result, we have a 0 - 1 - 1 bifurcation both at R_{cr} and B_{cr}.

On the other hand, we can specify a uniform column pressure and still observe the same 1 - ∞ - 1 and 0 - $/\infty$ - 1 bifurcations at R_{cr} and B_{cr}, respectively, by including non-zero tray heat duties Q_j. If we impose a uniform column pressure of 1 bar and include a heat gain $Q_j \approx 9.6 \cdot 10^2$ J/s in all trays ($2 \leq j \leq N-1$) in Case Study 1, we are able to reproduce essentially the same solutions and corresponding bifurcations of the original case study, with the same R_{cr}, B_{cr} values. In the case of a vapor feed, the original bifurcations with vaporless stages are recovered by imposing an external heat loss in the trays.

In a real distillation column, a significant fraction of the tray pressure drop is due to vapor flow through tray perforations. Therefore, we can expect a pressure gradient to be present in dry stages, creating the mathematical conditions for the occurrence of infinitely many steady states with superheated vapor streams at the critical parameter values. Additionally, tray columns are not perfectly insulated and heat exchange with the environment occurs at every stage. In above-ambient-temperature processes, tray heat losses could allow the liquid in vaporless stages to become subcooled, providing the necessary conditions for infinitely many vaporless states to be observed.

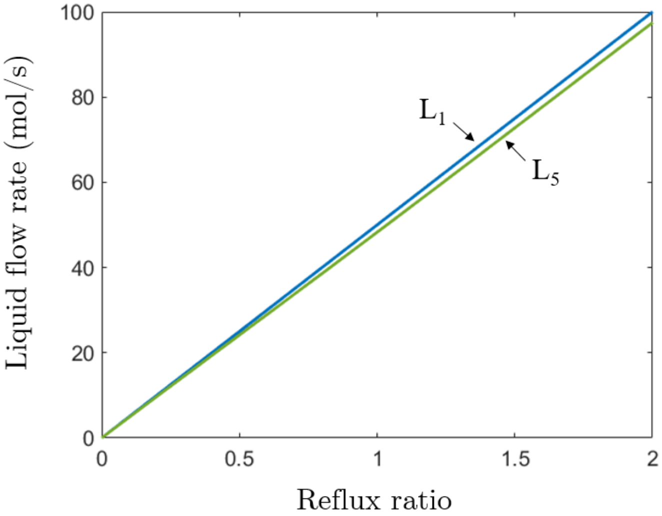


Figure 3.15: Liquid flow rates L_1 and L_5 versus R for Case Study 1 with a uniform column pressure of 1 bar.

3.8.5. Stage efficiencies

All numerical examples presented thus far involve ideal stages. If, instead, we specify a vaporization efficiency of 30% in all intermediate stages in Case Study 1, we still observe the same bifurcations previously presented, only at different critical parameter values ($R_{\text{cr}} \approx 0.00308$, $B_{\text{cr}} \approx 2.000$). On the other hand, if we specify vapor-phase Murphree tray efficiencies of 30%, we obtain the slightly different behavior depicted in Figure 3.16. In this case, mathematically there exists a unique solution for each value of R and therefore no bifurcation is present at $R_{\text{cr}} \approx 0.0024$. However, the limiting partial Jacobian matrices $\mathbf{J_x f}_{(i)}(\mathbf{x}, R)$ become extremely ill-conditioned, despite being non-singular, and the resulting behavior in Figure 3.16 remains essentially the same. We observe an extremely abrupt change in steady-states at R_{cr}, with graphs that still appear to be vertical. Finally, we note that the numerical values chosen for the tray efficiencies (either of the vaporization or Murphree types) do not change the types of behaviors observed. The difference in results depending on the type of efficiency chosen suggests that the functional form of the equilibrium relationship (Equation 3.19), maintained when using vaporization efficiencies, is necessary for the mathematical existence of infinitely many solutions.

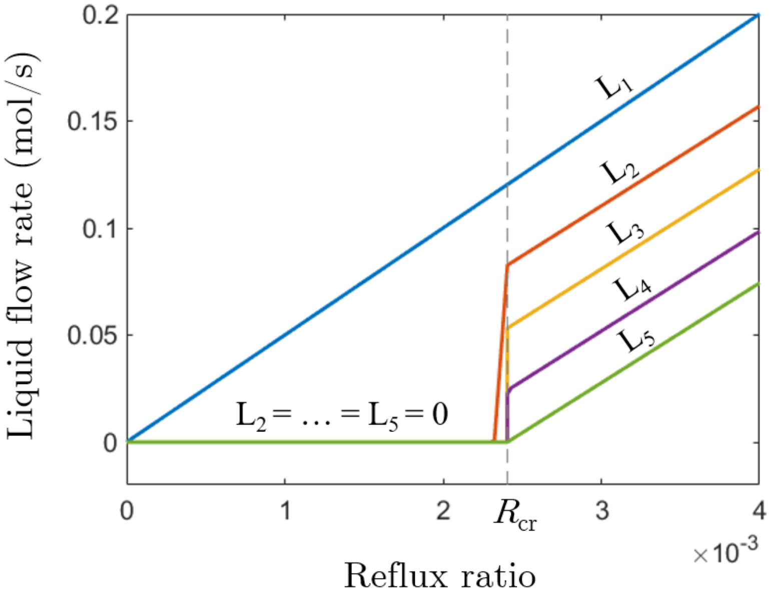


Figure 3.16: Above-feed liquid flow rates versus R for Case Study 1 with vapor-phase Murphree efficiencies of 30%.

Notwithstanding, we must take into account that stage efficiencies are only simplified descriptions of non-ideal vapor-liquid mass transfer in real stages, with shortcomings that become more pronounced in multicomponent, non-ideal mixtures [79]. Moreover, Murphree stage efficiencies are known to become undefined and/or physically incorrect in several instances, while vaporization efficiencies can be shown to, at least, always remain well-defined [37]. For these reasons, column simulation in distillation design is customarily performed with ideal stages, with stage efficiencies being estimated and included post-simulation only to yield an updated number of trays. In contrast, the non-equilibrium or rate-based modeling strategy is much better suited to describe mass transfer effects in real stages, and thus a nonsmooth version thereof could be more reliable in predicting the steady-state behavior of dry/vaporless columns.

However, the rate-based approach relies on correlations for mass and heat transfer, interfacial areas and other parameters, which depend on knowledge of column and tray design details [80], and are expected to be valid only when both vapor and liquid phases are present. On the other hand, in general the efficiency/equilibrium stage approach still remains quite accurate to describe binary, close-boiling ideal mixtures [79], such as the benzene-toluene system from Case Study 1. As demonstrated in this section, the fact that the infinitely many steady-state solutions of the nonsmooth MESH model persist under different types of column specifications, stage efficiencies, mixture components and thermodynamic models at least supports the hypothesis that some type of degenerate or near-degenerate behavior could be observed experimentally.

3.8.6. The feed state

It is the state of the feed stream(s) that determines which phase(s) can disappear in the column. We illustrate this for the original Case Study 1 in Figure 3.17, which shows how R_{cr} changes as we vary the feed temperature, and the corresponding phase regimes at, above or below the critical reflux curve. A change in regimes occurs at the feed temperature $T_f^* = 366.07K$, which is between the bubble and dew-point temperatures $T_{\text{bubble}} = 362.0K$ and $T_{\text{dew}} = 367.9K$ of the feed mixture. For feed temperatures below T_f^*, the column goes dry above the feed stage for $R \leq R_{\text{cr}}$, starting with $L_5 = 0$, in the same fashion depicted in Figure 3.8. For higher feed temperatures, the column becomes vaporless below the feed stage at $R = R_{\text{cr}}$ in the same fashion of Figure 3.13, starting with $V_7 = 0$, and smaller reflux ratios $R < R_{\text{cr}}$ are infeasible.

However, exactly at the transitional feed temperature T_f^*, the column becomes simultaneously dry above the feed and vaporless below the feed for $R \leq R_{\text{cr}}$, starting with $L_5 = V_7 = 0$. In this case, any of the infinitely many vaporless states of the below-feed

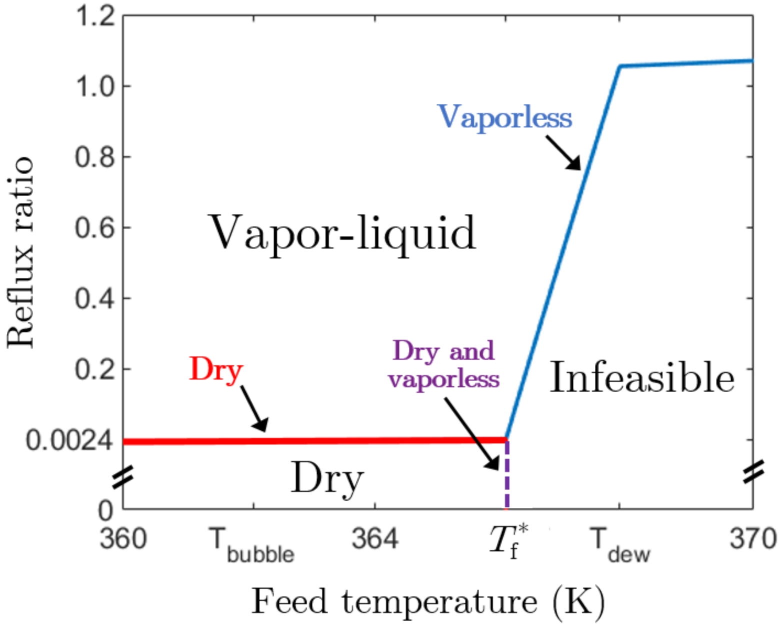


Figure 3.17: Critical reflux ratio R_{cr} versus feed temperature T_f for Case Study 1, with phase regimes for each (T_f, R) pair.

part of the column can occur simultaneously with any dry state of the above-feed section corresponding to $0 < R \leq R_{\text{cr}}$. As a result, we arrive at another type of nonsmooth bifurcation with a higher degree of degeneracy. The main aspects of this complex behavior can be illustrated by Figure 3.18, where L_4 (representing above-feed states) and V_8 (representing below-feed states) are plotted against R at $T_f = T_f^*$. At $R = R_{\text{cr}}$, the solution set is a 2-dimensional PC^r manifold of steady states instead of a 1-dimensional PC^r manifold. For each individual reflux ratio $0 \leq R < R_{\text{cr}}$ there are infinitely many vaporless steady states forming a 1-dimensional PC^r manifold, whereas the overall solution set for $0 \leq R \leq R_{\text{cr}}$ is a 2-dimensional PC^r manifold containing dry *and* vaporless steady states.

While all other bifurcations presented in this chapter have codimension 1 and thus require only a single parameter to be varied, the bifurcation at $R = R_{\text{cr}}, T_f = T_f^*$ is of codimension 2: two parameters are involved simultaneously, the reflux ratio and the feed temperature. Therefore, it is much less likely to be observed in practice during simulation of the nonsmooth MESH model.

Figure 3.19 presents the phase regimes in terms of the boilup ratio versus T_f for the

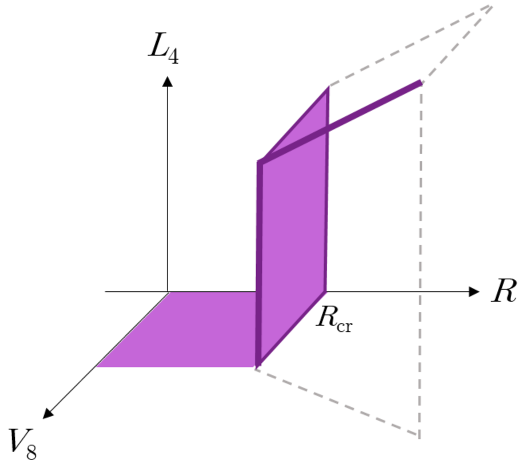


Figure 3.18: The codimension-2 bifurcation at $T_f = T_f^*$, in terms of L_4 and V_8 versus R.

same system, and we see a qualitatively analogous behavior. We can also conclude from Figures 3.17 and 3.19 that the magnitude of the critical parameter value is not necessarily small and depends on several factors. As a result, engineers or solution algorithms could inadvertently choose input parameters close to or below their critical values during column simulation and optimization, leading to failure of any current MESH-based software.

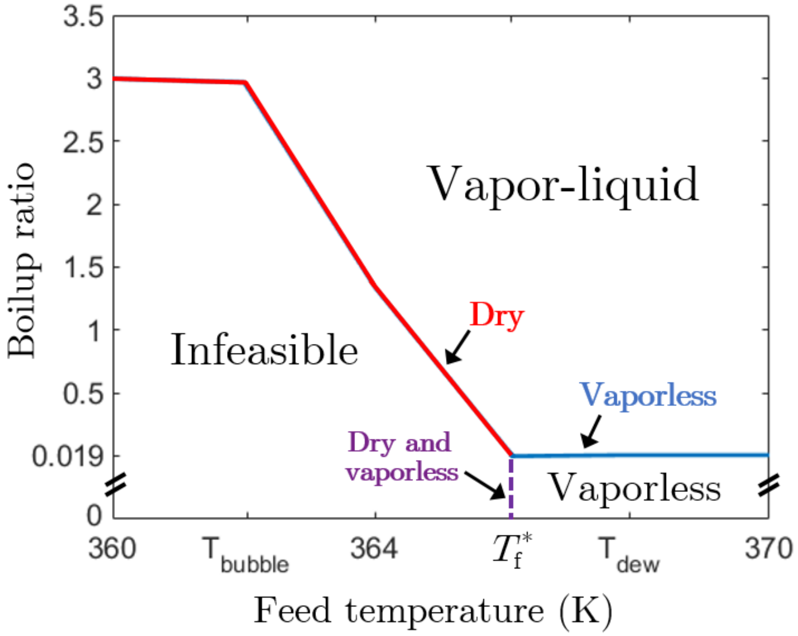


Figure 3.19: Critical boilup ratio B_{cr} versus feed temperature T_f for Case Study 1, with phase regimes for each (T_f, B) pair.

3.8.7. Multiple feeds

When multiple feeds are present, the location and phase (vapor/liquid) of the first vanishing stream is not obvious and cannot be predicted prior to simulation, but is usually directly above or below one of the feed stages since flow rates tend to vary monotonically in each column section.

To exemplify the different types of behavior that can be observed, consider the original Case Study 1 with a bubble-point liquid fed into Stage 6. If we introduce a second liquid feed stream into a stage above, say, Stage 4, then the same type of bifurcation from Figure 3.8 now happens above this second feed only, with Stages 2-3 becoming dry. If instead we introduce a second vapor feed below Stage 6, say, at Stage 8, the type of bifurcation observed at R_{cr} might change according to the behavior in Figure 3.17, with the horizontal axis now representing the vapor feed flow rate F_8. That is, for large enough values of F_8, the column is vaporless below Stage 8 at $R = R_{\text{cr}}$, and for small enough F_8 values, it remains dry above Stage 6 for $R \leq R_{\text{cr}}$. At a transitional vapor feed flow rate F_8^*, the column is simultaneously dry above Stage 6 and vaporless below Stage 8 for $R \leq R_{\text{cr}}$.

3.8.8. The type of condenser and the bifurcation at $R = 0$

The nonsmooth bifurcation that occurs at $R = 0$ in a column with dry stages depends on the type of condenser specified. As long as the vapor distillate fraction θ is smaller than 1, a non-zero amount of liquid distillate W_{L_1} is present and the condenser never goes dry. As a result, the solution curve stops at $R = 0$ and we observe the same 0 - 1 - 1 bifurcation described in the original Case Study 1.

However, if $\theta = 1$, the condenser becomes dry at $R = 0$ and the outlet vapor V_1 can become superheated, which is associated with a positive condenser heat duty Q_1. The solution curve is allowed to continue varying at $R = 0$, as the condenser temperature T_1 progressively increases, and we observe a 0 - ∞ - 1 bifurcation, depicted in Figure 3.20

for Case Study 1. This bifurcation is analogous to that observed at $B = 0$ for the reboiler in a vaporless column, which was illustrated in Figure 3.14.

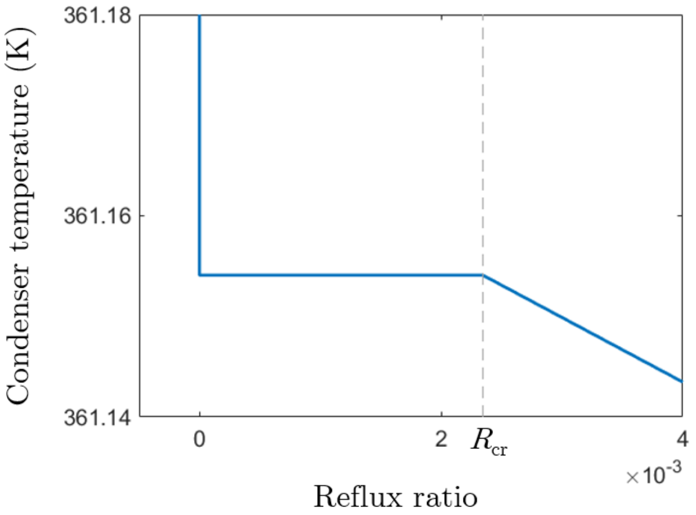


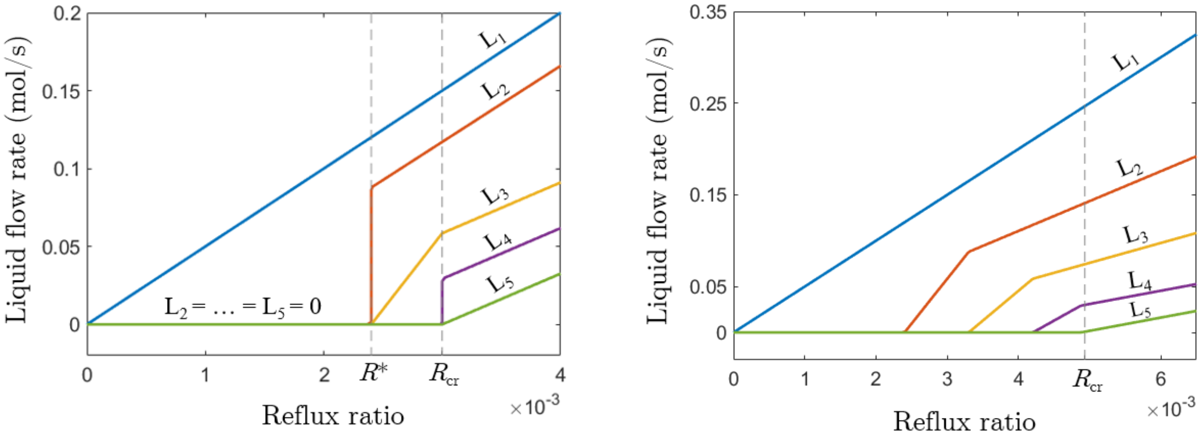
Figure 3.20: Condenser temperature T_1 versus R for Case Study 1 with $\theta = 1$.

3.8.9. Side products

The type of bifurcations occurring at the critical parameter values might change when we withdraw side products from intermediate stages, depending on their phase (vapor/liquid) and location relative to the feed streams. We illustrate the possible behaviors with Case Study 1, in which a liquid feed is introduced into Stage 6. If a vapor side product is included anywhere in the column or if a liquid side product is withdrawn from a stage below the feed, the types of bifurcations previously presented remain the same. However, the system behavior changes if we include a liquid side product above the feed, depending on the type of specification chosen and stage location.

3.8.9.1. Withdrawal ratio specification

If we specify a liquid withdrawal ratio $R_{L,j}$ for an intermediate stage j above the feed, the stage's drying process can proceed normally due to Equation 3.25 but is no longer degenerate, as illustrated in Figure 3.21a for Stage 3 with $R_{L,3} = 0.5$. At the new critical reflux ratio value $R_{\text{cr}} \approx 0.0030$, we observe infinitely many solutions due solely to the drying process of Stage 4. The drying process of Stage 3 happens with a unique solution for each R, until we reach a second reflux ratio value $R^* = 0.0024$ with infinitely many solutions corresponding to the drying process for Stage 2. Therefore, two 1 - ∞ - 1 bifurcations happen in series, at R_{cr} and R^*. Taking a step further, if we specify the same liquid withdrawal ratio $R_{L,j} = 0.5$ for Stages 2, 3 and 4, we can eliminate the occurrence of infinitely many solutions and the corresponding bifurcations altogether, as shown in Figure 3.21b.



(a) (b)

Figure 3.21: Above-feed liquid flow rates versus R for Case Study 1 with (a) $R_{L,3} = 0.5$, (b) $R_{L,j} = 0.5$, $j = 2, 3, 4$.

3.8.9.2. Flow rate specification

When we specify a liquid side product flow rate $W_{L,j}$ in stage j above the feed, the stage can never become dry. If the stage's liquid outlet L_j reaches zero, its corresponding mid equation (Equation 3.20) does not switch between its arguments and remains enforcing the summation relationship. Once $L_j = 0$ in the continuation process, the only way to continue tracing solutions would be with $L_j < 0$, which is not allowed by the specification Equation 3.26. Therefore, the nonsmooth MESH model ceases to have a solution, which is illustrated in Figure 3.22a for Case Study 1 with $W_{L,5} = 0.1$ mol/s. Stage 5, directly above the feed, goes dry at $R_{\text{cr}} \approx 0.00446$ and the solution curve cannot proceed any further, with a 0 - 1 - 1 bifurcation observed at R_{cr}. On the other hand, if we specify a liquid side product flow rate $W_{L,2} = 0.1$ mol/s in Stage 2, the drying process of Stages 3 and 4 below is allowed to happen in the original degenerate fashion, as depicted in Figure 3.22b. This originates infinitely many solutions at $R_{\text{cr}} = 0.00445$; however, the solution curve stops once the outlet liquid L_2 reaches zero, and we obtain a 0 - /∞ - 1 bifurcation at R_{cr}.

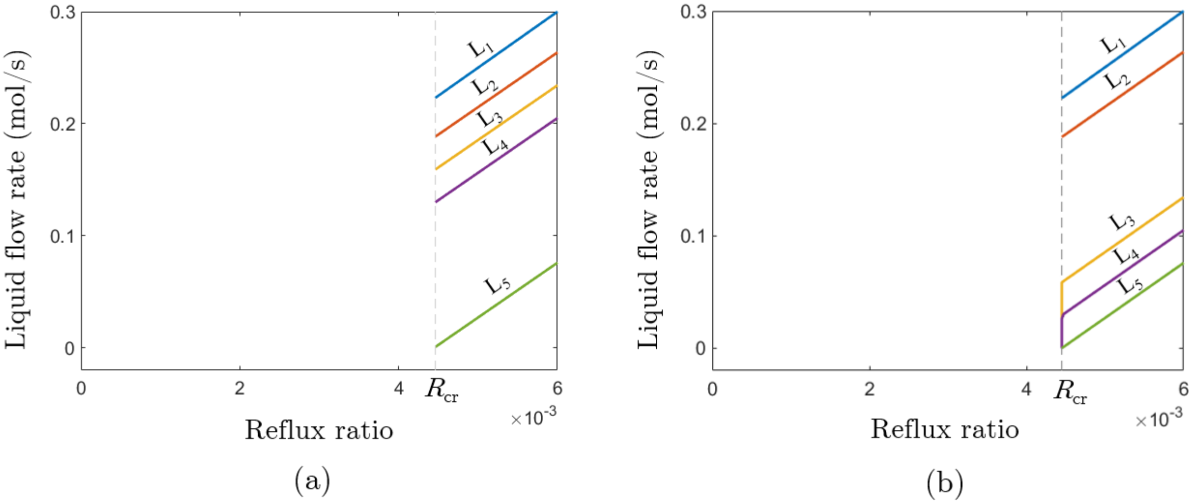
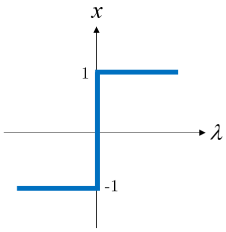
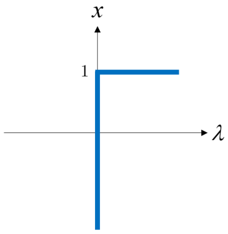
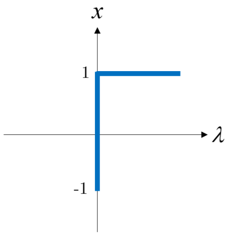
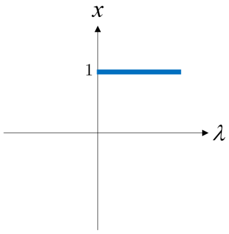


Figure 3.22: Above-feed liquid flow rates versus R for Case Study 1 with (a) $W_{L,5} = 0.1$ mol/s, (b) $W_{L,2} = 0.1$ mol/s.

3.9. Summary of bifurcations

Table 3.1 presents all codimension-1 bifurcations introduced in this chapter in their simplest or normal form, which corresponds to the bifurcation at $\lambda = 0$ of a simple single-equation, single-variable system $f(x, \lambda) = 0$ with the same essential behavior. In all instances, the overall solution set is a connected 1-dimensional PC^r manifold, with or without boundary; moreover, the three first bifurcations depicted in Table 3.1 are degenerate. Finally, we draw attention to the possibility that other combinations of column specifications not considered in this chapter could give rise to other types of novel, nonsmooth bifurcations in dry/vaporless distillation columns.

Table 3.1: Summary of codimension-1 bifurcations.

Bifurcation	**1 - ∞ - 1**	**0 - ∞ - 1**	**0 - /∞ - 1**	**0 - 1 - 1**
Normal form $f(x,\lambda)=0$	 $\mathrm{mid}(x+1,-\lambda,x-1)=0$	 $\max(x-1,-\lambda)=0$	 $\min(x+1,-\lvert\max(x-1,-\lambda)\rvert)=0$	 $\min(\lambda,-\lvert x-1\rvert)=0$
Solution set around $\boldsymbol{\lambda}=\mathbf{0}$	1-dimensional PC^r manifold	1-dimensional PC^r manifold	1-dimensional PC^r manifold w/ 1 boundary point	1-dimensional PC^r manifold w/ 1 boundary point
Solution set at $\boldsymbol{\lambda}=\mathbf{0}$	1-dimensional PC^r manifold w/ 2 boundary points	1-dimensional PC^r manifold w/ 1 boundary point	1-dimensional PC^r manifold w/ 2 boundary points	A single point (0-dimensional PC^r manifold)
Examples	- Figures 3.8, 3.10 at $R=R_{\text{cr}}$; - Figure 3.13b at $B=B_{\text{cr}}$; - Figure 3.21a at $R=R_{\text{cr}}$ and $R=R^*$.	- Figure 3.14 at $B=0$; - Figure 3.20 at $R=0$.	- Figures 3.11, 3.12 at $B=B_{\text{cr}}$; - Figure 3.22b at $R=R_{\text{cr}}$.	- Figures 3.8, 3.10 at $R=0$; - Figure 3.22a at $R=R_{\text{cr}}$.

3.10. Conclusions

We have presented a MESH-based steady-state model for multistage distillation, consisting of a system of nonsmooth equations, that can simulate columns operating with dry and/or vaporless stages. This task cannot be achieved with commercial software such as Aspen Plus, which have well-known dry column simulation failures, since the describing equations for each stage need to be automatically switched according to the phases present at the solution. The only other competing modeling strategy in the literature relies on complementarity constraints, which require several equation solving tasks or the use of optimization algorithms. On the other hand, our model can be solved in a single equation-solving task using automatically-computed generalized derivatives, and can bound flow rates and other variables to be non-negative. Moreover, the algebraic nature of our model has allowed us to develop the necessary continuation methods to reveal, for the first time, the occurrence of an infinite number of steady states in distillation columns with dry and/or vaporless stages.

In two case studies involving dry/vaporless distillation columns from the pertinent literature, we have observed four novel types of codimension-1 bifurcations, classified according to the change in the number of steady states with respect to a single input parameter: 1 - ∞ - 1, 0 - ∞ - 1 and 0 - /∞ - 1 (degenerate), and 0 - 1 - 1 (non-degenerate). By further analyzing several types of column configurations, we demonstrate that degenerate bifurcations occur at critical input parameter values in a general context, regardless of the mixture and thermodynamic models, as long as there is either a pressure gradient in the column or imposed heat transfer in the trays. The degeneracy persists even when non-ideal stage efficiencies of different types are specified and when multiple feed streams and side products are included. We have also found that a codimension-2 bifurcation with a higher degree of degeneracy can occur when two input parameters are varied simultaneously. All presented bifurcations exhibit nonsmooth behavior, mathematically described by the proposed concept of piecewise-smooth manifolds.

Our findings further demonstrate that the input parameter values leading to dry/vaporless stages in a distillation column are not necessarily small, cannot be predicted prior to simulation, and often give rise to an infinite number of steady states. This degeneracy is associated with singular generalized derivatives; therefore, it requires special continuation methods that are not currently implemented in general process flowsheeting software. Moreover, in some cases no feasible solutions exist below the critical parameter values. In Chapter 4 we will focus on developing alternative nonsmooth formulations to address both the singularity and infeasibility limitations, in order to create a distillation model that is robust to the issues associated with dry/vaporless stages and applicable for flowsheet simulation in industrial practice.

Chapter 4

Nonsmooth distillation models robust to dry column errors and infeasible specifications

In this chapter we overcome convergence errors in distillation simulation related both to dry stages and to infeasible specifications with our so-called nonsmooth adaptive models, which can be of two main types: "single-soft" and "double-soft". With the single-soft adaptive model we reset one user-chosen specification, if it happens to be infeasible, by bringing one of the liquid or vapor flow rates L_j, V_j within the column to one of its imposed lower or upper bounds. This is achieved by enforcing the original MESH equations together with a single nonsmooth specification, in a single equation-solving task.

However, we may encounter input conditions in which two specifications are infeasible simultaneously. For such cases we have developed the double-soft adaptive model, which relaxes two specifications and replaces them with nonsmooth equations that enforce upper and lower bounds on the flow rates, though not necessarily in a strict sense. Despite the fact that the obtained solution is not guaranteed to be the "nearest" feasible one, this second modeling strategy can effectively allow us to proceed through infeasible flowsheet iterations and ensure the overall convergence of processes involving distillation columns.

4.1. Introduction

As discussed in Chapter 3, the standard MESH model can only converge to a physically valid solution when the column specifications lead to a steady state with vapor-liquid equilibrium conditions in all stages. That is, according to the nomenclature introduced in Section 3.2.1, all column stages must be operating in either Phase Regime I (vapor and liquid) or Phase Regimes II (dew-point vapor or bubble-point liquid). In Figure 4.1a we illustrate this behavior by plotting the type of MESH model solution observed for each pair of reflux ratio and feed temperature specifications for the benzene-toluene column previously studied in Section 3.7.1 (Column 1 in Table 4.2). Figure 4.1b shows how select liquid flow rates in the column change when we vary only the reflux ratio, keeping the feed temperature fixed at its bubble-point. For specifications in the interior of the vapor-liquid region in Figure 4.1a, all stages operate in Phase Regime I. At the boundary of the vapor-liquid region, at least one stage is dry/vaporless but still operates in Phase Regime II. Specifications on the exterior of the vapor-liquid region are infeasible because they lead to a mathematical solution in which one or more flow rates are negative. We continue to observe these non-physical MESH solutions as we move our specifications further away from the vapor-liquid region, until eventually the mole fractions may reach such aberrant values that the thermodynamic equilibrium equations might cease to be well-defined.

Due to its ability to switch between different equations depending on the phase regime of each stage, the nonsmooth MESH model developed in Chapter 3, whose behavior is illustrated in Figure 4.2, can describe distillation columns more thoroughly than the standard MESH model. The two models agree with each other only within the interior of the vapor-liquid region of Figure 4.2a. In the exterior of said region the nonsmooth MESH model has either a feasible solution where one or more stages are dry and in Phase Regime III (i.e., superheated vapor) or no mathematical solution, in which case we conclude that

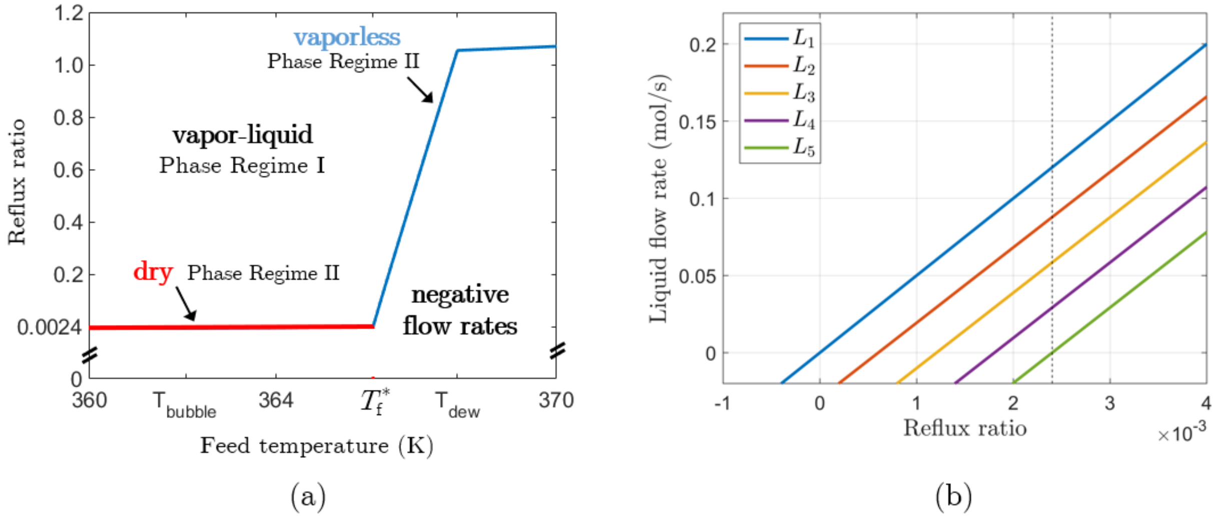
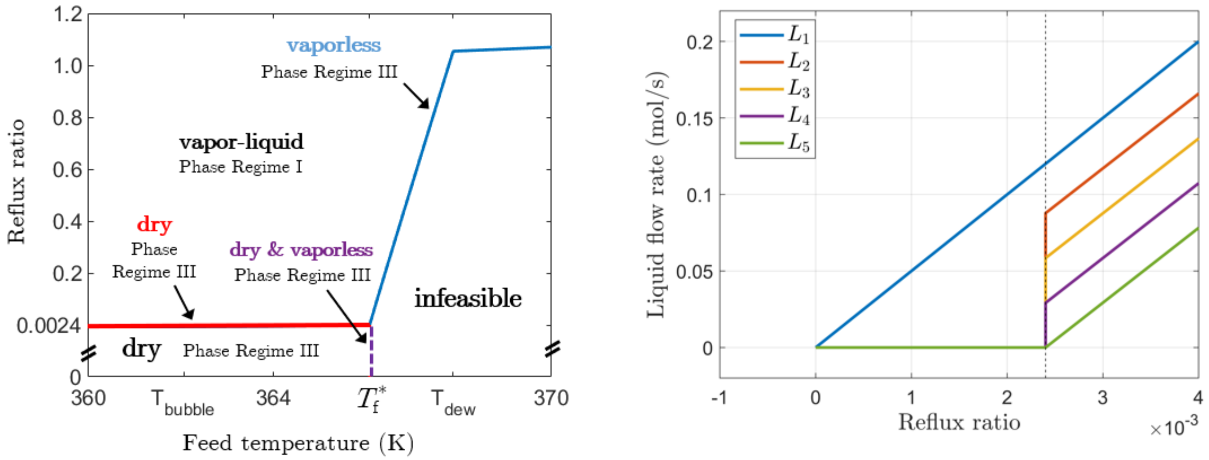


Figure 4.1: (a) Type of MESH solution for each reflux ratio and feed temperature; (b) liquid flow rates versus reflux ratio for the bubble-point feed temperature.

the specifications are truly infeasible. This conclusion is warranted based on the detailed parametric continuations performed in Chapter 3 and the nonsmooth MESH model's ability to bound flow rates to be non-negative. At the boundaries between regions in Figure 4.2a the model exhibits infinitely many solutions with dry and/or vaporless stages in Phase Regimes III, as illustrated by the bifurcation with degenerate vertical segments in Figure 4.2b.

With the nonsmooth MESH model we have demonstrated that dry/vaporless solutions in Phase Regimes III can indeed satisfy all the physical laws behind the MESH paradigm, thus revealing a wider range of feasible specification values in Figure 4.2a compared to Figure 4.1a. On the other hand, the nonsmooth MESH model cannot avoid dry column convergence errors for infeasible specifications. Another limitation is that, for specifications at the boundaries between regions in Figure 4.2, it is not clear which of the infinitely many solutions should be chosen as the column simulation output. Further, the model's generalized derivatives are singular at said boundaries, which hinders the performance of standard Newton-type equation solving methods.



(a)

(b)

Figure 4.2: (a) Type of nonsmooth MESH model solution for each reflux ratio and feed temperature; (b) liquid flow rates versus reflux ratio for the bubble-point feed temperature.

4.1.1. Aspen Plus' RadFrac model

Column specifications

The RadFrac model in Aspen Plus for rigorous distillation simulation with equilibrium stages employs the MESH equations, albeit the standard algorithm used to converge them is based on the inside-out method of Boston and Sullivan Jr [14] [15] (see Chapter 5 for a more detailed discussion). The exact structure of the latter as currently implemented in RadFrac is not described in the literature. As reported by Russell [75], a main modification from the formulation in [14] was the addition of a middle loop together with the original outer and inner loops.

To fix the two main degrees of freedom within the "Setup" section in RadFrac, we are only allowed to choose values for two of the following "directly-specifiable" variables: reflux ratio, reflux rate, boilup ratio, boilup rate, distillate rate, distillate-to-feed ratio, bottoms rate, bottoms-to-feed ratio, condenser duty and reboiler duty. This restriction is due to the specific (and efficient) structure of the inner loop of Boston and Sullivan Jr, as opposed to the inner loop of Russell [75] which can enforce any specification directly. In RadFrac, more "complicated" specifications such as product purities must be enforced indirectly through the “Design Specifications” section. The middle loop is responsible for converging these by varying the directly-specifiable variables that are used within the inner loop.

Dry column errors

With RadFrac we cannot obtain negative-flow-rate MESH solutions such as the ones illustrated in Figure 4.1 because the software throws a “dry column” severe error message when either

(a) a flow rate L_j, V_j reaches or goes below a threshold value of $10^{-5}F_s$, where F_s is the sum of all feed flow rates to the column, or

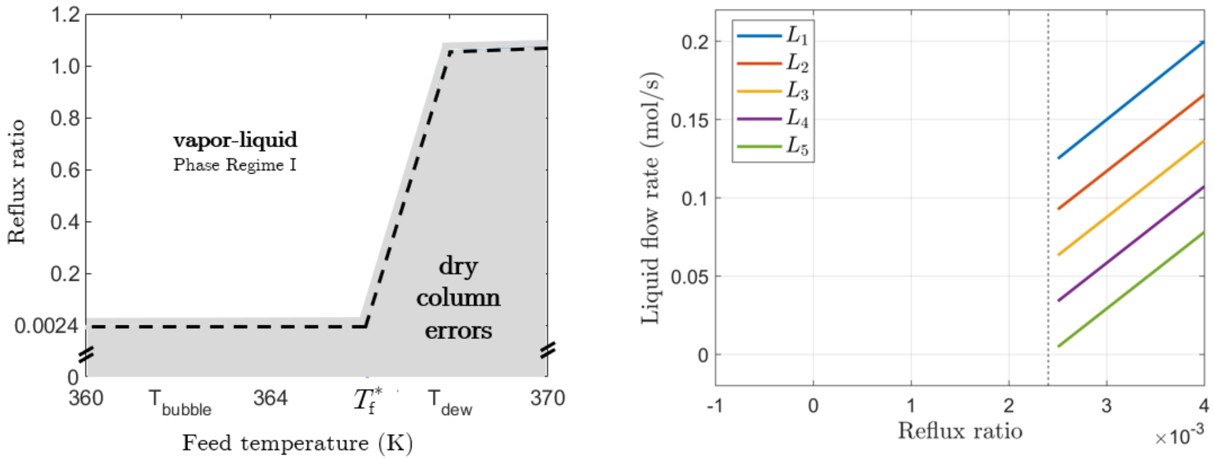
(b) a ratio V_j/L_j or L_j/V_j reaches or goes below a threshold value of 10^{-5}.

This error message causes the software to abort the simulation even within intermediate calculations, before the MESH equations could be potentially converged to a negative-flow-rate solution. The counterpart of Figure 4.1 for the RadFrac model is presented in Figure 4.3. Given the positive lower bound on flow rate values imposed by Aspen Plus, with RadFrac we cannot reach the dry/vaporless MESH solutions on the boundaries of Figure 4.1a, nor the vapor-liquid solutions directly above said boundary. Moreover, by comparing Figures 4.2 and 4.3, we can see that we obtain the same dry column error message in Aspen Plus both when there exists a feasible dry/vaporless solution in Phase Regimes III as well as when the specifications are truly infeasible.

General convergence errors

Dry column error messages in Aspen Plus provide the user with at least some insight into the mechanism behind RadFrac’s failure to converge. However, that is not the case for the other types of generic convergence error messages that are even more often encountered when using RadFrac. These messages include:

(a) Severe error: Fortran divide by zero encountered;



(a)

(b)

Figure 4.3: (a) Type of RadFrac solution for each reflux ratio and feed temperature; (b) liquid flow rates versus reflux ratio for the bubble-point feed temperature.

(b) Error: RadFrac not converged in 25 outside loop iterations;

(c) Error: (RadFrac) Block is not in mass balance.

(d) Severe error: Column not in mass balance. Check feeds, products, and column specifications.

Another possible error message when working with design specifications in RadFrac is:

(e) Warning: outside loop tolerance was satisfied but design spec iteration (middle) loop failed to converge. Reached an optimum value.

From the error message above, we assume that the termination criterion employed for RadFrac's middle loop is based on the sensitivity of the design specifications with respect to the directly specifiable variables reaching a near-zero value.

Both with the dry column and the more generic convergence error messages, Aspen Plus provides no explanation as to whether failure is due to numerical convergence difficulty or to infeasibility of the user-chosen specifications.

4.2. The proposed nonsmooth adaptive models

To overcome convergence errors related to dry stages and to infeasible specifications in a more general setting, we propose so-called nonsmooth adaptive models, which can be of two main types: “single-soft” and “double-soft”. Both of these models enforce the standard MESH equations, which are smooth, for each stage j:

$$L_{j-1} + V_{j+1} + F_j - (L_j + W_{L,j}) - (V_j + W_{V,j}) = 0, \tag{4.1}$$

$$x_{i,j-1}L_{j-1} + y_{i,j+1}V_{j+1} + z_{i,j}F_j - x_{i,j}(L_j + W_{L,j}) - y_{i,j}(V_j + W_{V,j}) = 0, \quad i = 1, \ldots, N_c, \tag{4.2}$$

$$h^L_{j-1}L_{j-1} + h^V_{j+1}V_{j+1} + h^F_j F_j - h^L_j(L_j + W_{L,j}) - h^V_j(V_j + W_{V,j}) + Q_j = 0, \tag{4.3}$$

$$y_{i,j} - K_{i,j}x_{i,j} = 0, \quad i = 1, \ldots, N_c, \tag{4.4}$$

$$\sum_{i=1}^{N_c} x_{i,j} - \sum_{i=1}^{N_c} y_{i,j} = 0, \tag{4.5}$$

where N is the number of stages, numbered from top to bottom, N_c is the number of components, V_j and L_j are the liquid and vapor molar flow rates leaving stage j, with the respective mole fractions $y_{i,j}$, $x_{i,j}$ of component i; F_j and $z_{i,j}$ are the molar flow rate and mole fractions of the feed stream to stage j; $W_{V,j}$ and $W_{L,j}$ are the flow rates of vapor and liquid side products withdrawn from the stage; h^V_j and h^L_j are the molar enthalpies of the outlet vapor and liquid phases, $K_{i,j}$ is the equilibrium ratio for component i, Q_j is the heat transfer rate to the stage, and F_s is the sum of all feed flow rates to the column. Some of the streams are absent in the first and last stages, as previously illustrated in Figure 3.3. The total distillate flow rate D is defined as $D = W_{L,1} + V_1$, the reflux ratio as $R = L_1/D$, and the boilup ratio as $B = V_N/L_N$. The vapor distillate fraction $\theta = V_1/(W_{L,1} + V_1)$, ranging from 0 to 1, is specified with an additional equation.

Equations 4.1-4.5 differ from the nonsmooth MESH model Equations 3.16-3.20 only in the summation Equation 4.5, which is no longer the median of three arguments. With this modification the adaptive models can still reach solutions with dry and vaporless stages operating in Phase Regimes II, but not in Phase Regimes III.

After specifying the feed stream conditions, stage pressures, heat duties Q_j and side product ratios $W_{L,j}/L_j, W_{V,j}/V_j$ or flow rates $W_{L,j}, W_{V,j}$ for intermediate stages $2 \leq j \leq N-1$, we still need to fix two degrees of freedom. Traditionally we would enforce specified values for two chosen column variables λ_1, λ_2 using the equations $\lambda_1 - \lambda_{1,\text{spec}} = 0$ and $\lambda_2 - \lambda_{2,\text{spec}} = 0$. The single-soft and double-soft models differ only in how they replace one or both of these two specification equations.

4.2.1. The single-soft adaptive model

The single-soft model replaces one "hard" specification equation $\lambda - \lambda_{\text{spec}} = 0$ for a user-chosen variable λ with the following "soft" nonsmooth specification equation:

$$\operatorname{mid}\left(-\frac{\min_j \{L_j, V_j\}}{F_s},\ \alpha\left(\lambda - \lambda_{\text{spec}}\right),\ \frac{1}{\beta}\left(1 - \frac{\max_j \{L_j, V_j\}}{r_{\max} F_s}\right)\right) = 0. \qquad (4.6)$$

Here,

- The first argument of the mid function is responsible for setting a lower bound for the flow rate values. The minimum is taken over the terms $L_1, \ldots, L_{N-1}, V_2, \ldots, V_N$ (the *internal* flow rates); over any side product flow rates; and over the terms $W_{L,1} - r_{\min} F_s$, $L_N - r_{\min} F_s$, and $V_1 - r_{\min} F_s$ if $\theta \neq 0$, unless one of these flow rates is being fixed by the user with a "hard" specification. If that is the case, we do not include the term corresponding to the specified flow rate in order to avoid structural singularity in the (generalized) derivative matrix of the model equations. Here, $0 \leq r_{\min} < 1$ is an arbitrarily chosen minimum ratio that can establish a positive lower bound $r_{\min} F_s$ for the external flow rates $W_{L,1}, L_N$, and also for V_1 when the condenser is partial; all other flow rates are bounded below by zero. A value of $r_{\min} > 0$ is desirable within flowsheet simulation to ensure that any equipment downstream of distillation columns has a non-zero feed. In our flowsheet simulations in Section 5.6 we have used $r_{\text{min}} = 0.05$, while in the single-column case studies of this chapter we set $r_{\text{min}} = 0$.

- $\alpha = \pm 1$ is determined from the column input conditions and reflects in which direction λ affects the flow rate values inside the column, as further explained below.
- In the second argument of the mid function, it might be necessary to include a scaling factor to ensure values remain approximately between 0 and 1. For example, for $\lambda = L_1$ or V_N we should include F_s in the denominator of the second argument.
- $\beta = 15$ is a scaling factor that we found to be adequate for our set of test cases. Although equation scaling does not have any influence on the Newton step for a smooth system, differences in scaling between the arguments of a nonsmooth function can change which argument is active and thus alter the semismooth Newton step. While the first argument in Equation 4.6 is designed to have magnitude between 0 and 1, we found that the third argument can reach much more elevated magnitudes during iterations and preclude convergence without an adequate value of β.
- The third argument of the mid function is responsible for setting the upper bound $r_{\text{max}} F_s$ on flow rate values, where $r_{\text{max}} > 1$ is an arbitrarily chosen maximum ratio; in our simulations we have used $r_{\text{max}} = 5$. The maximum only needs to be taken over the internal flow rates $L_1, \ldots, L_{N-1}$, $V_2, \ldots, V_N$ and not over the external product flow rates $W_{L,1}, L_N$ and V_1. This is the case because only the former can grow unbounded near an infinite flow rate discontinuity when using the standard MESH model, as discussed in Section 4.5.

Equation 4.6 is suitable for specified variables λ that (tend to) have a monotonic relationship with L_j, V_j values, such as the reflux ratio R, boilup ratio B, and product purities $x_{1,i},\ x_{N,i}$. We might need to use different formats for the nonsmooth specification equation when using other types of soft specified variables λ, which will be outside of the scope of this thesis.

The need for imposing a lower bound on L_j, V_j is clear from our previous discussion on dry column errors, and 0 is the natural value to choose (at least for the internal flow rates). On the other hand, we must impose an upper bound due to the infinite asymptoptic discontinuity in flow rate values that MESH solutions exhibit at azeotrope pinch points and also when $N < N_{\text{min}}$ for the desired separation, as will be presented in Section 4.5. In this case there is no natural upper bound value to choose, so we must select an arbitrary finite value $r_{\text{max}}F_s$.

We define $\lambda_{r_{\text{min}}}$ as the value of λ that leads to the "minimum flow rate" (MESH) solution in which the first argument of the mid function in Equation 4.6 is equal to zero. That corresponds to the minimum flow rate in the column reaching its imposed lower bound, i.e., either an internal flow rate L_j, V_j or side-product flow rate $W_{L,j}, W_{V,j}$ is equal to zero, or an external product flow rate $W_{L,1}, L_N$ (or V_1 if $\theta \neq 0$) is equal to $r_{\text{min}}F_s$. Similarly, $\lambda_{r_{\text{max}}}$ is the λ value that corresponds to the "maximum flow rate" (MESH) solution for which the third argument of the mid function is equal to zero. That is, the maximum internal flow rate L_j, V_j is equal to its upper bound $r_{\text{max}}F_s$. For a given distillation column and choice of λ, neither $\lambda_{r_{\text{min}}}$ nor $\lambda_{r_{\text{max}}}$ can be predicted prior to simulation.

In Equation 4.6, the specification $\lambda - \lambda_{\text{spec}} = 0$ is only enforced if λ_{spec} gives rise to a MESH solution that satisfies both the upper and lower constraints on all flow rates. Otherwise, λ is automatically reset to either $\lambda_{r_{\text{min}}}$ or $\lambda_{r_{\text{max}}}$, depending on our choice of $\alpha = \pm 1$, to yield either the maximum or lower flow rate MESH solution:

- For $\alpha = 1$, the single-soft model returns the maximum flow rate solution $\lambda = \lambda_{r_{\text{max}}}$ when $\lambda_{\text{spec}} \leq \lambda_{r_{\text{max}}}$, and the minimum flow rate solution $\lambda = \lambda_{r_{\text{min}}}$ when $\lambda_{\text{spec}} \geq \lambda_{r_{\text{min}}}$.
- For $\alpha = -1$, the single-soft model returns the minimum flow rate solution $\lambda = \lambda_{r_{\text{min}}}$

when $\lambda_{\text{spec}} \leq \lambda_{r_{\min}}$, and the maximum flow rate solution $\lambda = \lambda_{r_{\max}}$ when $\lambda_{\text{spec}} \geq \lambda_{r_{\max}}$.

In general, we should use $\alpha = -1$ if higher values of λ tend to increase L_j, V_j values (e.g., for $\lambda = R$), since in this case we expect to have $\lambda_{r_{\max}} > \lambda_{r_{\min}}$. Conversely, $\alpha = 1$ is the most adequate choice if higher values of λ tend to decrease flow rate values, given that we should have $\lambda_{r_{\max}} > \lambda_{r_{\min}}$. Table 4.1 presents the strategies for determining α from the column input conditions that we have used in the case studies of this chapter and of Chapter 5. We note that, since our single-soft model is designed to work even with extremely unreasonable soft specification values (e.g., negative compositions), we refrain from using the latter to determine α. However, we do make use of the hard specification value chosen by the user when both specifications are product purities.

Table 4.1: α values for each soft specified variable λ.

Specified variable λ	**α value**
Reflux ratio, $R = L_1/D$	-1
Boilup ratio, $B = V_N/L_N$	-1
Distillate purity, $x_{1,i}$	1, if i is heavy ($x_{j,i}$ increases with j); -1, if i is light ($x_{j,i}$ decreases with j). When x_N, i is the hard specification, we can use $\alpha = \text{sign}\left(x_{N,i,\text{spec}} - z_{c,i}\right)$.
Bottoms purity, $x_{N,i}$	-1, if i is heavy ($x_{j,i}$ increases with j); 1, if i is light ($x_{j,i}$ decreases with j). When x_1, i is the hard specification, we can use $\alpha = \text{sign}\left(x_{1,i,\text{spec}} - z_{c,i}\right)$.

The single-soft adaptive model directly enforces a specification equation for any variable λ (including in its inside-out formulation that will be presented in Chapter 5), while RadFrac can only specify "complicated" variables indirectly through its middle loop, as discussed in Section 4.1.1. Further, the single-soft model's strategy to arrive at a substitute value of λ when λ_{spec} is infeasible stands in contrast to RadFrac's middle-loop optimality criterion. The latter is based on the sensitivity of the actual specification λ (e.g., product purity) with respect to the variable that is enforced in the inner loop (e.g., R). On the other hand, when λ leads or tends to lead to a solution in which one or more flow rates are outside of their imposed bounds, our model returns the "nearest best" λ value by bringing the furthest-deviating flow rate in the column back to its corresponding bound.

4.2.2. The double-soft adaptive model

In the double-soft model we relax the two "hard" specification equations $\lambda_1 - \lambda_{1,\text{spec}} = 0$ and $\lambda_2 - \lambda_{2,\text{spec}} = 0$ and replace them with

$$\min\left(0, \frac{\min_j \{L_j, V_j\}}{F_s}\right) = 0, \tag{4.7}$$

$$\min\left(0, 1 - \frac{\max_j \{L_j, V_j\}}{r_{\max} F_s}\right) = 0, \tag{4.8}$$

where the minimum and the maximum terms are taken in the same way as in Equation 4.6.

The above equations ensure 0 (or $r_{\min} F_s$) $\leq L_j, V_j \leq r_{\max} F_s$ without necessarily enforcing either of the bounds strictly, in contrast to the single-soft model. Because of that, the double-soft adaptive model can also potentially avoid other types of convergence errors unrelated to the flow rate values going out of bounds. Due to the identically zero terms, Equations 4.7 and 4.8 may exhibit inherently singular generalized derivatives. Nevertheless, we have found that the double-soft model can be solved successfully by using the pseudo-inverse of the generalized derivatives within the semismooth Newton method (see Section 4.4), a strategy that had not been attempted before in nonsmooth process modeling.

Every solution that the double-soft model converges to is MESH feasible, given that all first-principles physical laws are still enforced within the model equations. However, since any specification equations $\lambda_1 - \lambda_{1,\text{spec}} = 0$ and $\lambda_2 - \lambda_{2,\text{spec}} = 0$ we might wish to try to enforce are not included in the model, in general we do not obtain $\lambda_1 = \lambda_{1,\text{spec}}$ and/or $\lambda_2 = \lambda_{2,\text{spec}}$ even when both specified values are feasible or when only a single one of them is infeasible. In the latter cases, if we construct our initial guess based on $\lambda_{1,\text{spec}}, \lambda_{2,\text{spec}}$ we might still converge to a solution for which $\lambda_1 \approx \lambda_{1,\text{spec}}$ and/or $\lambda_2 \approx \lambda_{2,\text{spec}}$. Despite its "looseness", we have found that this modeling strategy can effectively allow us to proceed through infeasible iterations of flowsheets with distillation columns, as demonstrated in the case studies of Chapter 5.

4.3. Initialization procedures

Column initialization depends on which variables are specified to fix the two main degrees of freedom, and the standard strategy is to use the user-chosen values for the latter to compute the other column variables. The best case scenario is when these allow us to estimate all L_j, V_j values directly, which is the case for D or L_N and R or B specifications. When at least one of the specifications is a product purity, for example, we must instead guess values for D, L_N or R, B. Since for our adaptive models we cannot assume that user-chosen specifications will lead to a feasible initial guess or even be physically valid themselves, we then correct the initialized variables to make sure they are within their physical bounds. Further, for a top or bottom product purity soft specification, we have found it advantageous to screen for the appearance of an azeotrope at Stage 2 or Stage N, respectively, in order to get better initial estimates for compositions and flow rates (see Section 4.3.4).

Our procedures to generate an initial guess for the column variables, considering each type of pair of specifications used in our case studies, are presented below. Flashing the composite feed stream in Procedure 4.3.1 is standard for initializing temperatures and compositions (e.g., see Chapter 10 in [79]); the main difference is that we used one extra flash calculation at the top and bottom stages. In Procedure 4.3.2, assuming constant molar overflow for initializing flow rates is also standard. However, we specifically reset L_j, V_j values to reflect the imposed upper and lower bounds in our adaptive models.

4.3.1. D or L_N and R or B specifications

1. Initialize $\mathbf{T}$,$\mathbf{x}$,$\mathbf{y}$ values according to Procedure 4.3.1.

2. Initialize $\mathbf{L}$,$\mathbf{V}$ values according to Procedure 4.3.2 with the user-specified values for D or L_N, and R or B.

4.3.2. $x_{1,i}$ and $x_{N,i}$ specifications for a binary mixture

The procedure below applies to the single-soft model, i.e., when either $x_{1,i}$ or $x_{N,i}$ is a soft specification.

1. Set $x_{1,i}, x_{N,i}$ equal to their specified values. If the soft composition is greater than 1 or smaller than 0, reset it to 0.99 or 0.01, respectively.

2. Screen for an azeotrope at the top or bottom of the column according to Section 4.3.4 for the soft composition $x_{1,i}$ or $x_{N,i}$, respectively. If an azeotrope of composition $\mathbf{x}_{\text{azeo}}$ is found that does not correspond to a pure component, check if the soft specification value goes beyond $x_{\text{azeo},i}$ when compared to $z_{c,i}$, the composite feed mole fraction of i. If so, reset the soft composition value (but not its specified value) to be approximately equal to $x_{\text{azeo},i}$, within a safety distance of 0.02.

3. If $z_{c,i}$ does not lie in the interval between $x_{1,i}$ and $x_{N,i}$, reset the soft purity so that $z_{c,i}$ belongs to said interval within a safety distance of 0.05.

4. Compute D from the total mass balance

$$D = F_s \frac{z_{c,i} - x_{N,i}}{x_{1,i} - x_{N,i}}. \tag{4.9}$$

5. Estimate $R = 0.1$ if the soft purity specification was reset in Step 3, otherwise guess $R = 1$.

6. Initialize $\mathbf{L}$,$\mathbf{V}$ values according to Procedure 4.3.2 using the D, R estimates.

7. If the soft specification $x_{1,i}$ or $x_{N,i}$ was reset in Step 2, we expect the adaptive model to converge to a near infinite-flow-rate solution. Therefore, in this case reset the internal flow rate L_j, V_j of maximum magnitude to a near-maximum value of $0.8\, r_{\max} F_s$ and increment all other internal flow rates in the same amount.

8. Set values for $x_{1,k}$ and $x_{N,k}$ for $k \neq i$ so that $\sum_k x_{1,k} = \sum_k x_{N,k} = 1$.

9. Obtain values for $\mathbf{y}_1$ and T_1 by performing a bubble-point flash at $P = P_1$ using $\mathbf{z} = \mathbf{x}_1$.

10. Obtain values for $\mathbf{y}_N$ and T_N by performing a dew-point flash at $P = P_N$ using $\mathbf{z} = \mathbf{x}_N$.

11. Calculate remaining $\mathbf{T}, \mathbf{x}, \mathbf{y}$ values with linear interpolation between Stages 1 and N.

4.3.3. Other specifications

For other pairs of specifications, and for purity specifications within the double-soft model, we can perform the following generic initialization:

1. Initialize $\mathbf{T}$,$\mathbf{x}$,$\mathbf{y}$ values according to Procedure 4.3.1.

2. Initialize $\mathbf{L}$,$\mathbf{V}$ values according to Procedure 4.3.2 with $D = 0.5F_s$ (if D not specified) and $R = 1$ (if R not specified).

Procedure 4.3.1 (Initialization of $\mathbf{T}, \mathbf{x}, \mathbf{y}$ values)**.**

Given compositions and flow rates of feed streams, and stage pressures:

1. Combine all column feed streams to form a composite feed with composition $\mathbf{z}_c$ and flow rate F_s.

2. Set T_1 and T_N as the bubble-point and dew-point temperatures of the composite feed, respectively, at pressure $P_m = (P_1 + P_N)/2$.

3. Perform a PT-flash of the composite feed at $P = P_m$ and $T = (T_1 + T_N)/2$ and obtain liquid and vapor mole fractions $\mathbf{x}_f, \mathbf{y}_f$.

4. Perform a bubble-point flash at $P = P_2$ using $\mathbf{z} = \mathbf{y}_f$ and set $\mathbf{y}_2$ equal to the resulting bubble-point vapor composition.

5. For simplicity, assume a total condenser and set $\mathbf{x}_1 = \mathbf{y}_2$. To guess $\mathbf{y}_1$, perform a bubble-point flash at $P = P_1$ using $\mathbf{z} = \mathbf{x}_1$.

6. Perform a dew-point flash at $P = P_{N-1}$ using $\mathbf{z} = \mathbf{x}_f$ and set $\mathbf{x}_N$ equal to the resulting dew-point liquid composition.

7. To guess $\mathbf{y}_N$, perform a dew-point flash at $P = P_N$ using $\mathbf{z} = \mathbf{x}_N$.

8. Obtain the remaining $\mathbf{T}, \mathbf{x}, \mathbf{y}$ values via via linear interpolation.

Procedure 4.3.2 (Initialization of $\mathbf{L}$, $\mathbf{V}$ values)**.**

Given values for D or L_N, and R or B:

1. Reset the given D or L_N value so that $r_{\min} F_s \leq D, L_N \leq (1 - r_{\min})F_s$, then use the relationship $F_s = D + L_N$ to obtain L_N or D.

2. Set $V_1 = \theta D$ and $W_{L,1} = D - V_1$.

3. Initialize any side product flow rates $W_{L,j}, W_{V,j}$ at their specified values. If withdrawal ratios are specified instead, initialize $W_{L,j}, W_{V,j}$ values at zero for simplicity.

4. If R is specified, set $L_1 = R_{\text{spec}} D$ and $V_2 = L_1 + D$, then perform constant molar overflow (CMO) calculations down the column to obtain values for $L_2, \ldots, L_{N-1}$ and $V_2, \ldots, V_N$. If B is specified instead, set $V_N = B_{\text{spec}} L_N$ and $L_{N-1} = V_N + L_N - F_N$, then perform CMO calculations up the column to obtain values for $L_{N-2}, \ldots, L_1$ and $V_{N-1}, \ldots, V_2$.

5. If the internal L_j, V_j flow rate of smallest value is smaller than $r_{\min} F_s$, reset it to that value. Then, increase all other internal flow rates in the same amount so that CMO is still satisfied.

6. If the internal L_j, V_j flow rate of maximum value is larger than $0.8 r_{\max} F_s$, reset it to that value. Then, decrease all other internal flow rates in the same amount so that CMO is still satisfied.

4.3.4. Screening for an azeotrope

To screen for a non-trivial azeotrope at the top of the column, for example, the naive approach would be to solve the following system of equations for $\mathbf{x}$ and T at the top tray pressure P_2:

$$x_i = K_i(T, P_2, \mathbf{x}, \mathbf{x}), \quad i = 1, \ldots, N_c, \tag{4.10}$$

$$\sum_{i=1}^{N_c} x_i = 1. \tag{4.11}$$

However, this system of equations has at least N_c trivial solutions corresponding to the pure components. As with any nonlinear system with multiple solutions, we are not guaranteed to converge to the desired one without performing continuation procedures.

Instead, our approach to screen for an azeotrope at the top of the column is to solve a fixed-point problem $\mathbf{w} = \mathbf{f}(\mathbf{w})$, where $\mathbf{f}(\mathbf{w})$ is the vapor composition that results from solving a bubble-point flash at $P = P_2$ using $\mathbf{w}$ as the flash feed composition. Solving $\mathbf{w} = \mathbf{f}(\mathbf{w})$ with direct substitution using $\mathbf{w} = \mathbf{z}_c$ as the initial guess corresponds to passing the composite feed through a sequence of (infinitely many) bubble-point flashes, where the bubble-point vapor composition output of each flash is fed as input to the next one. In the limit of the sequence of flashes, both the vapor and liquid outputs of the bubble-point flash have the same composition as the flash feed. This series of flash vessels mimics an infinite sequence of distillation trays in the upper column section, and in this case it must asymptotically reach the "nearest" azeotrope that is lighter than $\mathbf{z}_c$, albeit that might be a pure component. However, this is not always guaranteed to be the same azeotrope that will tend to be formed at the top of the column for the chosen process conditions.

To screen for an azeotrope at the bottom of the column we perform an analogous procedure, except that $\mathbf{f}(\mathbf{w})$ is now the liquid composition that results from solving a dew-point flash at $P = P_N$ using $\mathbf{w}$ as the flash feed composition.

4.4. The proposed equation-solving method

We solve our single-soft and double-soft adaptive models $\mathbf{f}(\mathbf{X}) = \mathbf{0}$, which are systems of piecewise-smooth (PC^∞) equations, with a modified version of the semismooth Newton method [69]. The two main modifications involve using the pseudoinverse semismooth Newton step (see Section 2.3.5) conditionally, in a "try/catch" approach, and performing a model-specific "forward tracking" line search. The algorithmic structure of the method for a generic iteration k is presented in Algorithm 1.

To generate the next iterate $\mathbf{X}^{k+1}$ from $\mathbf{X}^k$, first we obtain a B-subdifferential element $\mathbf{G}(\mathbf{X}^k) \in \partial^B \mathbf{f}(\mathbf{X}^k)$ exactly with the automatic differentiation algorithm of Khan and Barton [46] (see Section 2.1.3). To obtain the direction $\mathbf{d}^k$, first we attempt to compute

Algorithm 1: k-th iteration of the equation solving method.

```
Input  : $\mathbf{X}^k$, $\mathbf{f}(\mathbf{X}^k)$, $\theta > 1$, $\alpha_{\min}$ function.
Output: $\mathbf{X}^{k+1}$.

Obtain $\mathbf{G}(\mathbf{X}^k) \in \partial^B \mathbf{f}(\mathbf{X}^k)$ using the algorithm of Khan and Barton [46]
try:
    Solve $\mathbf{G}(\mathbf{X}^k)\mathbf{d} = -\mathbf{f}(\mathbf{X}^k)$ for $\mathbf{d}$
    $\mathbf{d}^k \leftarrow \mathbf{d}$
catch $\mathbf{G}(\mathbf{X}^k)$ singular or ill-conditioned:
    $\mathbf{d}^k \leftarrow -\mathbf{G}(\mathbf{X}^k)^\dagger \mathbf{f}(\mathbf{X}^k)$
end
$\alpha \leftarrow \alpha_{\min}(\mathbf{d}^k)$
$\mathbf{X}^{k+1} \leftarrow \mathbf{X}^k + \alpha \mathbf{X}^k$
if $\|\mathbf{f}(\mathbf{X}^{k+1})\| < \|\mathbf{f}(\mathbf{X}^k)\|$ and $\alpha < 0.9$ and $\mathbf{X}^{k+1}$ within physical bounds then
    $n_{\text{current}} \leftarrow \|\mathbf{f}(\mathbf{X}^k)\|$
    while $\|\mathbf{f}(\mathbf{X}^{k+1})\| < n_{current}$ and $\alpha \leq 1$ and $\mathbf{X}^{k+1}$ within physical bounds do
        $n_{\text{current}} \leftarrow \|\mathbf{f}(\mathbf{X}^{k+1})\|$
        $\alpha \leftarrow \theta\alpha$
        $\mathbf{X}^{k+1} \leftarrow \mathbf{X}^k + \alpha \mathbf{d}^k$
    end
    $\alpha \leftarrow \alpha/\theta$
    $\mathbf{X}^{k+1} \leftarrow \mathbf{X}^k + \alpha \mathbf{d}^k$
end
return $\mathbf{X}^{k+1}$
```

the standard semismooth Newton step, i.e., we try to solve

$$\mathbf{G}(\mathbf{X}^k)\mathbf{d} = -\mathbf{f}(\mathbf{X}^k) \tag{4.12}$$

for $\mathbf{d}$ (see Section 2.3.2). If this fails due to $\mathbf{G}(\mathbf{X}^k)$ being singular or ill-conditioned, then we compute the so-called pseudoinverse semismooth Newton step $\mathbf{d}^k = -\mathbf{G}(\mathbf{X}^k)^\dagger \mathbf{f}(\mathbf{X}^k)$ (see Section 2.3.5). This failure is expected in most iterations for the double-soft adaptive model, given the structurally singular nature of some of its selection functions, but it can also sometimes happen in intermediate iterations with the single-soft model.

Finally, we select a step size α in the direction $\mathbf{d}^k$. Traditionally, one would start with a generic and reasonably large step size $\alpha = \alpha_0$ and then progressively decrease it through a backtracking line search, so as to approximately minimize $\left\|\mathbf{f}(\mathbf{X}^k + \alpha\mathbf{d}^k)\right\|$. However, in the case of our adaptive models, there is no such generic "safe" step size α_0 that we could start from. The model function $\mathbf{f}$ can become very ill-conditioned during intermediate iterations, especially when the model solution is near an infinite flow rate pinch point (see Section 4.5). A high condition number $\mathbf{G}(\mathbf{X}^k)$ causes one or more components of the direction $\mathbf{d}^k$ to have extremely large magnitude, particularly the ones corresponding to column flow rates, such that an unwise step size might preclude convergence or lead $\mathbf{f}$ to become undefined. Therefore, our models require a so-called forward tracking type of line search, in which we start from a small enough step size and attempt to increase it to approximately minimize $\left\|\mathbf{f}(\mathbf{X}^k + \alpha\mathbf{d}^k)\right\|$. However, once again there is no generic, constant α_0 that is guaranteed to be small enough for ill-conditioned directions.

In our approach we compute the initial step size through a model-specific function of the direction, $\alpha_{\min}(\mathbf{d}^k)$, which limits how much the column variables are allowed to change in the direction $\mathbf{d}^k$. We impose "maximum" values ΔF and Δy for the magnitude in variation of flow rate values and mole fractions, respectively, such that $\alpha_{\min}(\mathbf{d}^k)$ corresponds to the maximum step size that satisfies these limitations. We then set $\alpha \leftarrow \alpha_{\min}(\mathbf{d}^k)$. As a result, the magnitude change of at least one variable corresponds to its threshold ΔF or Δy when going from $\mathbf{X}^k$ to $\mathbf{X}^k + \alpha\mathbf{d}^k$. In our case studies, we have found the values $\Delta F = 0.3F_s$ and $\Delta y = 0.1$ to be adequate.

If $\left\|\mathbf{f}(\mathbf{X}^k + \alpha\mathbf{d}^k)\right\| < \left\|\mathbf{f}(\mathbf{X}^k)\right\|$ we progressively increase the step size $\alpha \leftarrow \theta\alpha$, where $\theta > 1$, until the norm of $\mathbf{f}$ stops decreasing, and as long as $\alpha \leq 1$ and all flow rate and mole fractions variables in $\mathbf{X}^k + \alpha\mathbf{d}^k$ stay within their imposed physical bounds. In our simulations we have used $\theta = 1.4$. At last, we set $\mathbf{X}^{k+1} = \mathbf{X}^k + \alpha\mathbf{d}^k$ and repeat the procedure described in this section for iteration $k + 1$.

Both our single-soft and double-soft models consist of a square system of equations $\mathbf{f}(\mathbf{X}) = \mathbf{0}$; thus, any rank deficiency of the generalized derivative matrix $\mathbf{G}(\mathbf{X}^k)$ precludes it from being full column or row rank. Therefore, the conditions for the convergence theorems of the pseudoinverse semismooth Newton method presented in Section 2.3.5 (and also of the LP-Newton method presented in Section 2.3.3) cannot be verified to hold. Moreover, we can expect the double-soft model to have a 2-dimensional set of non-isolated solutions, since its equation system has two unspecified degrees of freedom. Nevertheless, we have found that both types of adaptive models can be successfully converged with the equation solving method described in this section.

4.5. Examples

Table 4.2 presents the distillation column examples that we will analyze in this chapter and/or in Chapter 5. Columns 1 and 7 correspond to Case Studies 1 and 2 of Chapter 3, respectively, and Column 2 is a slight modification of Column 1. We extracted Columns 3 and 4 from the ethanol-benzene pressure-swing distillation flowsheet of Example 11.5 in [79], which is schematized in Figure 5.10; this mixture forms a minimum-boiling azeotrope. Columns 5 and 6 are taken from the pressure-swing distillation flowsheet from Figure 3a of [92], as presented in Figure 5.17, which separates the maximum-boiling azeotropic system diethylamine-methanol. Both of these flowsheets will be simulated in full in Section 5.6 of Chapter 5.

We implemented our models and equation solving algorithms in MATLAB. Parameter values for all thermodynamic property correlations were obtained from the Aspen Plus V10 database. To validate our distillation model implementation, we compared our simulation results with those of Aspen Plus' Radfrac model using parameter values for which the latter converges without errors. Unless otherwise noted, we used $r_{\max} = 5$ and $r_{\min} = 0$ for all examples in this section. We simulate the single-soft model in Examples 1 through 5 with our (pseudoinverse) semismooth Newton method from Section 4.4, using standard and/or pseudo-arclength continuation procedures, when required, to plot the bifurcation diagrams. We will analyze convergence robustness of the single-soft model when using our "blind" initialization procedures from Section 4.3 in Chapter 5. In Example 6 we employ both the pseudoinverse semismooth Newton and the LP-Newton methods to simulate the double-soft model. In all examples, we imposed a maximum error tolerance of $\epsilon = 10^{-7}$ on the infinity norm of the residual of the single/double-soft model equations.

Table 4.2: Parameters and specifications for each example column.

	Column 1	**Column 2**	**Column 3**
Components	Benzene ($i = 1$) Toluene	Benzene ($i = 1$) Toluene	Ethanol ($i = 1$) Benzene
Liquid phase model	Ideal	Ideal	NRTL
Vapor phase model	Ideal	Ideal	Ideal
Number of stages	$N = 27$	$N = 5$	$N = 9$
Pressure profile	Linear between $P_1 = 1.05$ bar and $P_N = 1.2$ bar	Linear between $P_1 = 1.05$ bar and $P_N = 1.2$ bar	$P_1 = 0.26$ bar, linear between $P_2 = 0.3$ bar and $P_N = 0.4$ bar
Feed stream	$\mathbf{z} = (0.7, 0.3)$ Stage 6 $P_F = 1.013$ bar bubble-point	$\mathbf{z} = (0.7, 0.3)$ Stage 3 $P_F = 1.013$ bar bubble-point	$\mathbf{z} = (2/3, 1/3)$ Stage 6 $P_F = 1.013$ bar bubble-point
Hard specification	$D = 0.5F_s$	$x_{N,1} = 0.3$	$x_{N,1} = 0.99$
Soft specification	R	$x_{1,1}$	$x_{1,1}$
Relevant azeotrope	None	None	$\mathbf{x}_{\text{azeo}} = (0.3585, 0.6415)$

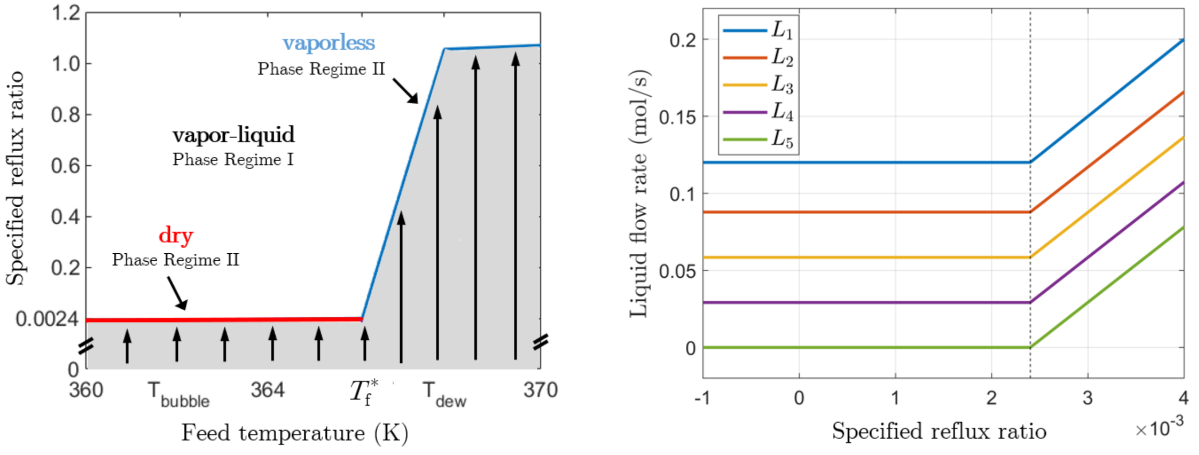
Table 4.2: Parameters and specifications for each example column (continued).

	Column 4	**Column 5**	**Column 6**	**Column 7**
Components	Ethanol ($i = 1$) Benzene	Diethylamine ($i = 1$) Methanol	Diethylamine ($i = 1$) Methanol	Methanol ($i = 1$) Acetone ($i = 2$) Methyl acetate Benzene Chloroform
Liquid phase model	NRTL	UNIQUAC	UNIQUAC	UNIQUAC
Vapor phase model	Ideal	Ideal	Ideal	Hayden O'Connell
Number of stages	$N = 5$	$N = 39$	$N = 39$	$N = 19$
Pressure profile	$P_1 = 1.013$ bar, linear between $P_2 = 1.06$ bar and $P_N = 1.2$ bar	Linear between $P_1 = 0.8$ atm and $P_N = 1.1$ atm	Linear between $P_1 = 10$ atm and $P_N = 10.3$ atm	Linear between $P_1 = 1.015$ bar and $P_N = 1.1$ bar
Feed stream	$\mathbf{z} = (0.37, 0.63)$ Stage 2 $P_F = 0.26$ bar bubble-point	$\mathbf{z} = (0.5, 0.5)$ Stage 16 $P_F = 1.013$ bar $T_F = 320$ K	$\mathbf{z} = (0.3, 0.7)$ Stage 19 $P_F = 1.1$ atm bubble-point	$\mathbf{z} = (0.15, 0.4, 0.05, 0.2, 0.2)$ Stage 7 $P_F = 1.013$ bar dew-point
Hard specification	$x_{N,1} = 0.01$	$x_{1,1} = 0.996$	$x_{1,1} = 0.004$	$D = 0.3F_s$
Soft specification	$x_{1,1}$	$x_{N,1}$	$x_{N,1}$	R or $x_{1,1}$
Relevant azeotrope	$\mathbf{x}_{\text{azeo}} = (0.449, 0.551)$	$\mathbf{x}_{\text{azeo}} = (0.285, 0.715)$	$\mathbf{x}_{\text{azeo}} = (0.579, 0.421)$	$\mathbf{x}_{\text{azeo}} = (0.272, 0.550, 0.156, 0.022, 0.00062)$

4.5.1. Example 1: benzene-toluene, soft R specification

In this section we consider Column 1 from Table 4.2, for which $D = 0.5F_s$ is a hard specification, $\lambda = R$ is the soft specification, and thus $\alpha = -1$ in Equation 4.6. The single-soft nonsmooth adaptive model yields the results in Figure 4.4, which stands in contrast to Figures 4.1, 4.2 and 4.3.

First we recall some key nomenclature from Section 3.7.1. As we decrease the reflux ratio R while keeping all other specifications constant, we eventually reach a critical value R_{cr} at which the first flow rate in the column becomes equal to zero. In Figure 4.2a, each feed temperature value corresponds to its own R_{cr} value. For each feed temperature the nonsmooth MESH model exhibits a continuum of solutions at R_{cr}, as seen in Figure 4.2b,



(a)

(b)

Figure 4.4: (a) Type of solution for each R and feed temperature using the single-soft adaptive model to reset R; (b) liquid flow rates versus R for the bubble-point feed temperature.

which range from the upper critical solution (corresponding to $R \to R_{\text{cr}}^{+}$) to the lower critical solution (corresponding to $R \to R_{\text{cr}}^{-}$). Among the whole set of solutions at R_{cr}, the upper critical solution is the only one that is MESH-feasible.

Using the nomenclature we just introduced in Section 4.2.1, R_{cr} corresponds to $R_{r_{\min}}$ and the upper critical solution corresponds to the minimum flow rate solution. Therefore, if $R_{\text{spec}} \leq R_{\text{cr}} = R_{r_{\min}}$ the single-soft adaptive model automatically resets $R = R_{r_{\min}}$ instead of enforcing $R = R_{\text{spec}}$ (hence the upward vertical arrows in Figure 4.4a), and returns the upper critical solution, as evidenced in Figure 4.4b. As a result, the degenerate bifurcation at R_{cr} is removed and the single-soft model exhibits a unique solution for each value of R. The model behaves in the same way when applied to all other case studies from Section 3.7, regardless of column configuration.

Figure 4.4b only addresses the behavior of the single-soft model for small values of R. On the other hand, Figure 4.5 shows how the normalized flow rate L_5/F_s (which is the flow rate that reaches zero at the upper critical solution) varies with both low and high values of R for a bubble-point liquid feed. We can see that the behavior of the single-soft model with respect to the soft specification is analogous to the phenomenon of sensor saturation. That is, the model output flatlines at the minimum flow rate solution if the input R_{spec} goes below the $R_{r_{\min}} \approx 0.0024$ value, and it flatlines at the maximum flow rate solution if $R_{\text{spec}} > R_{\text{r}_{\max}} \approx 8$. At the latter solution, L_5 stabilizes at a maximum value of $4F_s$, while the maximum flow rate in the column (which happens to be L_{20} in this example) is set to exactly $r_{\max}F_s = 5F_s$.

We can also conclude that the model exhibits three different modes of behavior with respect to R. This is precisely the desired outcome when using the mid function to create nonsmooth models; e.g., we can describe three different physical regimes for each stage of a distillation column by using the mid function within the nonsmooth MESH model of Chapter 3.

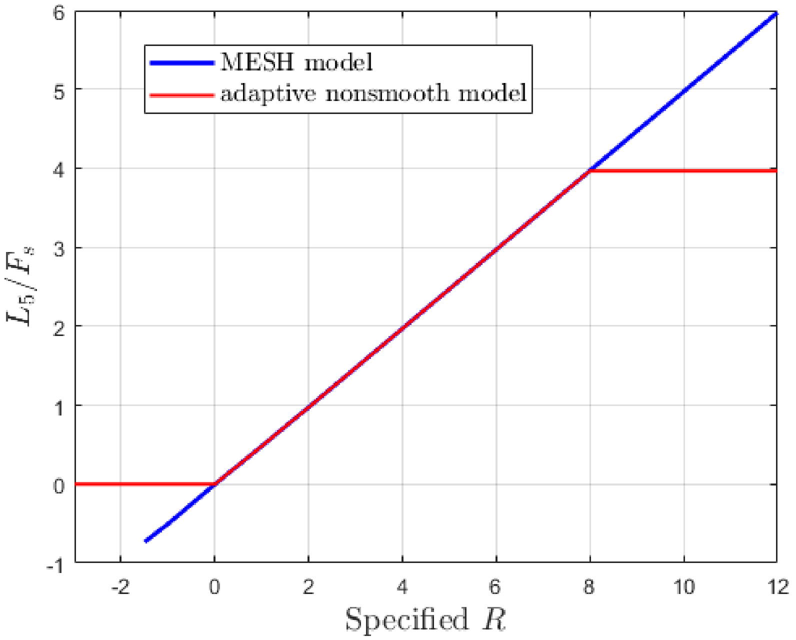


Figure 4.5: L_5/F_s versus R_{spec} for Example 1 using $\alpha = -1$ in Equation 4.6.

The specification $\lambda = R$ (and also $\lambda = B$) increases any internal flow rate in a monotonic, linear fashion, and thus $\alpha = -1$ is the correct sign for the second argument in Equation 4.6. Figure 4.6 shows the same plot from Figure 4.5 when we use $\alpha = 1$ instead of $\alpha = -1$. As explained in Section 4.2.1, reversing the α sign switches which solution the model returns when $R_{\text{spec}} \leq R_{r_{\min}}$ and when $R_{\text{spec}} \geq R_{r_{\max}}$, and in this case the specification-resetting strategy no longer makes intuitive sense. Further, though the intermediate feasible MESH solutions are preserved in the bifurcation diagram, we end up artificially introducing multiple steady states for feasible R values in Figure 4.6. As such, our ability to reach the intermediate feasible solutions numerically would be compromised.

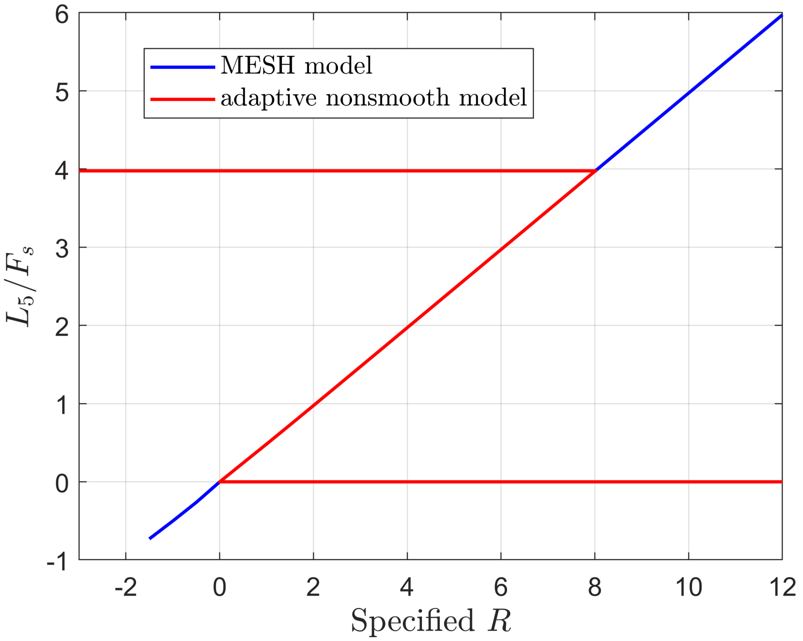


Figure 4.6: L_5/F_s versus R_{spec} for Example 1 using $\alpha = 1$ in Equation 4.6.

As will be further illustrated in Examples 2 through 5, we choose to impose an upper bound on flow rates values with the single-soft model mainly due to the infinite discontinuities observed when λ is a product purity specification. On the other hand, such behavior is not possible when $\lambda = R$ or B. In fact, due to the linear relationship depicted in Figure 4.5, we do not observe an infinite asymptote even as $R, B \to \infty$ and thus the system remains numerically well-conditioned. Therefore, for $\lambda = R$ or B we might choose not to bound the column flow rates above by substituting Equation 4.6 with

$$\min\left(\frac{\min_j \{L_j, V_j\}}{F_s},\ -\alpha\left(\lambda - \lambda_{\text{spec}}\right)\right) = 0, \tag{4.13}$$

which only enforces the lower bound on flow rate values.

4.5.2. Example 2: benzene-toluene, soft $x_{1,1}$ specification

We now consider the benzene-toluene Column 2 from Table 4.2, which is the same as Column 1 except that we set $N = 5$ and introduce the feed stream into Stage 3. Moreover, we use $x_{N,1} = 0.3$ as the hard specification, $\lambda = x_{1,1}$ as the soft specification, and $\alpha = \text{sign}\,(x_{N,1,\text{spec}} - z_1) = -1$ according to Table 4.1. Figure 4.7 presents the normalized flow rate L_2/F_s as a function of $x_{1,1,\text{spec}}$ for both the MESH and the single-soft adaptive models.

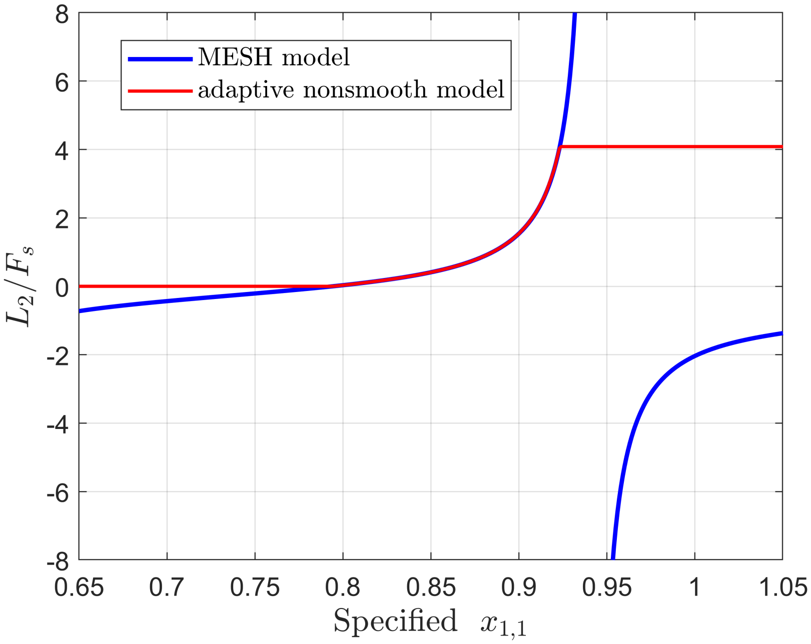


Figure 4.7: L_2/F_s versus $x_{1,1,\text{spec}}$ for Example 2.

When λ is a product purity specification, we have discovered that there is, in general, a narrow range of MESH-feasible λ values between the first flow rate reaching its lower bound at $\lambda_{r_{\min}}$ and an infinite asymptotic discontinuity in flow rate values at some $\lambda = \lambda_{\text{asym}}$. This discontinuity is observed regardless of the mixture being non-ideal or forming azeotropes. In Figure 4.7, we have $\lambda_{r_{\min}} \approx 0.8$ and $\lambda_{\text{asym}} \approx 0.94$. As expected, the curve of MESH solutions continues beyond $\lambda < \lambda_{r_{\min}}$ to form an infeasible branch with negative flow rates. At $\lambda_{\text{asym}} \approx 0.94$ we observe a two-sided asymptotic discontinuity that is characteristic of rational functions, with an infeasible branch in which internal flow rate values L_j, V_j approach $-\infty$. It seems that this type of bifurcation diagram has not been explored before in the distillation literature. That could be due to the numerical difficulty in reaching the negative asymptote branch, as well as to the fact that internal safeguards in Aspen Plus prevent RadFrac from converging to any mathematical MESH solution with zero or negative flow rates. Further, we can draw a parallel between this type of asymptotic discontinuity of the MESH model and the degenerate bifurcations of the nonsmooth MESH model observed at λ_{cr}. Both are associated with (near) vertical slopes and (nearly) singular sensitivity matrices.

While there is usually no room for the user to influence the value of $\lambda_{r_{\min}}$ (unless the minimum flow rate is $W_{L,1}, L_N$ or V_1, in which case $\lambda_{r_{\min}}$ depends on $r_{\min}$), the value of $\lambda_{r_{\max}}$ can be a strong function of the chosen $r_{\max}$ depending on the steepness of the asymptote. Figure 4.7 was plotted using $r_{\max} = 5$. In this example, choosing $r_{\max}$ values of 5, 10 and 20 lead to $\lambda_{r_{\max}}$ values of 0.932, 0.933 and 0.938, respectively. Evidently, as $r_{\max} \to +\infty$ we have $\lambda_{r_{\max}} \to \lambda_{\text{asym}}$. The choice of $r_{\max}$ needs to be a compromise, since higher values allow us to reach solutions closer to the asymptote but might make the model equations ill-conditioned and hard to converge near $\lambda = \lambda_{r_{\max}}$.

The single-soft adaptive model flatlines at the minimum flow rate solution ($L_2 = 0$) for $\lambda \leq \lambda_{r_{\min}}$, and at the maximum flow rate solution for $\lambda \geq \lambda_{r_{\max}}$. As with Example 1, in Figure 4.7 we have $L_2 < r_{\max}F_s = 5F_s$ at the maximum flow rate solution due to the fact that L_2 is not the maximum flow rate in the column. With the single-soft adaptive model we can effectively "chop off" the undesirable MESH solutions and obtain a more well-behaved curve in Figure 4.7.

By comparing the behavior of the MESH model solutions depicted in Figures 4.5 and 4.7, we can visualize why specifying R and D (or L_N) is a much easier task than specifying one or more product purities. This difficulty is what leads Aspen Plus' RadFrac model not to accept any product purity specifications directly, requiring instead a design specification procedure that manipulates more well-behaved variables such as R and D. With the single-soft adaptive model, we remove the ill-conditioned infinite discontinuities and can therefore specify product purities directly.

Figure 4.8 presents the type of MESH model solution for each pair of specified $\lambda = x_{1,1}$ and $x_{N,1}$ values, in an analogous fashion to Figure 4.1a. For each value of the hard specification $x_{N,1}$ we have particular values for $\lambda_{r_{\min}}$ and $\lambda_{r_{\text{asym}}}$ which describe the boundary curves in red and in black, respectively, in Figure 4.8. Specification values in the interior of the region described by both types of boundary curves are MESH-feasible, leading to the existence of vapor and liquid phases at every stage. Specifications outside of said region lead to infeasible MESH solutions in which one or more flow rates are negative. For these infeasible specifications, in RadFrac we obtain either dry column errors or the more generic convergence failure messages presented in Section 4.1.1. At each point of the black boundary curve in Figure 4.8 the MESH model exhibits an infinite asymptotic discontinuity in flow rate values, and each point of the red curves corresponds to a minimum flow rate solution.

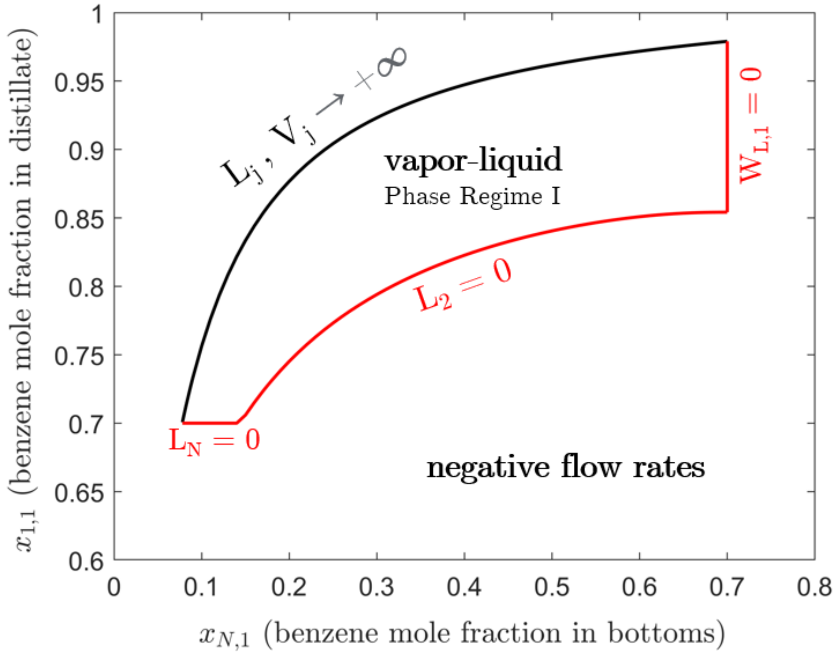


Figure 4.8: Type of MESH model solution for each pair of $x_{1,1}$ and $x_{N,1}$ values in Example 2.

In contrast, Figure 4.9 presents the type of solution of the single-soft model for each pair of $x_{1,1,\text{spec}}$, $x_{N,1,\text{spec}}$ values when using $\lambda = x_{1,1}$ as the soft specification. In this figure, the red boundary curves corresponding to $\lambda = \lambda_{r_{\min}}$ are the same as in Figure 4.8. On the other hand, the black boundary curve now corresponds to $\lambda = \lambda_{r_{\max}}$ for $r_{\max} = 5$ instead of $\lambda = \lambda_{r_{\text{asym}}}$, and is therefore (only) slightly different than in Figure 4.8. Analogously to Figure 4.4b, the vertical arrows illustrate that the single-soft model automatically resets MESH-infeasible values of λ to the boundary of the vapor-liquid region, i.e., either to $\lambda_{r_{\min}}$ or $\lambda_{r_{\max}}$. However, that is only possible for $x_{N,1,\text{spec}}$ values that vertically intersect the vapor-liquid region, in this case, for $0.075 \leq x_{N,1,\text{spec}} \leq 0.7$. For all other $x_{N,1,\text{spec}}$ values it is not possible to reset only $x_{1,1}$ to obtain a feasible MESH solution, and thus the single-soft model exhibits no solution. In such cases both $x_{1,1,\text{spec}}$ and $x_{N,1,\text{spec}}$ are infeasible, therefore we must relax both specifications by using the double-soft adaptive model.

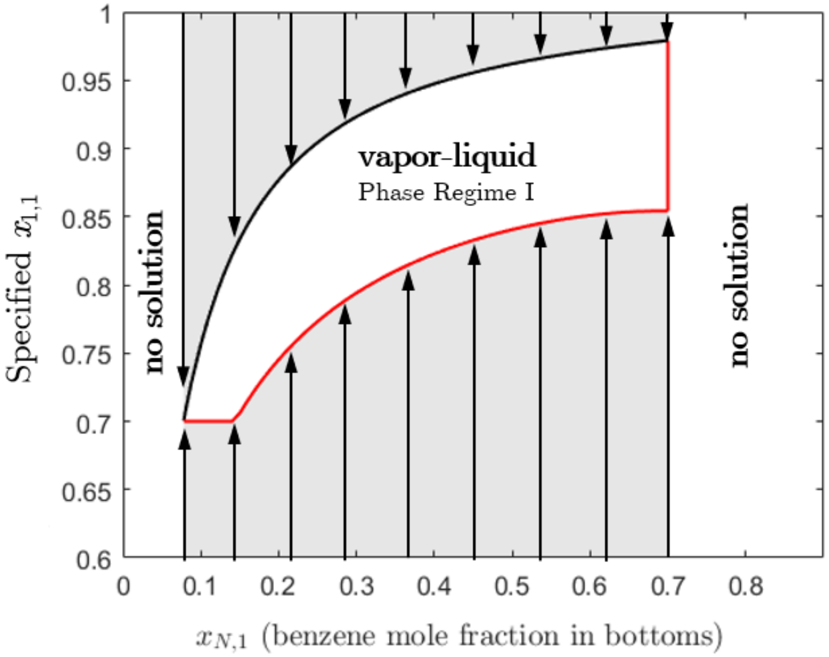
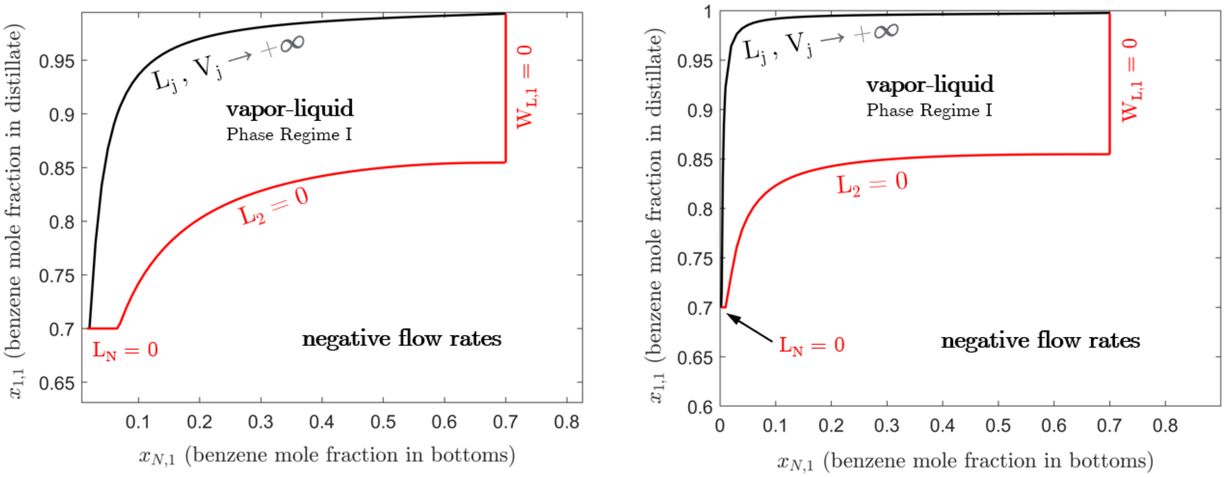


Figure 4.9: Type of single-soft adaptive model solution for each pair of $x_{1,1,\text{spec}}$ and $x_{N,1,\text{spec}}$ values in Example 2.

Figure 4.10 depicts how the diagram from Figure 4.8 changes as we increase the number of stages N from 5 to 7 (Figure 4.10a) and to 10 (Figure 4.10b). The positive flow rate branch of the asymptotic discontinuity at λ_{asym} corresponds to infinite reflux ($R \to +\infty$) operation. For azeotropic non-ideal systems, this scenario can arise from specifying a product purity beyond the relevant azeotrope composition. For ideal systems such as the benzene-toluene Column 2, the $R \to +\infty$ situation can only present itself if the number of stages N is insufficient to promote the specified top/bottom product separation. This can happen for any value of N as long as our product purity specification is close enough to that of a pure mixture, e.g., $x_{1,1,\text{spec}} \approx 1$. As we decrease N, we start to encounter the $R \to +\infty$ discontinuity at less stringent purity specifications, e.g., at $x_{1,1} \approx 0.94$ in Figure 4.7 for $N = 5$. As expected, in Figure 4.10 we can see that the black boundary curve moves closer to the pure component compositions as we increase N.



(a) (b)

Figure 4.10: Type of MESH model solution for each pair of $x_{1,1}$ and $x_{N,1}$ values in Example 2, using (a) $N = 7$ and (b) $N = 10$ instead of $N = 5$.

Finally, Figures 4.11 and 4.12 present the same types of feasibility plots as Figures 4.8 and 4.9, respectively, except that we vary the benzene feed composition z_1 instead of $x_{N,1}$. We observe the same general behavior of both the MESH and the single-soft adaptive models as previously discussed for Figures 4.8 and 4.9. Analyzing how column feasibility varies with respect to the feed composition is particularly relevant within flowsheet simulation, as we might inadvertently encounter two infeasible specifications during intermediate flowsheet passes while feed stream compositions are still being adjusted.

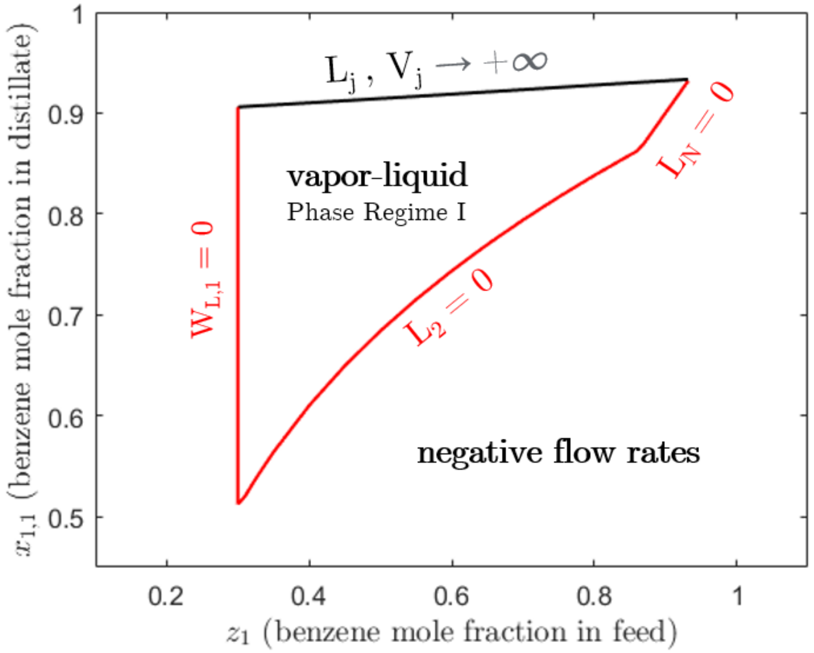


Figure 4.11: Type of MESH model solution for each pair of $x_{1,1}$ and z_1 values in Example 2.

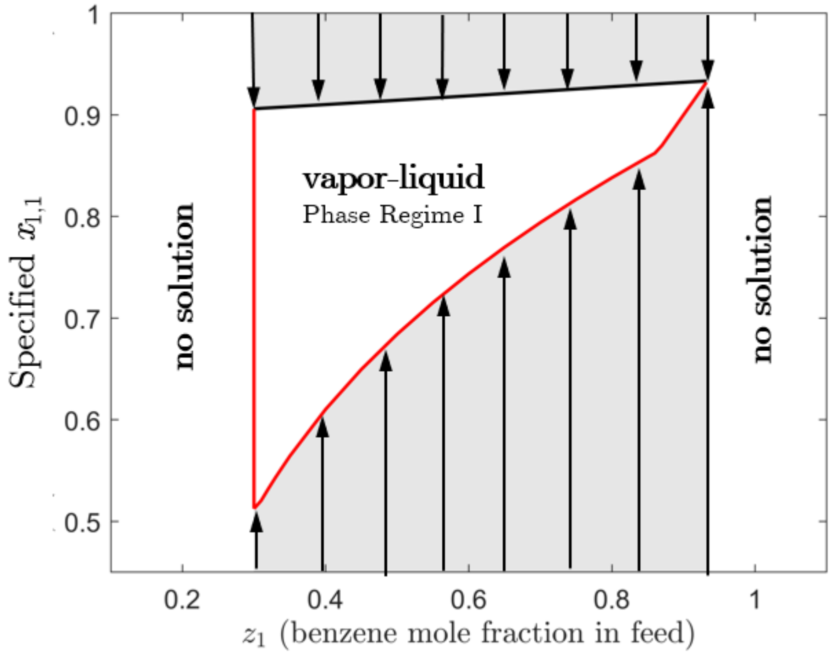


Figure 4.12: Type of single-soft adaptive model solution for each pair of $x_{1,1}$ and z_1 values in Example 2.

4.5.3. Example 3: ethanol-benzene, soft $x_{1,1}$ specification

In this example we study the ethanol-benzene Column 3, which is the first column of the pressure-swing distillation flowsheet from Example 11.5 of [79] (see Figure 5.10). The original specifications are $x_{1,1} = 0.37$ and $x_{N,1} = 0.99$. Since this binary mixture tends to form a minimum-boiling azeotrope at the top of the column, we choose $\lambda = x_{1,1}$ as the soft specification and keep $x_{N,1} = 0.99$ as the hard specification. We use $\alpha = \text{sign}\left(x_{N,1,\text{spec}} - z_1\right) = 1$ according to Table 4.1. Figure 4.13 illustrates how the normalized flow rate L_5/F_s changes with respect to $x_{1,1,\text{spec}}$ for both the MESH and single-soft adaptive models.

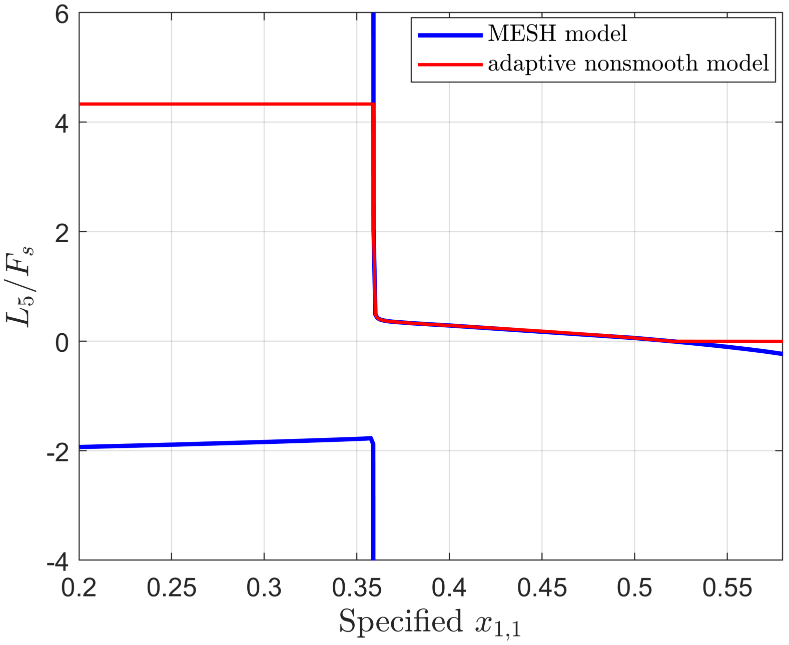


Figure 4.13: L_5/F_s versus $x_{1,1,\text{spec}}$ for Example 3.

In Figure 4.13 we observe the same general behavior as in Figure 4.7 for the ideal benzene-toluene Column 2. However, for Column we have $\lambda_{r_{\max}} < \lambda_{r_{\min}}$. Moreover, in this case $\lambda_{\text{asym}} \approx 0.3585$ corresponds to the composition of a non-trivial azeotrope, instead of the near-pure component composition $\lambda_{\text{asym}} \approx 0.94$ from Example 2. As expected, we observe an infinite flow rate discontinuity corresponding to infinite reflux operation as $x_{1,1}$

approaches the azeotropic composition. Given the steepness of the asymptote, $r_{\max} = 5$ yields a value of $\lambda_{r_{\max}} \approx 0.359$ that is quite close to λ_{asym}.

Interestingly, the MESH model for Example 3 exhibits several other branches of infeasible solutions with negative flow rates not shown in Figure 4.13, some of which are presented in Figure 4.14. The points where two of the branches seem to stop abruptly in Figure 4.14 ($x_{1,1} \approx 0.4$ and $x_{1,1} \approx 0.54$) correspond to asymptotic discontinuities in flow rates other than L_5, which tend to $-\infty$. These branches contain extremely infeasible solutions, with some mole fraction values in the order of 200.

In Figure 4.15 we state the type of MESH solution for each pair of specified $x_{1,1}$, $x_{N,1}$ values, which is analogous to Figure 4.8 for Example 2.

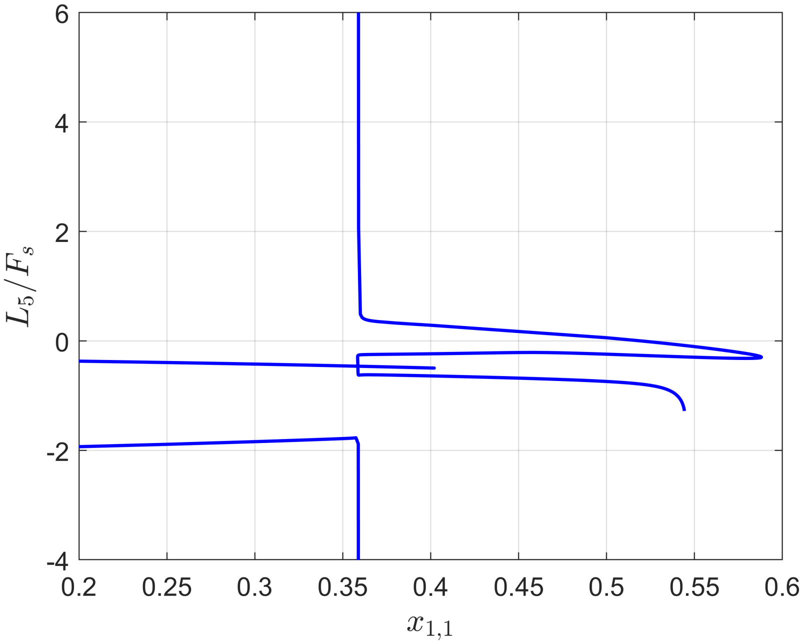


Figure 4.14: Negative flow rate branches in the L_5/F_s versus $x_{1,1}$ plot for Example 3 using the MESH model.

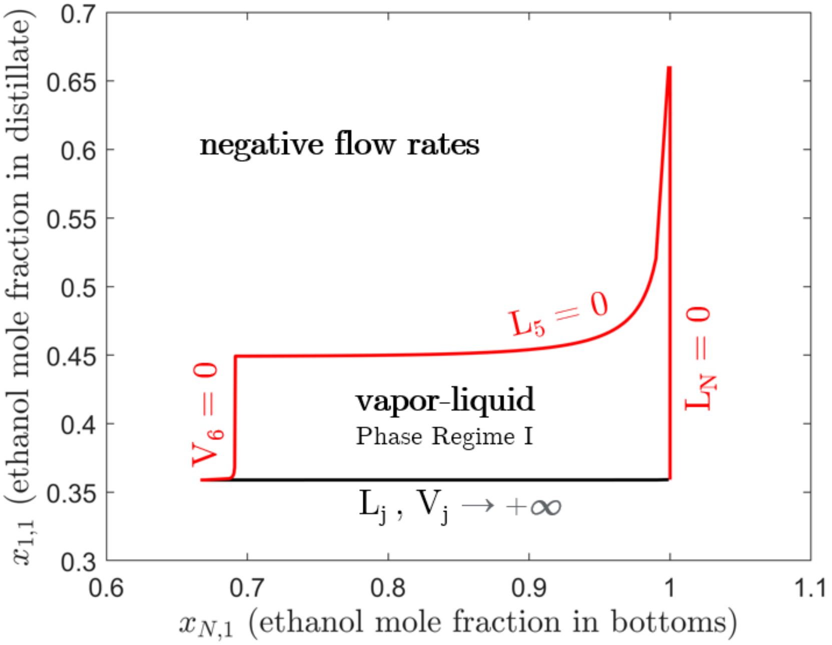


Figure 4.15: Type of MESH model solution for each pair of $x_{1,1}$ and $x_{N,1}$ values in Example 3.

4.5.4. Example 4: diethylamine-methanol, soft $x_{N,1}$ specification

We now analyze the diethylamine-methanol Column 5, which is the first column of the pressure-swing distillation flowsheet from Figure 3a of [92] (see Figure 5.17). This system tends to form a maximum-boiling azeotrope at the bottom of the column, thus we choose $\lambda = x_{N,1}$ as the soft specification and keep $x_{1,1} = 0.996$ as the hard specification. Table 4.1 gives $\alpha = \text{sign}\,(x_{N,1,\text{spec}} - z_1) = 1$. Figures 4.16 and 4.17 present $L_{N-1}/F_s = L_{39}/F_s$ and $W_{L,1}/F_s$ versus $x_{N,1,\text{spec}}$, respectively, for the MESH and single-soft adaptive models. In Figure 4.16 we observe the same behavior as in Figure 4.13 for the azeotropic ethanol-benzene Column 3, also with quite steep asymptotes.

In this example, for which we used $r_{\min} = 0$, the minimum flow rate solution corresponds to the upper section of the column (above the feed stage) having all vapor and liquid flow rates equal to zero. Since $W_{L,1} = 0$ in this case, as seen in Figure 4.17, the bottoms product is identical to the feed stream except for its temperature, which is higher. However, if we use $r_{\min} > 0$, the external product flow rate $W_{L,1}$ becomes the first and unique flow rate to achieve its lower bound of $r_{\min}F_s$, and $\lambda_{r_{\min}}$ becomes a function of the chosen $r_{\min}$ value. We will use $r_{\min} = 0.05$ in Section 5.6.2 to simulate Columns 5 and 6, since $W_{L,1}$ is the first flow rate to become dry in both of them. Finally, we note that the MESH model also exhibits additional negative flow rate branches which we do not plot in Figure 4.16.

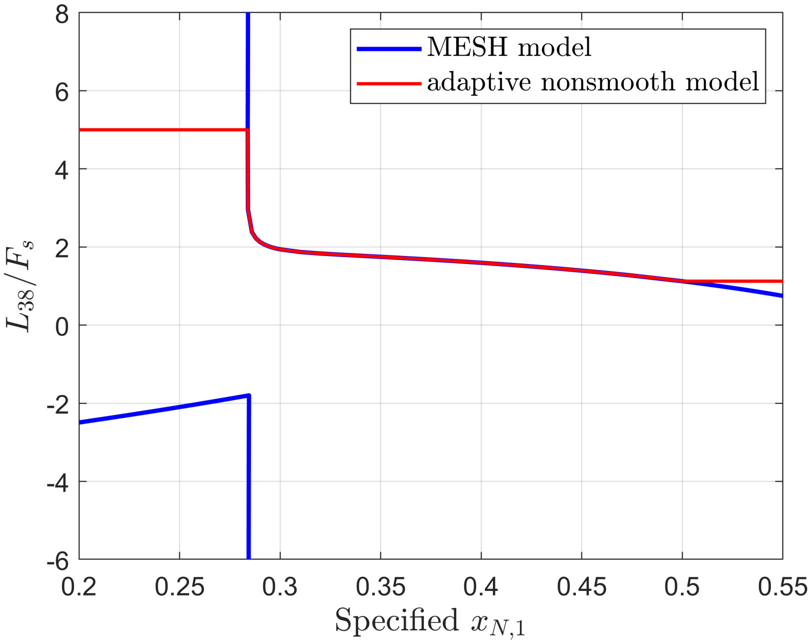


Figure 4.16: L_{38}/F_s versus $x_{N,1,\text{spec}}$ for Example 4.

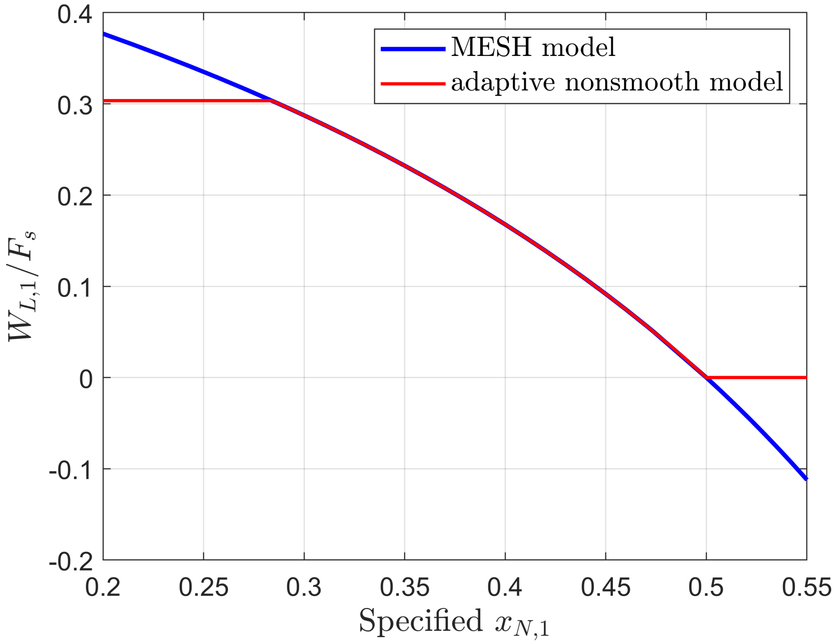


Figure 4.17: $W_{L,1}/F_s$ versus $x_{N,1,\text{spec}}$ for Example 4.

4.5.5. Example 5: five-component mixture, soft $x_{1,1}$ specification

In this last example using the single-soft adaptive model we analyze Column 7, which is the five-component system of Case Study 2 from Section 3.7.1 with a dew-point vapor feed. This highly non-ideal mixture forms several azeotropes. We keep $D = 0.3F_s$ as a hard specification and choose $\lambda = x_{1,1}$ as the soft specification. Figures 4.18 and 4.19 present the bifurcation diagrams of the normalized flow rate V_8/F_s with respect to $x_{1,1,\text{spec}}$ using either $\alpha = 1$ or $\alpha = -1$ in Equation 4.6, respectively. In this example we used a higher value of $r_{\max} = 10$ because the infinite flow rate asymptote at $x_{1,1} \to 0.272$ (corresponding to the azeotrope composition presented in Table 4.2) is approached at a particularly slow rate.

The standard MESH model exhibits multiple feasible steady states with respect to $x_{1,1}$ due to a turning point bifurcation at $x_{1,1} \approx 0.382$. The feasible values of $x_{1,1}$ range from ≈ 0.272 at the infinite flow rate asymptote up to ≈ 0.382 at the turning point, and then back down to ≈ 0.336 when the value of V_8 reaches zero. The curve of MESH solutions,

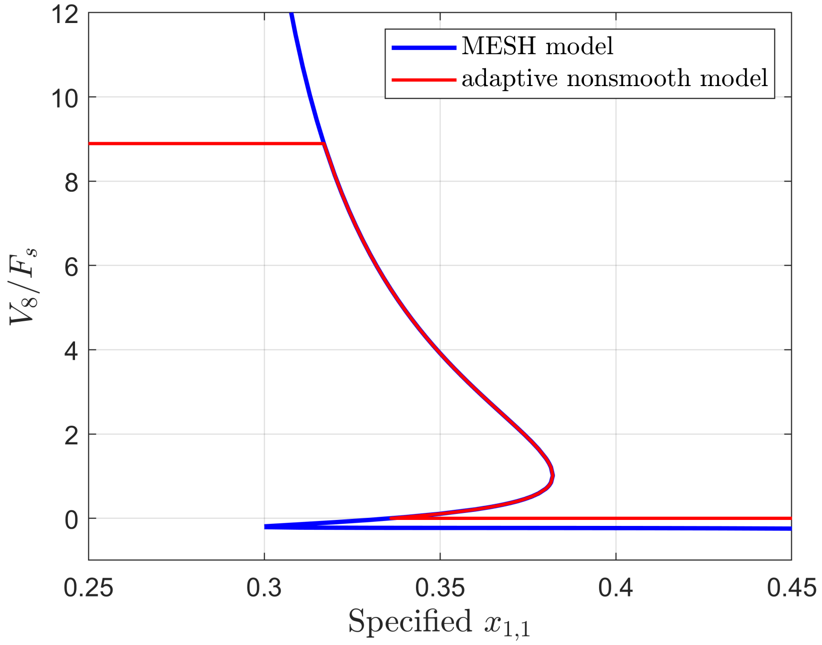


Figure 4.18: V_8/F_s versus $x_{1,1,\text{spec}}$ for Example 5 using $\alpha = 1$ in Equation 4.6.

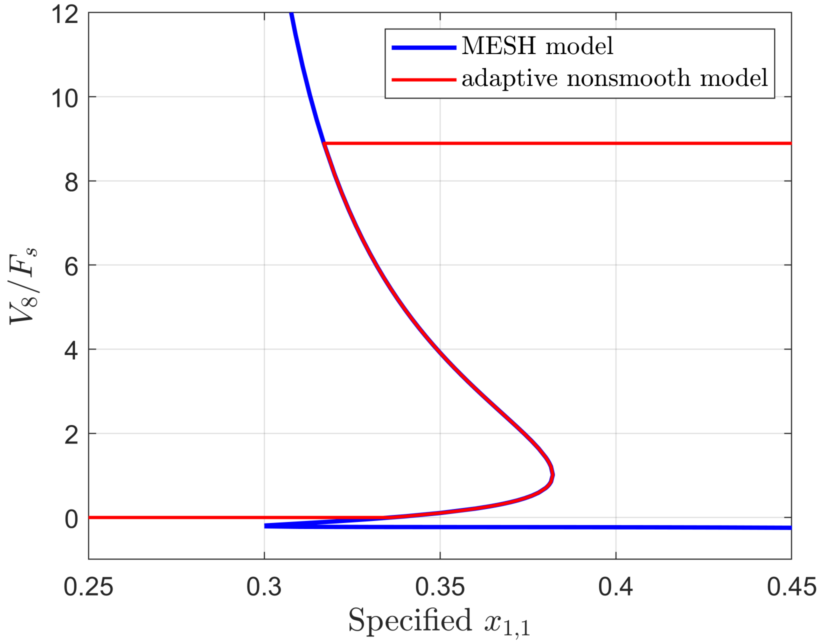


Figure 4.19: V_8/F_s versus $x_{1,1,\text{spec}}$ for Example 5 using $\alpha = -1$ in Equation 4.6.

now with infeasible negative flow rates, continues to vary in the negative direction until another turning point bifurcation happens at $x_{1,1} \approx 0.3$, after which $x_{1,1}$ increases once again.

Methanol (component $i = 1$) is a light component within this mixture. The general strategy stated in Table 4.2 would lead us to use $\alpha = -1$, since in this case we would expect flow rates to increase with $x_{1,1}$. However, the relationship between $x_{1,1}$ and L_j, V_j values is not monotonic in this example and $\lambda_{r_{\max}} \approx 0.317$ ends up being lower than $\lambda_{r_{\min}} \approx 0.336$. Therefore, as seen by comparing Figures 4.18 and 4.19, using $\alpha = 1$ yields the most intuitive parameter-resetting behavior for the single-soft model. This example illustrates the fact that determining the most appropriate value of α to use in Equation 4.6 is not necessarily an obvious task. Nevertheless, we could perhaps argue that choosing the "wrong" value $\alpha = -1$ in this example would still allow the single-soft model to achieve its main goal, given the narrow difference between $\lambda_{r_{\min}}$ and $\lambda_{r_{\max}}$.

In Section 5.4 of Chapter 5 we will analyze the numerical convergence of the single-soft model when using both $\alpha = 1$ and $\alpha = -1$ for this example, and we will also compare the results with those of Aspen Plus' RadFrac model.

4.5.6. Example 6: diethylamine-methanol, double-soft model

In this example we simulate Column 5 using the double-soft adaptive model; that is, we do not specify values for $x_{1,1}$ nor $x_{N,1}$ within the model equations, and instead simply enforce the flow rate bounds $0 \leq L_j, V_j \leq r_{\max} F_s$ through Equations 4.7, 4.8.

First we vary the diethylamine feed composition z_1 from 0.5 to 0.95 in 0.05 increments, and for each z_1 we simulate the double-soft model with the pseudoinverse semismooth Newton method of Section 4.4 starting from the same initial guess, which is either

(a) the point $\mathbf{X}^{\text{init}}$ obtained according to the generic initialization procedure from Section 4.3.3 with $z_1 = 0.5$, which is not biased towards any specific $x_{1,1,\text{spec}}$ or $x_{N,1,\text{spec}}$ values, or

(b) the point $\mathbf{X}^{\text{guess}}$ obtained by setting all internal flow rates equal to $0.2F_s$, $D = L_N = 0.5F_s$, and imposing constant mole fraction and temperature values $\mathbf{x}_j =$

$\mathbf{x}_f = (0.461, 0.539)$, $\mathbf{y}_j = \mathbf{y}_f = (0.575, 0.425)$, and $T_j = T_m = 337.5$ K for all stages j, where $\mathbf{x}_f, \mathbf{y}_f, T_m$ are computed from Procedure 4.3.1 using $z_1 = 0.5$.

Figures 4.20a and b present the $x_{1,1}$ and $x_{N,1}$ values, and the internal flow rates L_j, V_j of minimum and maximum magnitude, respectively, at the solution obtained for each z_1 using $\mathbf{X}^{\text{init}}$ as the initial guess. Figures 4.21a and b show the same variables obtained when using $\mathbf{X}^{\text{guess}}$ as the initial guess.

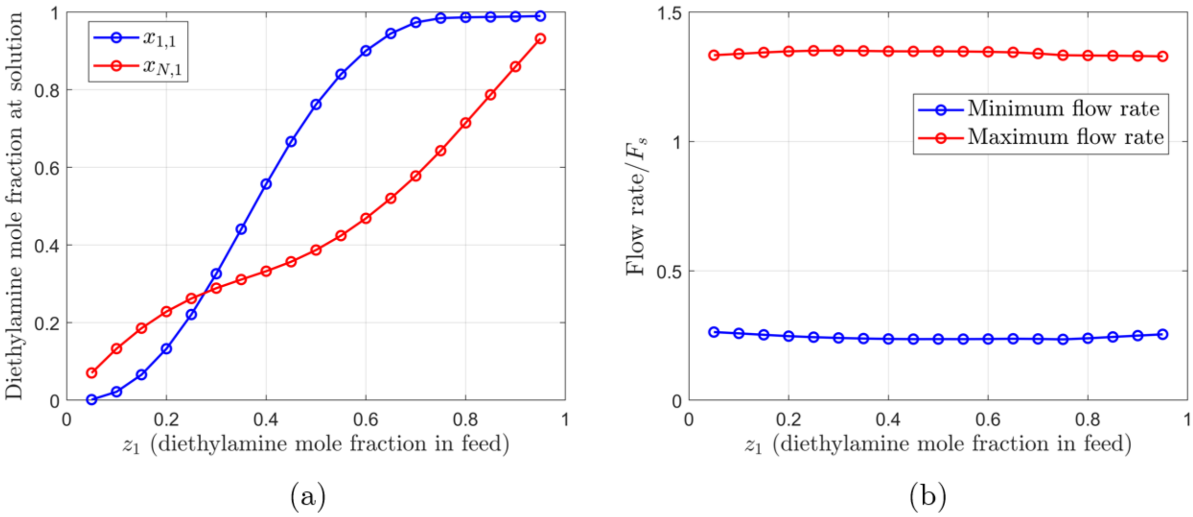


Figure 4.20: Solutions obtained for each z_1 value with the pseudoinverse semismooth Newton method starting from $\mathbf{X}^{\text{init}}$: (a) $x_{1,1}$, $x_{N,1}$ values, (b) Maximum and minimum internal flow rates.

Given that two degrees of freedom are relaxed in the double-soft model, we have a two-dimensional continuum of infinitely-many, non-isolated solutions for each set of simulation specifications. That stands in contrast to systems with multiple (yet finitely many) steady states such as the five-component Column 7 from Example 5. As expected, by comparing Figures 4.20 and 4.21 we see that the solution we converge to depends on the initial guess. Though any point of the two-dimensional set of solutions can technically be reached, we often observe a preference towards certain points. In both $x_{1,1}, x_{N,1}$ plots we can observe a general bias of the model in converging to "low-separation" solutions, in which the top and bottom compositions gravitate around the feed composition itself. Moreover, we

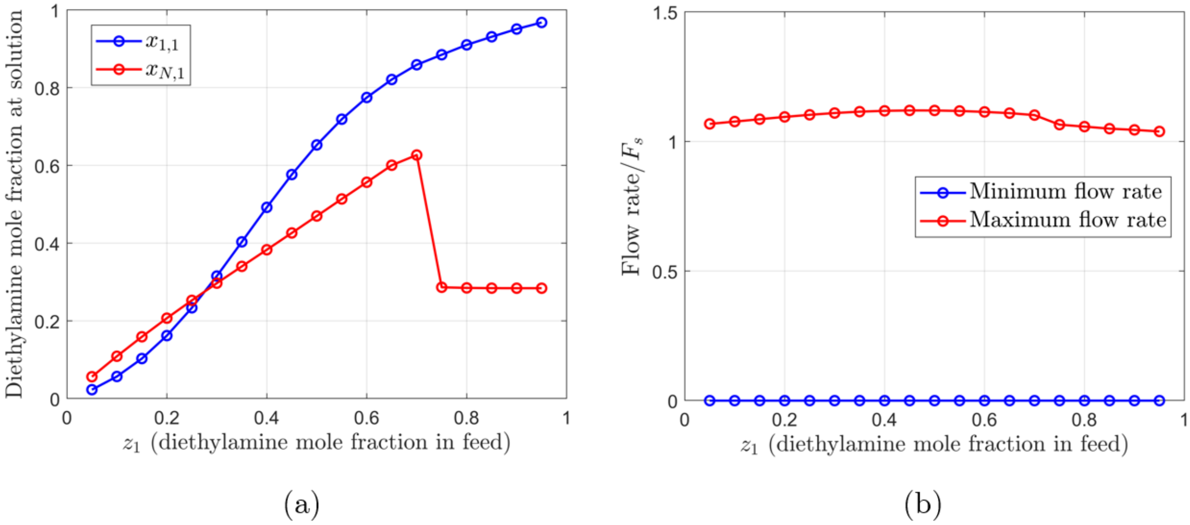


Figure 4.21: Solutions obtained for each z_1 value with the pseudoinverse semismooth Newton method starting from $\mathbf{X}^{\text{guess}}$: (a) $x_{1,1}$, $x_{N,1}$ values, (b) Maximum and minimum internal flow rates.

see that the solutions reached with both initial guesses are similar in general for most z_1 values. Unlike with the single-soft model, the flow rate bounds may or may not be strictly enforced at the solution we converge to. In Figures 4.20b and 4.21b we see that the minimum internal flow rate L_j, V_j achieves the lower bound of zero at the solution when we start from $\mathbf{X}^{\text{guess}}$, while no bounds are reached when we start from $\mathbf{X}^{\text{init}}$.

In general, the generalized Jacobian matrices of the double-soft model are singular at every solution reached and essentially at every iterate. As discussed in Section 2.3.5, in such cases there are currently no known theoretical conditions under which quadratic converge, or even convergence itself, is guaranteed for the pseudoinverse semismooth Newton method. Nevertheless, in our test cases we have observed that this method is able to convergence the double-soft model fastly and reliably. Moreover, as illustrated in Figure 4.22 for $\mathbf{X}^{\text{init}}$ as the initial guess, the typical convergence rate of our algorithm is quadratic.

We then performed the same set of simulations using the LP-Newton method to converge the double-soft model equations. Figure 4.23 presents the results obtained when using $\mathbf{X}^{\text{init}}$ as the starting point. The LP-Newton method failed to converge for all tested z_1 values when starting from $\mathbf{X}^{\text{guess}}$; specifically, the algorithm got stuck at the zero lower bound constraint for almost all mole fractions and flow rates.

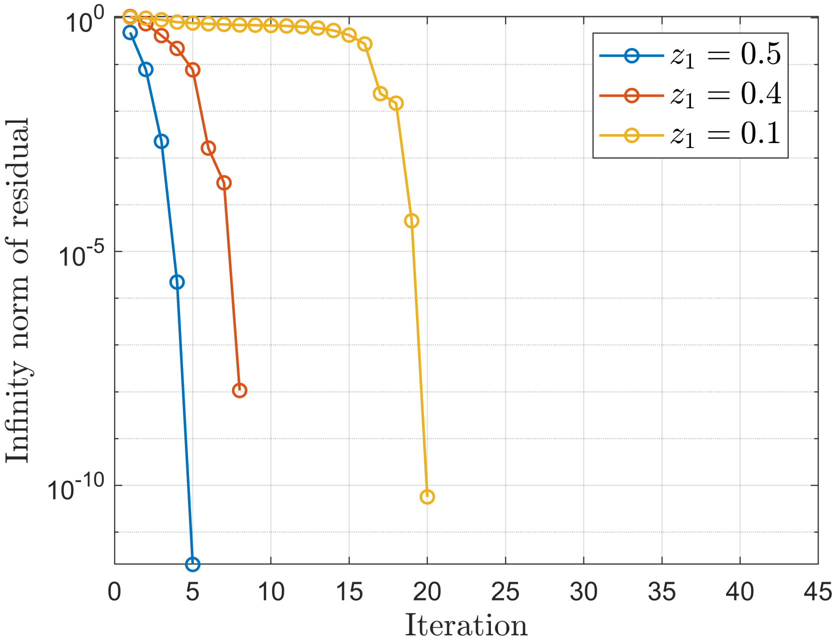


Figure 4.22: Convergence rate of the pseudoinverse semismooth Newton method when using $\mathbf{X}^{\text{init}}$ as the initial guess.

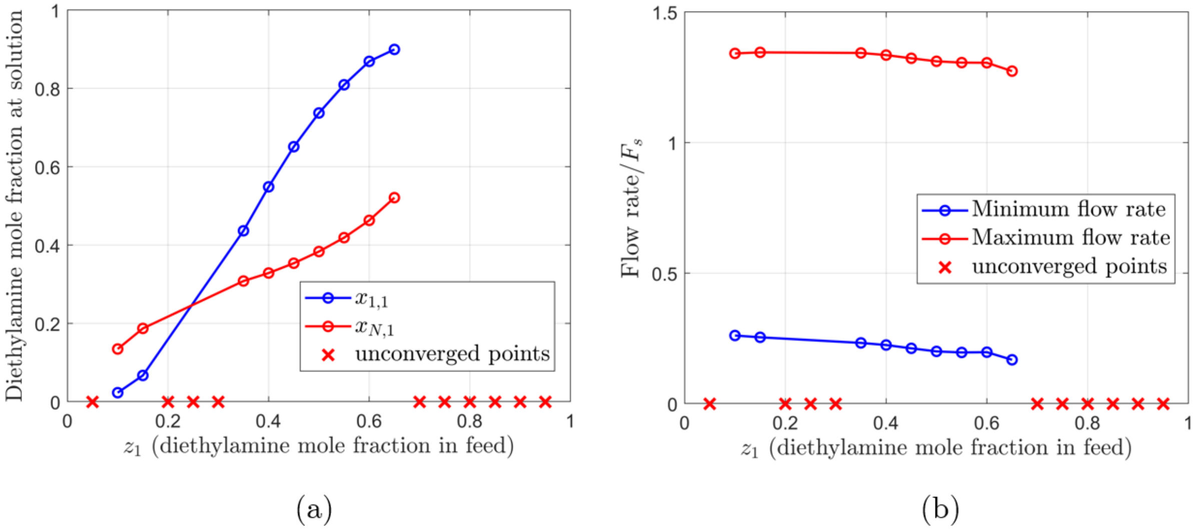


Figure 4.23: Solutions obtained for each z_1 value with the LP-Newton method starting from $\mathbf{X}^{\text{init}}$: (a) $x_{1,1}$, $x_{N,1}$ values, (b) Maximum and minimum internal flow rates.

We have observed that the LP-Newton method fails to approach a solution often enough to render it unsuitable for solving the double-soft model in general, despite its theoretical ability to reach non-isolated solutions (see Section 2.3.3). Further, in our MATLAB implementation the LP-Newton step becomes undefined when the method does get near a solution due to the linear program from Equation 2.31 being deemed infeasible by the `linprog` solver. Therefore, we had to increase the tolerance on the norm residual to 10^{-6} or lower to converge the LP-Newton method. Given that we did not observe this unexpected issue in any of our single-soft model test cases, we can speculate that it may be related to the structural singularity of the generalized Jacobians of the double-soft model.

For z_1 values for which the LP-Newton method does converge in Figure 4.23, it reaches essentially the same solution as the pseudoinverse semismooth Newton method does when starting from the same initial guess $\mathbf{X}^{\text{init}}$. This indicates that the double-soft model bias towards certain solutions is not algorithm-specific. When successful, the convergence speed of the LP-Newton method is considerably lower than that of the pseudoinverse semismooth Newton method, as illustrated in Figure 4.24. In said figure we observe a linear behavior for most of the iteration span, with a quadratic slope only being observed much nearer the solution when compared to Figure 4.22.

In a second case study for Column 5, we fix $z_1 = 0.5$ and attempt to steer the double-soft model towards converging to the solution $\mathbf{X}^*$ that corresponds to $x_{1,1} = x_{1,1,\text{spec}} = 0.9$ and $x_{N,1} = x_{N,1,\text{spec}} = 0.3$, which are the MESH-feasible specifications originally used in [79]. We change the initial guess $\mathbf{X}^0$ according to a homotopy transformation

$$\mathbf{X}^0 = \gamma \mathbf{X}^* + (1 - \gamma)\mathbf{X}^{\text{guess}} \tag{4.14}$$

from $\mathbf{X}^{\text{guess}}$ into the solution $\mathbf{X}^*$. Figure 4.25 presents the $x_{1,1}$ and $x_{N,1}$ values at the obtained solution using the pseudoinverse semismooth Newton method for each value of γ, which we varied from 0 to 1 in 0.05 increments. As shown in Figure 4.26, the algorithm's convergence rate remains quadratic throughout the continuation procedure, with less iterations needed as γ approaches 1.

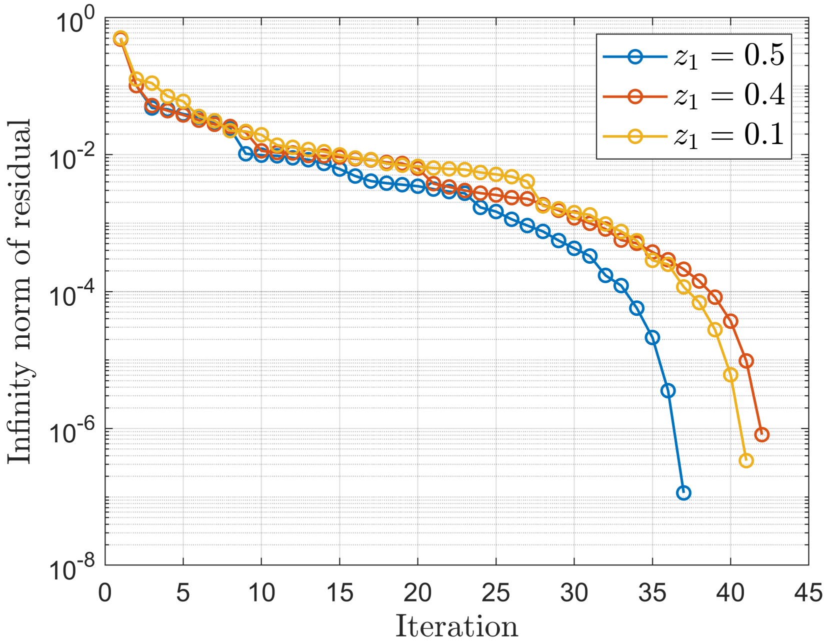


Figure 4.24: Convergence rate of the LP-Newton method when using $\mathbf{X}^{\text{init}}$ as the initial guess.

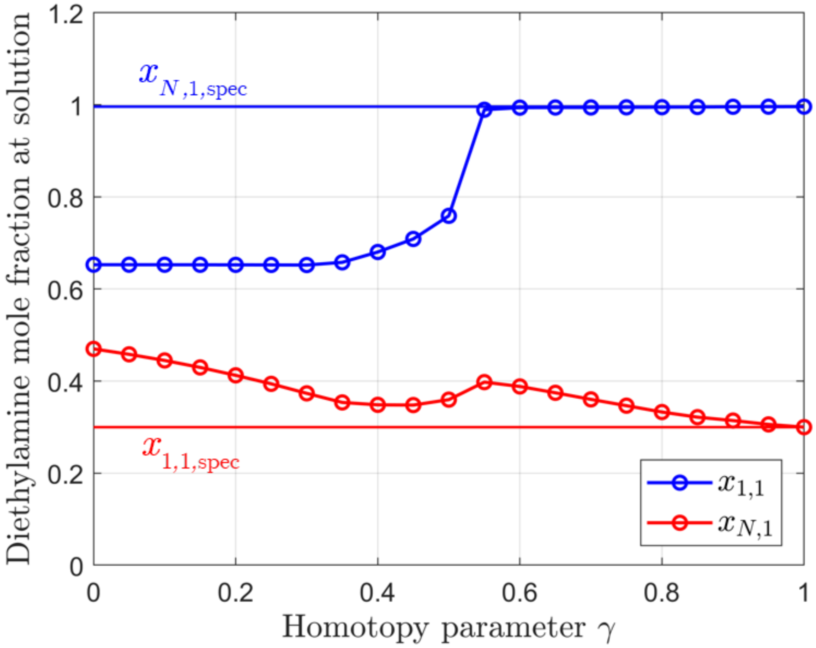


Figure 4.25: $x_{1,1}$ and $x_{N,1}$ values at the solution obtained with the pseudoinverse semismooth Newton method using the initial guess given by Equation 4.14.

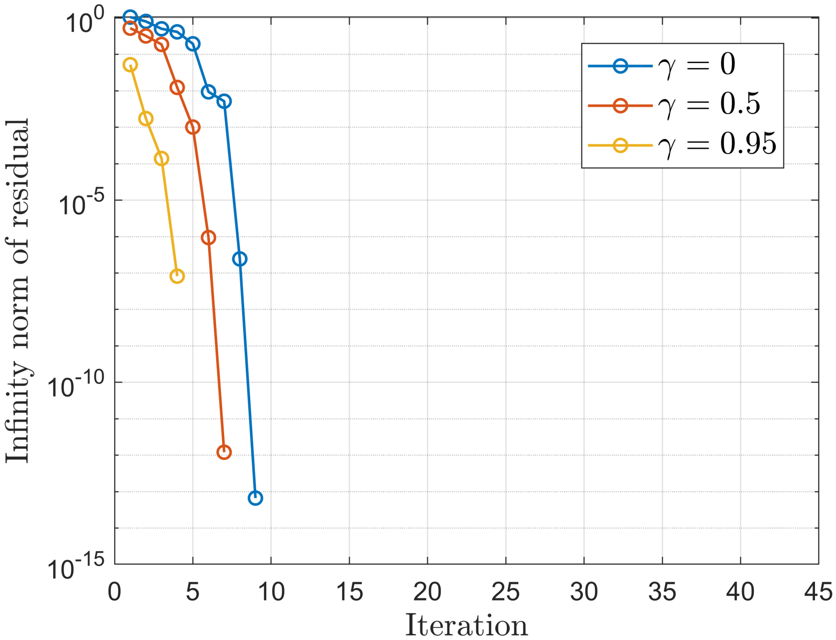


Figure 4.26: Convergence rate of the pseudoinverse semismooth Newton method for $z_1 = 0.5$ when using the initial guess from Equation 4.14.

The double-soft model has a general tendency not to deviate too much from the initial guess, since the first iterate that satisfies all the first-principles model equations will be taken as the final solution. As expected, in Figure 4.25 the double-soft solution we converge to approaches the MESH feasible point $\mathbf{X}^*$ as γ approaches 1. In general, we cannot realistically expect the initial guess to be near a desirable MESH-feasible point for single-column simulation with the double-soft model. However, we do encounter this scenario when converging flowsheets with recycle streams, since we use solutions from the previous flowsheet pass as the initial guess for each equipment. As will be exemplified in Section 5.6.2, this property of the double-soft model may allow us to converge to a flowsheet solution in which at least one of the relaxed column specifications λ is approximately equal to its desired value λ_{spec}, even when the latter is infeasible.

We performed the same homotopy procedure with the LP-Newton method; interestingly, as illustrated in Figure 4.27, the method only (slowly) converges when starting extremely near the point $\mathbf{X}^*$ with values of $\gamma \geq 0.95$.

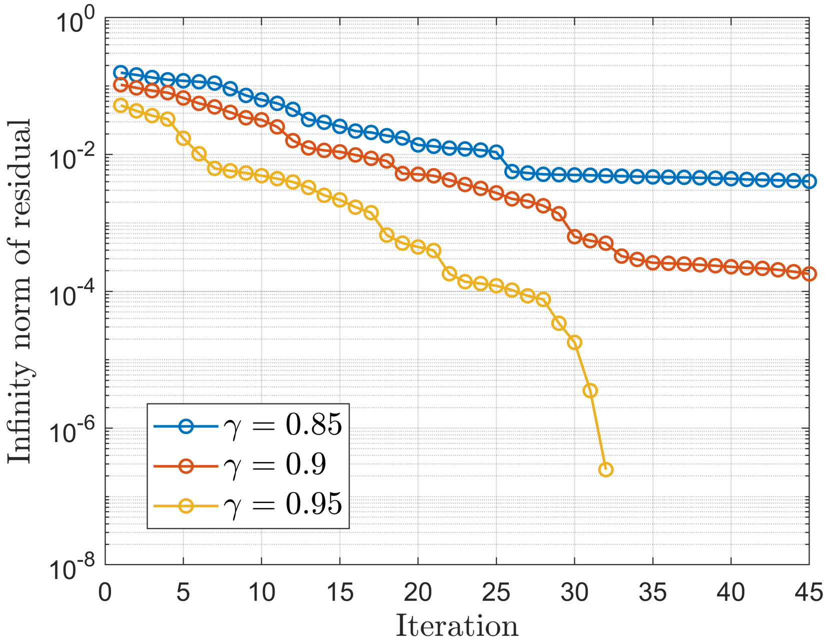


Figure 4.27: Convergence rate of the LP-Newton method for $z_1 = 0.5$ when using the initial guess from Equation 4.14.

4.6. Conclusions

In this chapter we presented two types of nonsmooth adaptive models for distillation that can be used to provide an alternative MESH-feasible solution even when one or more specifications are infeasible. The single-soft adaptive model automatically resets a user-chosen soft specification if it happens to be infeasible due to flow rates L_j, V_j going out of bounds, and returns the "nearest" MESH solution in which either the upper or lower bound is strictly enforced. Consequently, the single-soft model eliminates the negative flow rate solutions exhibited by the MESH model, as well as the infinite asymptotic discontinuities in flow rate values observed when specifying product purities. This new modeling strategy also removes the degenerate bifurcations of the nonsmooth MESH model associated with dry/vaporless solutions. The double-soft model can be used to converge to a MESH-feasible solution even when both column specifications are infeasible. This model exhibits a two dimensional set of non-isolated solutions, which can nevertheless be reached through the pseudoinverse semismooth Newton method with a typical quadratic convergence rate.

Chapter 5

Nonsmooth inside-out algorithms for robust distillation simulation with non-ideal mixtures

5.1. Introduction

Simultaneous convergence of the MESH equations in their original or primitive form can be unreliable and highly dependent on a good initial guess, especially with complex non-ideal multicomponent systems. To overcome this problem Boston and Sullivan Jr [14] introduced the inside-out method, an algorithmic strategy that creates 2 nested “inner” and “outer” loops to improve reliability of convergence and reduce the cost of repeated thermodynamic property evaluations. In the outer loop, the rigorous property models for the equilibrium K-values and vapor and liquid phase enthalpy departures are used to generate simple linearized models. The inner loop employs the latter to converge a rearrangement of the MESH equations, for which only the energy balances and column specifications need to be converged iteratively with equation-solving methods.

The inner loop of Boston and Sullivan Jr relies on the product flow rates and the reflux ratio being specified by the user; this allows the inner loop to be formulated as a fixed-point problem. Broyden's method is then used with the identity as the initial Jacobian matrix, thus finite differencing need not be performed even once and the inner loop can be converged at very low cost. Later on Boston added a third "middle" loop, which adjusts the product flow rates and reflux ratio values used within the inner loop with optimization methods, to be able to enforce more general specifications. However, the exact structure of the middle loop was only presented by Boston at an AIChE conference in 1979, as reported by Russell (see Reference 12 in [75]), and was not made available in the literature. In 1983, Russell [75] proposed a modified inner loop that converges the set of energy balances and general specifications simultaneously, without the need for an extra middle loop. An approximation to the Jacobian matrix is computed with finite differences and subsequently updated with Broyden's formula. Several other versions of the inside-out method have been proposed, such as adaptations to reactive distillation systems [82]; in particular, RadFrac in Aspen Plus uses a proprietary modification of the Boston and Sullivan Jr method which includes the middle loop (see Section 4.1.1).

After the inside-out algorithm for multistage columns was presented, Boston and Britt [13] introduced another version of the method tailored to the single-stage flash problem. The general outer loop structure remained essentially the same, while the flash inner loop variable corresponds to a "weighted" vapor fraction instead of the so-called stripping factors used in distillation (see Section 5.2.4). In [87], Watson et al. developed a nonsmooth version of the flash inside-out algorithm which can reliably converge to solutions in any phase regime (i.e., Phase Regimes I, II or III, as defined in Section 3.2.1). The modified inner loop involves two variables, and two equations: the energy balance residual and the nonsmooth Equation 3.12.

In this chapter we present a nonsmooth inside-out algorithm to converge both the single-soft and the double-soft adaptive models for distillation from Chapter 4. The algorithm employs the outer loop structure of the standard inside-out algorithm, which allows for reliable and low-cost convergence under non-ideal thermodynamics, and introduces a modified nonsmooth inner loop. The latter can be solved with nonsmooth equation-solving methods using automatically-computed generalized derivative elements. Distillation simulation with our nonsmooth inside-out algorithms is robust to dry column errors and to infeasible specifications, and reliably converges to a feasible MESH solution even when highly nonlinear, non-ideal thermodynamic property models are used.

5.2. The multistage inside-out algorithm

In this section we will describe the inside-out algorithm according to the general structure of Russell's method [75]. However, we also include the possibility of modeling the activity coefficient as a separate factor within the K-values as proposed by Boston in [15]. The algorithm described here is the basis method that we will modify in Section 5.3.

We denote the set of physical or so-called "primitive" column variables as

$$\mathbf{X} = (\mathbf{x}, \mathbf{y}, \mathbf{L}, \mathbf{V}, \mathbf{T}, \mathbf{W}_L, \mathbf{W}_V), \tag{5.1}$$

where $\mathbf{L}, \mathbf{V}, \mathbf{T} \in \mathbb{R}^N$, $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{N \times N_c}$, $\mathbf{W}_L \in \mathbb{R}^{n_{W_L}}$ (which includes the liquid distillate stream with flow rate W_{L_1}) and $\mathbf{W}_V \in \mathbb{R}^{n_{W_V}}$ are vectors or matrices containing the flow rates, temperatures and mole fractions for each stage as defined in Chapter 3. Here, n_{W_L} and n_{W_V} are the numbers of liquid and vapor side-product streams, respectively. In order to describe the inner loop we also define $l_{j,i}$ as the liquid flow rate for each individual component i at stage j, and $\mathbf{l} \in \mathbb{R}^{N \times N_c}$ as the matrix containing all $l_{j,i}$ values.

In describing the inside-out algorithm in the following sections, we assume that the vapor distillate fraction θ, the feed stream conditions, stage pressures, heat duties Q_j and side product ratios $W_{L,j}/L_j, W_{V,j}/V_j$ or flow rates $W_{L,j}, W_{V,j}$ for intermediate stages $2 \leq j \leq N-1$, and the two main degrees of freedom $\lambda_1 - \lambda_{1,\text{spec}} = 0$ and $\lambda_2 - \lambda_{2,\text{spec}} = 0$ have been specified by the user.

5.2.1. Initialization

An initial guess $\mathbf{X}^0$ for the primitive variables must be specified through some initialization procedure. In [14], Boston and Sullivan Jr propose the following sequence of calculations:

Procedure 5.2.1 (Column initialization from [14]).

Given compositions and flow rates of feed streams, stage pressures, and specified reflux ratio and product flow rates:

1. Combine all column feed streams to form a composite feed with composition $\mathbf{z}_c$ and flow rate F_s.

2. Set T_1 and T_N as the bubble-point and dew-point temperatures of the composite feed, respectively, at pressure $P_m = (P_1 + P_N)/2$. Obtain the remaining $\mathbf{T}$ values via linear interpolation.

3. Perform a PT-flash of the composite feed at $P = P_m$ and $T = (T_1 + T_N)/2$ and obtain liquid and vapor mole fractions $\mathbf{x}_f, \mathbf{y}_f$. Set $\mathbf{x}_j = \mathbf{x}_f$, $\mathbf{y}_j = \mathbf{y}_f$ for each stage j.

4. Obtain $\mathbf{L}$ and $\mathbf{V}$ values from constant molar overflow calculations, using the reflux ratio and the product flow rates.

5. Solve the MES equations from the inner loop using $K_{b,j} = 1$ and then update the $\mathbf{X}$ values (see Section 5.2.5).

In [75], Russell reports that one can simply set internal column temperatures and flow rates to be of the same magnitude as those of the feed stream(s) to obtain a loose estimate $\mathbf{X}^0$. Then, the $S_{b,j}$ scaling factors defined in Section 5.2.4 are heuristically adjusted in the first outer loop iteration so that some chosen flow rate in the column is as specified or estimated; however, this procedure is not described further by the author.

5.2.2. Thermodynamic properties

In its basic format, the inside-out method expresses K-values as a product of two terms:

$$K_{j,i} = \alpha_{j,i} K_{b,j}, \tag{5.2}$$

where $\alpha_{j,i} \equiv K_{j,i}/K_{b,j}$ is a relative volatility with respect to a "weighted average" K-value defined as

$$\ln K_{b,j} \equiv \frac{\sum_i t_{j,i} \ln K_{j,i}}{\sum_i t_{j,i}}, \quad t_{j,i} \equiv y_{j,i} \frac{\partial \ln K_{j,i}}{\partial(1/T_j)}. \tag{5.3}$$

For considerably non-ideal mixtures, we can choose to express K-values using the activity coefficient as a separate, third factor:

$$K_{j,i} = \alpha_{j,i} K_{b,j} \gamma_{j,i}. \tag{5.4}$$

In this case, the definition of $K_{b,j}$ remains the same while the "pseudo" relative volatility is redefined as $\alpha_{j,i} \equiv K_{j,i}/(K_{b,j}\gamma_{j,i})$.

The molar vapor and liquid phase enthalpies H_j^V, H_j^L are expressed using molar enthalpy departures $\Delta H_j^V, \Delta H_j^L$ from ideal gas mixture state, defined as

$$\Delta H_j^V = H_j^V - \sum_i y_{j,i} H_i^{\mathrm{ig}}(T_j), \tag{5.5}$$

$$\Delta H_j^L = H_j^L - \sum_i x_{j,i} H_i^{\mathrm{ig}}(T_j), \tag{5.6}$$

where the ideal gas molar enthalpy for pure component i, H_i^{ig}, is usually given as a polynomial-like function of temperature.

5.2.3. Simplified thermodynamic property models

The outer loop creates simplified models for the thermodynamic properties $K_{b,j}, \Delta H_j^V, \Delta H_j^L$ and optionally also for $\gamma_{j,i}$, expressing them as local linear functions of temperature and/or composition. According to the general approach of Russell's version of the inside-out method, the simplified models are

$$\ln K_{b,j} = A_j + \frac{B_j}{T_j}, \tag{5.7}$$

$$\frac{\Delta H_j^V}{\sum_i y_{j,i} MM_i} = C_j + D_j(T_j - T_j^*), \tag{5.8}$$

$$\frac{\Delta H_j^L}{\sum_i x_{j,i} MM_i} = E_j + F_j(T_j - T_j^*), \tag{5.9}$$

where MM_i stands for the molar mass of component i, and T_j^* is a constant reference temperature for stage j; the latter can be defined based on the initial guess for the temperature values T_j. We divide the enthalpy departures ΔH by the molar mass of the vapor or liquid phase because, according to Boston [15], this ensures that the C, D, E, F coefficients remain relatively insensitive to both temperature and composition.

In case the activity coefficient is included as a separate factor in Equation 5.4, then Boston [15] treats it as a pseudo-binary function of composition according to the following simplified model:

$$\ln \gamma_{j,i} = a_{j,i} + b_{j,i} x_{j,i}. \tag{5.10}$$

5.2.4. Algorithmic structure

The inside-out algorithm does not manipulate the primitive variables $\mathbf{X}$ directly. Instead, the high-level task of this method is to converge the outer loop, which is the fixed-point problem

$$\mathbf{f}_{\text{outer}}(\mathbf{v}_{\text{outer}}) = \mathbf{v}_{\text{outer}}. \tag{5.11}$$

The outer loop iteration variables correspond to the parameters of the simplified thermodynamic property models:

$$\mathbf{v}_{\text{outer}} = (\boldsymbol{\alpha}, \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}, \mathbf{F}, \mathbf{a}, \mathbf{b}), \tag{5.12}$$

where $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}, \mathbf{F} \in \mathbb{R}^N$ and $\boldsymbol{\alpha} \in \mathbb{R}^{N \times N_c}$; the parameters $\mathbf{a}, \mathbf{b} \in \mathbb{R}^{N \times N_c}$ are only included if the activity coefficient is modeled as a separate factor in Equation 5.4.

The outer loop residual function $\mathbf{f}_{\text{outer}}$ will be described in Section 5.2.6. Direct substitution is the the main fixed-point method to converge the outer loop in Russell's approach [75], except that a damping factor is employed for the volatilities $\alpha_{j,i}$ in non-ideal systems. Moreover, the "slope" coefficients B_j, D_j, F_j are only conditionally updated if temperature variations are large enough. The outer loop is converged when the simplified model parameters stop changing, i.e., when $||\mathbf{f}_{\text{outer}}(\mathbf{v}_{\text{outer}}) - \mathbf{v}_{\text{outer}}||_\infty < \epsilon_{\text{outer}}$, where $\epsilon_{\text{outer}} > 0$ is some chosen tolerance. We can use the initial guess $\mathbf{X}^0$ for the primitive variables to initialize $\mathbf{v}^0_{\text{outer}}$ according to Section 5.2.6.

Each evaluation of the outer loop residual function $\mathbf{f}_{\text{outer}}$ requires converging the inner loop, which is a fixed-point problem in the original Boston and Sullivan Jr method and the following nonlinear system of equations in Russell's method:

$$\mathbf{f}_{\text{inner}}(\mathbf{v}_{\text{inner}}, \mathbf{v}_{\text{outer}}) = \mathbf{0}, \tag{5.13}$$

where $\mathbf{v}_{\text{outer}}$ is treated as a fixed parameter and $\mathbf{v}_{\text{inner}}$ must be solved for. The inner loop iteration variables $\mathbf{v}_{\text{inner}}$ are defined as $\ln(S_j/S_{b,j})$ for each stage $1 \leq j \leq N$, and $\ln(1 + R_{L,j})$ and $\ln(1 + R_{V,j})$ for stages where liquid and vapor side products are present, respectively, where

$$S_j = \frac{K_{b,j} V_j}{L_j} \tag{5.14}$$

are so-called stripping factors, $K_{b,j}$ values are defined in Equation 5.3, $R_{L,j} = W_{L,j}/L_j$ and $R_{V,j} = W_{V,j}/V_j$. According to Russell, the $S_{b,j}$ values are scaling factors that are heuristically adjusted in the first outer loop iteration. The logarithm is taken to prevent liquid and vapor flow rates from becoming near zero or negative within intermediate iterations.

The inner loop residual function $\mathbf{f}_{\text{inner}}$ will be defined in Section 5.2.5. The inner loop is converged for a given $\mathbf{v}_{\text{outer}}$ when $\|\mathbf{f}_{\text{inner}}(\mathbf{v}_{\text{inner}}, \mathbf{v}_{\text{outer}})\|_{\infty} < \epsilon_{\text{inner}}$, where $\epsilon_{\text{inner}} > 0$ is some chosen tolerance. In iteration $k = 0$ of the outer loop, the initial guess for $\mathbf{v}_{\text{inner}}$ can be computed from $\mathbf{X}^0$ using the definitions of the inner loop variables $S_j, R_{L,j}, R_{V,j}$. For each subsequent outer loop iteration $k > 0$, we use the previously converged value of $\mathbf{v}_{\text{inner}}$ in iteration $k-1$ as the initial guess.

5.2.5. The inner loop residual function

For a given pair of vectors $\mathbf{v}_{\text{inner}}, \mathbf{v}_{\text{outer}}$, the inner loop residual function $\mathbf{f}_{\text{inner}}(\mathbf{v}_{\text{inner}}, \mathbf{v}_{\text{outer}})$ is evaluated as follows. First, we solve the following system for the component liquid flow rates $l_{j,i}$:

$$\left[1 + R_{L,1} + \alpha^*_{1,i} S_1\right] l_{1,i} - \left[\alpha^*_{2,i} S_2\right] l_{2,i} = f_{1,i}, \tag{5.15}$$

$$-l_{j-1,i} + \left[1 + R_{L,j} + \alpha^*_{j,i} S_j (1 + R_{V,j})\right] l_{j,i} - \left[\alpha^*_{j+1,i} S_{j+1}\right] l_{j+1,i} = f_{j,i}, \quad 2 \leq j \leq N-1, \tag{5.16}$$

$$-l_{N-1,i} + \left[\alpha^*_{N,i} S_N (1 + R_{V,N})\right] l_{N,i} = f_{N,i}. \tag{5.17}$$

where

$$\alpha^*_{j,i} = \begin{cases} \alpha_{j,i}, & \text{if Equation 5.2 is used;} \\ \alpha_{j,i} \exp\left(a_{j,i} + b_{j,i} \frac{l_{j,i}}{\sum_k l_{j,k}}\right), & \text{if Equation 5.4 is used,} \end{cases} \tag{5.18}$$

and $f_{j,i}$ is the individual flow rate of component i in the feed stream (if any) to stage j.

This system combines the MES (Mass balance, Equilibrium, and mole fraction Summation) equations. When the activity coefficient is not modeled as a separate factor in the K-value expression, i.e., when Equation 5.2 is used instead of Equation 5.4, then the above equations constitute a linear tridiagonal system that can be solved explicitly with the Thomas algorithm. If Equation 5.4 is used, then the MES system is nonlinear and must be solved iteratively.

With the component liquid flow rates $\mathbf{l}$ and with $\mathbf{v}_{\text{inner}}$ and $\mathbf{v}_{\text{outer}}$ we are able to compute all primitive variables $\mathbf{X}$ explicitly according to the following sequence of equations:

$$L_j = \sum_i l_{j,i}, \tag{5.19}$$

$$x_{j,i} = \frac{l_{j,i}}{L_j}, \tag{5.20}$$

$$\alpha^*_{j,i} = \begin{cases} \alpha_{j,i}, & \text{if Equation 5.2 is used;} \\ \alpha_{j,i} \exp\left(a_{j,i} + b_{j,i} x_{j,i}\right), & \text{if Equation 5.4 is used,} \end{cases} \tag{5.21}$$

$$K_{b,j} = \frac{1}{\sum_i \alpha^*_{j,i} x_{j,i}}, \tag{5.22}$$

$$V_j = \frac{S_j L_j}{K_{b,j}}, \tag{5.23}$$

$$y_{j,i} = \alpha^*_{j,i} K_{b,j} x_{j,i}, \tag{5.24}$$

$$T_j = \frac{B_j}{\ln K_{b,j} - A_j}, \tag{5.25}$$

$$W_{L_j} = R_{L_j} L_j, \tag{5.26}$$

$$W_{V_j} = R_{V_j} V_j. \tag{5.27}$$

Note that Equation 5.22 for computing $K_{b,j}$ derives from the “bubble point equation” $\sum_i K_{j,i} x_{j,i} = 1$.

With the updated primitive variables $\mathbf{X}$ we can evaluate $\mathbf{f}_{\text{inner}}(\mathbf{v}_{\text{inner}}, \mathbf{v}_{\text{outer}})$, which corresponds to the residuals of the energy balances (i.e., H equations), of the two main column specifications, and of any side product specifications. The H equations follow the standard MESH model, except that the vapor and liquid phase molar enthalpies are expressed using the simplified models:

$$h^L_{j-1} L_{j-1} + h^V_{j+1} V_{j+1} + h^F_j F_j - h^L_j (L_j + W_{L,j}) - h^V_j (V_j + W_{V,j}) + Q_j = 0, \tag{5.28}$$

$$h^V_j = \sum_i y_{j,i} H^{\text{ig}}_i(T_j) + \left(\sum_i y_{j,i} MM_i\right) \Big(C_j + D_j(T_j - T^*_j)\Big), \tag{5.29}$$

$$h_j^L = \sum_i x_{j,i} H_i^{\text{ig}}(T_j) + \left(\sum_i x_{j,i} MM_i \right) \Big(E_j + F_j (T_j - T_j^*) \Big). \tag{5.30}$$

In general the H equations are only enforced for stages $j = 2, \ldots, N-1$ and the energy balances for stages $j = 1, N$ are substituted with the two main column specifications $\lambda_1 - \lambda_{1,\text{spec}} = 0$ and $\lambda_2 - \lambda_{2,\text{spec}} = 0$, unless the latter involve the condenser and/or reboiler heat duties. Together, these give N residuals that match up the N inner loop variables S_j. Since we consider the condenser (Stage 1) to have a liquid side product of flow rate W_{L_1}, $R_{L,1}$ is always also present as an inner loop variable and thus we include the residual of the equation

$$\theta - \frac{V_1}{V_1 + W_{L,1}} = 0. \tag{5.31}$$

Finally, if any side products are present at intermediate stages we have the corresponding $R_{L,j}, R_{V,j}$ as inner loop variables, and thus we must include the same number of specification equations for the latter within $\mathbf{f}_{\text{inner}}(\mathbf{v}_{\text{inner}}, \mathbf{v}_{\text{outer}})$. For example, we would include the residual of the equation $R_{L,j} - R_{L,j,\text{spec}} = 0$ to specify a withdrawal ratio $R_{L,j,\text{spec}}$ for the liquid side product of stage j. This way, the resulting inner loop system $\mathbf{f}_{\text{inner}}(\mathbf{v}_{\text{inner}}, \mathbf{v}_{\text{outer}}) = \mathbf{0}$ is square.

5.2.6. The outer loop function

For a given vector $\mathbf{v}_{\text{outer}}$, the outer loop function value $\mathbf{f}_{\text{outer}}(\mathbf{v}_{\text{outer}})$ is computed as follows. First, we solve the inner loop $\mathbf{f}_{\text{inner}}(\mathbf{v}_{\text{inner}}, \mathbf{v}_{\text{outer}}) = \mathbf{0}$ for $\mathbf{v}_{\text{inner}}$. We can compute the set of primitive variables $\mathbf{X}$ from the converged $\mathbf{v}_{\text{inner}}$ values through Equations 5.15 to 5.27, as described in Section 5.2.5. Given $\mathbf{X}$, we obtain new values for the simplified model parameters $\mathbf{A}$ to $\mathbf{F}$, $\mathbf{a}, \mathbf{b}$ together with the relative volatilities $\boldsymbol{\alpha}$ according to the following procedure for each stage j:

1. Evaluate ΔH_j^V, ΔH_j^L and the $K_{j,i}$ values with rigorous thermodynamic models at the current $(T_j, P_j, \mathbf{x}_j, \mathbf{y}_j)$ values and also at $(T_j + \epsilon, P_j, \mathbf{x}_j, \mathbf{y}_j)$, where $\epsilon > 0$ is a

small temperature perturbation. Use the pairs of values for the two temperatures to update the coefficients $A_j, B_j, C_j, D_j, E_j, F_j$ according to Equations 5.7 to 5.9.

2. Update the $t_{j,i}$ coefficients using finite differencing:

$$t_{j,i} = y_{j,i}\left(\frac{\ln K_{j,i}(T_j+\epsilon) - \ln K_{j,i}(T_j+\epsilon) - \ln K_i(T_j)}{1/(T_j+\epsilon) - 1/T_j}\right). \tag{5.32}$$

3. Compute $\ln K_{b,j}(T_j)$ and $\ln K_{b,j}(T_j+\epsilon)$ using Equation 5.3 with the updated $t_{j,i}$.

4. If the activity coefficient is modeled as a separate factor, evaluate $\gamma_{j,i}$ rigorously at the current $(T_j, P_j, \mathbf{x}_j, \mathbf{y}_j)$ values, then perturb only the mole fraction $x_{j,i}$ and reevaluate $\gamma_{j,i}$ for each i. Use the pairs of $\gamma_{j,i}$ values to update $a_{j,i}, b_{j,i}$ according to Equation 5.10.

5. Update the $\alpha_{j,i}$ values using Equation 5.2 or 5.4.

The outer loop function value corresponds to the updated parameters:

$$\mathbf{f}_{\text{outer}}(\mathbf{v}_{\text{outer}}) = (\boldsymbol{\alpha}, \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}, \mathbf{F}, \mathbf{a}, \mathbf{b}). \tag{5.33}$$

5.3. The proposed nonsmooth inside-out algorithm

In this section we propose a nonsmooth version of the inside-out algorithm that can be used to successfully converge both our single-soft and double-soft adaptive models from Chapter 4. The algorithm retains the same structure described in Section 5.2 except for the modifications that we now present. Moreover, we use the initialization procedures from Section 4.3 to obtain $\mathbf{X}^0$.

5.3.1. Inner loop modifications

Inner loop variables

Even though dry/vaporless solutions in Phase Regimes II are feasible within the standard MESH model, the inside-out algorithms of both Boston and Sullivan Jr and Russell cannot converge to these solutions due to their choice of inner loop variables. Firstly, from Equation 5.14 we see that the stripping factor S_j is undefined for a dry stage j. Moreover, since the inner loop manipulates the variables $\ln(S_j/S_{b,j})$, the algorithm does not allow for vaporless solutions either ($S_j = 0$). Another issue is that the inner loop variable $R_{L,1} = W_{L_1}/L_1$ is undefined when the reflux rate (or ratio) is equal to zero, which is a situation often encountered when using our adaptive models from Chapter 4.

Therefore, in our inner loop we replace the stripping factors with the "weighted" vapor fractions

$$\phi_j = \frac{K_{b,j}V_j}{K_{b,j}V_j + K_{0,j}L_j}, \quad 2 \leq j \leq N, \tag{5.34}$$

which become our main inner loop variables. These are analogous to the inner loop variables for the single-stage flash inside-out algorithm of Boston and Britt [13]. In the above equation, $K_{0,j}$ is a constant scaling factor that we set to be equal to our initial guess for $K_{b,j}$ when starting the calculations. Moreover, we eliminate $R_{L,1} = W_{L_1}/L_1$ as an inner loop variable and remove the corresponding Equation 5.31 from the inner loop residual by incorporating it into the MES equation for a partial condenser (see Equation 5.37). As a result, the inner loop variable ϕ_1 for Stage 1 is defined as follows:

$$\phi_1 = \begin{cases} \dfrac{W_{L_1}}{W_{L_1} + L_1}, & \theta = 0; \\ \dfrac{K_{b,1}V_1}{K_{b,1}V_1 + K_{0,1}L_1}, & 0 < \theta \leq 1. \end{cases} \tag{5.35}$$

With these changes to the inner loop variables, our algorithm can handle zero and negative flow rates in intermediate iterations without taking logarithms, and it can converge to dry/vaporless solutions in Phase Regime II as required by our adaptive models.

When side products at intermediate stages are present, we must retain the corresponding $R_{L,j} = W_{L,j}/L_j$ and $R_{V,j} = W_{V,j}/V_j$ as inner loop variables to preserve the structure of the inner loop. As a result, side product flow rate specifications $W_{L,j} = W_{L,j,\text{spec}}$ or $W_{V,j} = W_{V,j,\text{spec}}$ cannot be enforced by the algorithm for dry/vaporless stages, though withdrawal ratio specifications $R_{L,j} = R_{L,j,\text{spec}}$ or $R_{V,j} = R_{V,j,\text{spec}}$ can still be used. The latter are only enforced at the solution if the stage j is not dry neither vaporless; otherwise, we obtain $W_{L,j} = 0$ or $W_{V,j} = 0$ (see Equations 5.51 and 5.52).

To be able to specify side product flow rates, we can alternatively use a "hybrid" strategy between the inside-out algorithm and simultaneous convergence of the adaptive model equations. That is, we retain the outer loop and modify the inner loop so that the latter consists of the full set of model equations written in terms of the simplified thermodynamic property models. This same type of strategy is used in [82] for reactive distillation simulation, since in that case the original inner loop structure cannot be maintained either.

MES equations

The component liquid flow rates $l_{j,i}$ are all equal to zero for a dry stage j. In this case, the liquid mole fractions $x_{j,i}$ cannot be computed using Equation 5.20 within the inner loop. Therefore, we formulate our MES equation system in terms of the variables $p_{j,i}$, which are defined through the relationship

$$l_{j,i} = (1 - \phi_j)\, p_{j,i}, \tag{5.36}$$

instead of the component liquid flow rates $l_{j,i}$. This way, for a dry stage j (for which $\phi_j = 1$) we may have $p_{j,i} \neq 0$ despite the fact that $l_{j,i} = 0$. The modified MES equations for each stage in terms of the $\mathbf{p}$ variables are as follows:

- Stage $j = 1$ (condenser):

$$\begin{cases} p_{1,i} - \left[\alpha^*_{2,i} K_{0,2} \phi_2\right] p_{2,i} = f_{1,i} & \theta = 0; \\ \left[1 - \phi_1 + \left(\alpha^*_{1,i} + \frac{1-\theta}{\theta K_{b,1}}\right) K_{0,1} \phi_1\right] p_{1,i} - \left[\alpha^*_{2,i} K_{0,2} \phi_2\right] p_{2,i} = f_{1,i}, & 0 < \theta \leq 1. \end{cases} \tag{5.37}$$

- Intermediary stages ($2 \leq j \leq N - 1$):

$$\begin{aligned} \left[\phi_{j-1} - 1\right] p_{j-1,i} + \left[(1 - \phi_j)(1 + R_{L,j}) + \alpha^*_{j,i} K_{0,j} \phi_j (1 + R_{V,j})\right] p_{j,i} \\ - \left[\alpha^*_{j+1,i} K_{0,j+1} \phi_{j+1}\right] p_{j+1,i} = f_{j,i} \end{aligned} \tag{5.38}$$

- Stage $j = N$ (reboiler):

$$\left[\phi_{N-1} - 1\right] p_{N-1,i} + \left[1 - \phi_N + \alpha^*_{N,i} K_{0,N} \phi_N (1 + R_{V,N})\right] p_{N,i} = f_{N,i}. \tag{5.39}$$

Here,

$$\alpha^*_{j,i} = \begin{cases} \alpha_{j,i}, & \text{if Equation 5.2 is used;} \\ \alpha_{j,i} \exp\left(a_{j,i} + b_{j,i} \frac{p_{j,i}}{\sum_k p_{j,k}}\right), & \text{if Equation 5.4 is used.} \end{cases} \tag{5.40}$$

For a partial condenser ($0 < \theta \leq 1$), we see from Equation 5.37 that the MES equation for Stage 1 depends on $K_{b,1}$. Since the MES equations can only be formulated in terms of $\mathbf{v}_{\text{inner}}$, $\mathbf{v}_{\text{outer}}$ and $\mathbf{p}$, in this case we include $K_{b,1}$ as an extra outer loop variable. We treat it as a constant and simply update it at each outer loop iteration.

As detailed later on in this section, to solve our nonsmooth inner loop we must be able to compute directional derivatives of $\mathbf{f}_{\text{inner}}$ with respect to $\mathbf{v}_{\text{inner}}$. When we choose to treat $\gamma_{j,i}$ as a separate factor through Equation 5.4, the MES equations are a smooth nonlinear system which we solve with `fsolve` within our implementation. In this case, since $\mathbf{p}$ is solved for iteratively, we cannot apply automatic differentation techniques directly to obtain the directional derivatives of $\mathbf{p}$ with respect to $\mathbf{v}_{\text{inner}}$. Instead, we compute the latter analytically using the smooth Implicit Function Theorem; that is, in terms of partial Jacobian matrices and in simplified notation,

$$\mathbf{J}_{\mathbf{v}_{\text{inner}}}\mathbf{p} = -\left(\mathbf{J}_{\mathbf{p}}\mathbf{f}_{\text{MES}}\right)^{-1}\left(\mathbf{J}_{\mathbf{v}_{\text{inner}}}\mathbf{f}_{\text{MES}}\right), \tag{5.41}$$

where $\mathbf{f}_{\text{MES}}$ is the residual function for the MES equations, and the individual Jacobian matrices on the right-hand side can be obtained with automatic differentiation. Another point to consider is that the nonlinear MES equation system can be hard to converge for highly non-ideal systems without a good initial guess for $\mathbf{p}$. To improve speed and reliability of convergence, we supply the converged $\mathbf{p}$ values from the previous inner loop iteration as an initial guess.

To compute the primitive variables $\mathbf{X}$ from the converged $\mathbf{p}$ values, we use the following sequence of equations instead of Equations 5.19 to 5.27:

$$L_j = \sum_i l_{j,i} = (1-\phi_j)\sum_i p_{j,i} \tag{5.42}$$

$$x_{j,i} = \frac{l_{j,i}}{L_j} = \frac{p_{j,i}}{\sum_i p_{j,i}} \tag{5.43}$$

$$\alpha^*_{j,i} = \begin{cases} \alpha_{j,i}, & \text{if Equation 5.2 is used;} \\ \alpha_{j,i}\exp\left(a_{j,i} + b_{j,i}x_{j,i}\right), & \text{if Equation 5.4 is used,} \end{cases} \tag{5.44}$$

$$K_{b,j} = \frac{1}{\sum_i \alpha^*_{j,i}x_{j,i}} \tag{5.45}$$

$$T_j = \frac{B_j}{\ln K_{b,j} - A_j} \tag{5.46}$$

$$V_1 = \begin{cases} 0, & \theta = 0; \\ \dfrac{K_{0,1}\phi_1}{K_{b,1}}\displaystyle\sum_i p_{1,i}, & 0 < \theta \leq 1 \end{cases} \tag{5.47}$$

$$V_j = \frac{K_{0,j}\phi_j}{K_{b,j}}\sum_i p_{j,i}, \quad 2 \leq j \leq N \tag{5.48}$$

$$y_{j,i} = \alpha^*_{j,i} K_{b,j} x_{j,i} \tag{5.49}$$

$$W_{L_1} = \begin{cases} \phi_1 \sum_i p_{1,i}, & \theta = 0; \\ \dfrac{1-\theta}{\theta} V_1, & 0 < \theta \leq 1. \end{cases} \tag{5.50}$$

$$W_{L_j} = R_{L_j} L_j, \quad j \neq 1 \tag{5.51}$$

$$W_{V_j} = R_{V_j} V_j \tag{5.52}$$

Nonsmooth specification equations

We replace one or both of the two main column specifications $\lambda_1 - \lambda_{1,\text{spec}} = 0$, $\lambda_2 - \lambda_{2,\text{spec}} = 0$ within $\mathbf{f}_{\text{inner}}$ with the nonsmooth specification equation(s) presented in Chapter 4 for the single-soft and double-soft adaptive models, respectively.

Equation solving methods

Despite its smaller size compared to the full-size adaptive models from Chapter 4, the inner loop system $\mathbf{f}_{\text{inner}}(\mathbf{v}_{\text{inner}}) = \mathbf{0}$ is nonsmooth (in particular, PC^∞) in the same way as the latter, given that it contains the same nonsmooth specification equations(s). Therefore, we solve the inner loop using the same method described in Section 4.4, which employs the (pseudoinverse) semismooth Newton step together with a forward tracking line search. The only difference is in the function $\alpha_{\min}$, which now imposes a threshold of $\Delta\phi = 0.05$ for the magnitude of variation in the ϕ_j inner loop variables.

5.3.2. Outer loop modifications

Activity coefficient simplified model

The simplified model for each activity coefficient $\gamma_{j,i}$ presented in Equation 5.10, as proposed by Boston [15], models $\ln(\gamma_{j,i})$ as an affine function only of $x_{j,i}$. That is, the effects of varying the mole fractions of components other than i are not taken into account. We have found that this strategy precludes the outer loop from converging for some non-ideal systems, including the diethylamine-methanol and the five-component azeotropic systems from Chapter 4 that we will revisit in this chapter. This is the case, in particular, for solutions that are near an infinite flow rate pinch point.

Therefore, for highly non-ideal systems we propose a simplified activity model of the form

$$\ln \gamma_{j,i} = a_{j,i} + b_{j,i}x_{j,i} + c_{j,i}x_{j,k} + \dots \tag{5.53}$$

That is, to model the activity coefficient $\gamma_{j,i}$ of component i we include linear terms not only for $x_{j,i}$ but also for mole fractions $x_{j,k}$ of all other components $k \neq i$. This way, the activity coefficient model accounts for $N_c + 1$ parameter matrices $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \dots \in \mathbb{R}^{N \times N_c}$ among the outer loop variables instead of only two matrices $\mathbf{a}, \mathbf{b}$. In the specific case of a binary mixture, however, we can avoid including an extra parameter matrix $\mathbf{c}$ by perturbing both $x_{j,1}$ and $x_{j,2}$ simultaneously, in opposite amounts, when updating the coefficients $\mathbf{a}$ and $\mathbf{b}$ (see Section 5.2.6).

Equation solving methods

Once $\mathbf{v}_{\text{outer}}^{k+1}$ is computed using a chosen fixed-point method at a given outer loop iteration k, it might not be possible to evaluate $\mathbf{f}_{\text{outer}}(\mathbf{v}_{\text{outer}}^{k+1})$ due to convergence failure of the inner loop. We have observed this type of problem even when applying pure direct substitution to especially non-ideal liquid mixtures, such as the diethylamine-methanol azeotropic system, near an infinite flow rate pinch point. Moreover, solving the inner loop within the outer loop function evaluation is a nonsmooth equation-solving task in our method, and thus $\mathbf{f}_{\text{outer}}$ is implicitly nonsmooth.

Therefore, we use the Anderson acceleration algorithm of Zhang et al. [91] (see Section 2.3.6) for nonsmooth fixed-point problems to converge our outer loop, together with a "try-catch" approach to promote some measure of step-size control. Here, we will temporarily denote the outer loop fixed-point problem as $\mathbf{f}(\mathbf{x}) = \mathbf{x}$ for simplicity of notation. For choosing a step size, one could initially consider a standard line search procedure in the direction $\mathbf{d}^k = \mathbf{x}^{k+1} - \mathbf{x}^k$ and thus select a value for $\alpha \in (0, 1)$ that minimizes the error

$\left\|\mathbf{f}(\mathbf{x}^k + \alpha\mathbf{d}^k) - (\mathbf{x}^k + \alpha\mathbf{d}^k)\right\|$ and that also allows for $\mathbf{f}$ to be evaluated. However, each evaluation of the outer loop function is quite costly because it requires convergence of the inner loop. Instead, we opt for a "try-catch" approach: we reduce the step size from $\alpha = 1$ to some $\alpha^* \in (0, 1)$ only in case we fail to evaluate $\mathbf{f}(\mathbf{x}^{k+1})$. Moreover, we continue to use the reduced step size α^* until we are able to successfully evaluate $\mathbf{f}(\mathbf{x}^{k+1})$ and reduce the error value for 3 consecutive outer loop iterations, as summarized in Algorithm 2. The value $\alpha^* = 0.3$ was found to be suitable for our test cases.

Additionally, we found that we can predict with fair certainty if the inner loop will fail to converge based on the initial value of the inner loop residual. Specifically, if the initial inner loop residual for the outer iteration k is sufficiently larger than that of the previous outer loop iteration $k - 1$, then we conclude the outer loop step size is too large for the inner loop to converge. This way, we can avoid unnecessary extra computations.

With particularly non-ideal liquid mixtures, for which we must treat $\gamma_{j,i}$ as a separate factor through Equation 5.4, we might fail to converge the inner loop while trying to evaluate $\mathbf{f}_{\text{outer}}$ at the initial guess $\mathbf{v}^0_{\text{outer}}$. In such cases, first we converge the inner loop without the separate $\gamma_{j,i}$ factor (i.e., using Equation 5.2). Then, we use the converged $\mathbf{v}_{\text{inner}}$ value to obtain an updated guess for $\mathbf{v}^0_{\text{outer}}$ following the procedure in Section 5.2.6 and reattempt to evaluate $\mathbf{f}_{\text{outer}}(\mathbf{v}^0_{\text{outer}})$ with the full $\gamma_{j,i}$ model. This strategy proved to be quite effective in converging our most problematic test cases involving azeotropic systems.

Lastly, as done in [87] for the nonsmooth flash inside-out algorithm, we choose to update the "slope" coefficients B_j, D_j, F_j at each outer loop iteration.

Algorithm 2: Outer loop $\mathbf{x} = \mathbf{f}(\mathbf{x})$ of the nonsmooth inside-out distillation method.

```
Input  : $\mathbf{x}^0$, $\epsilon_{\text{outer}}$, $k_{\max}$, $\mathbf{f}$, $\alpha^*$.
Input  : $\mathbf{g}$ (update function of the chosen fixed-point method).
Output: $\mathbf{x}^k$

$\mathbf{f}^0 \leftarrow \mathbf{f}(\mathbf{x}^0)$
$k \leftarrow 0$
flag $\leftarrow 0$
counter $\leftarrow 0$
while $k < k_{max}$ and $\left\|\mathbf{f}^k - \mathbf{x}^k\right\| > \epsilon_{\text{outer}}$ do
    $\mathbf{x}^{k+1} \leftarrow \mathbf{g}(\mathbf{x}^k, \mathbf{x}^{k-1}, \dots, \mathbf{f}^k, \mathbf{f}^{k-1}, \dots)$
    if flag $= 1$ then
        $\mathbf{x}^{k+1} \leftarrow \alpha^* \mathbf{x}^{k+1} + (1 - \alpha^*) \mathbf{x}^k$
    end
    try:
        $\mathbf{f}^{k+1} \leftarrow \mathbf{f}(\mathbf{x}^{k+1})$
        if flag $= 1$ and $\left\|\mathbf{f}^{k+1} - \mathbf{x}^{k+1}\right\| < \left\|\mathbf{f}^k - \mathbf{x}^k\right\|$ then
            counter $\leftarrow$ counter $+ 1$
            if counter $= 3$ then
                flag $\leftarrow 0$
            end
        end
    catch Failure in computing $\mathbf{f}(\mathbf{x}^{k+1})$:
        $\mathbf{x}^{k+1} \leftarrow \alpha^* \mathbf{x}^{k+1} + (1 - \alpha^*) \mathbf{x}^k$
        try:
            $\mathbf{f}^{k+1} \leftarrow \mathbf{f}(\mathbf{x}^{k+1})$
            counter $\leftarrow 0$
            flag $\leftarrow 1$
        catch Failure in computing $\mathbf{f}(\mathbf{x}^{k+1})$:
            return $\mathbf{x}^0$
        end
    end
    $k \leftarrow k + 1$
end
return $\mathbf{x}^k$
```

5.4. Single-column simulation test cases

We now compare the performance of three different simulation methods for single-column simulation (i.e., all feed streams are fully specified and the column is not part of an overarching flowsheet):

- **Method 1:** the single-soft adaptive model converged with the so-called simultaneous algorithm (i.e., we solve the model equations described in Section 4.2.1 simultaneously);

- **Method 2:** the single-soft adaptive model converged with the nonsmooth inside-out algorithm developed in Section 5.3;

- **Method 3:** the RadFrac model in Aspen Plus converged with its standard algorithm, which is based on Boston and Sullivan Jr's [14] inside-out method.

5.4.1. Simultaneous versus inside-out algorithms for the single-soft adaptive model

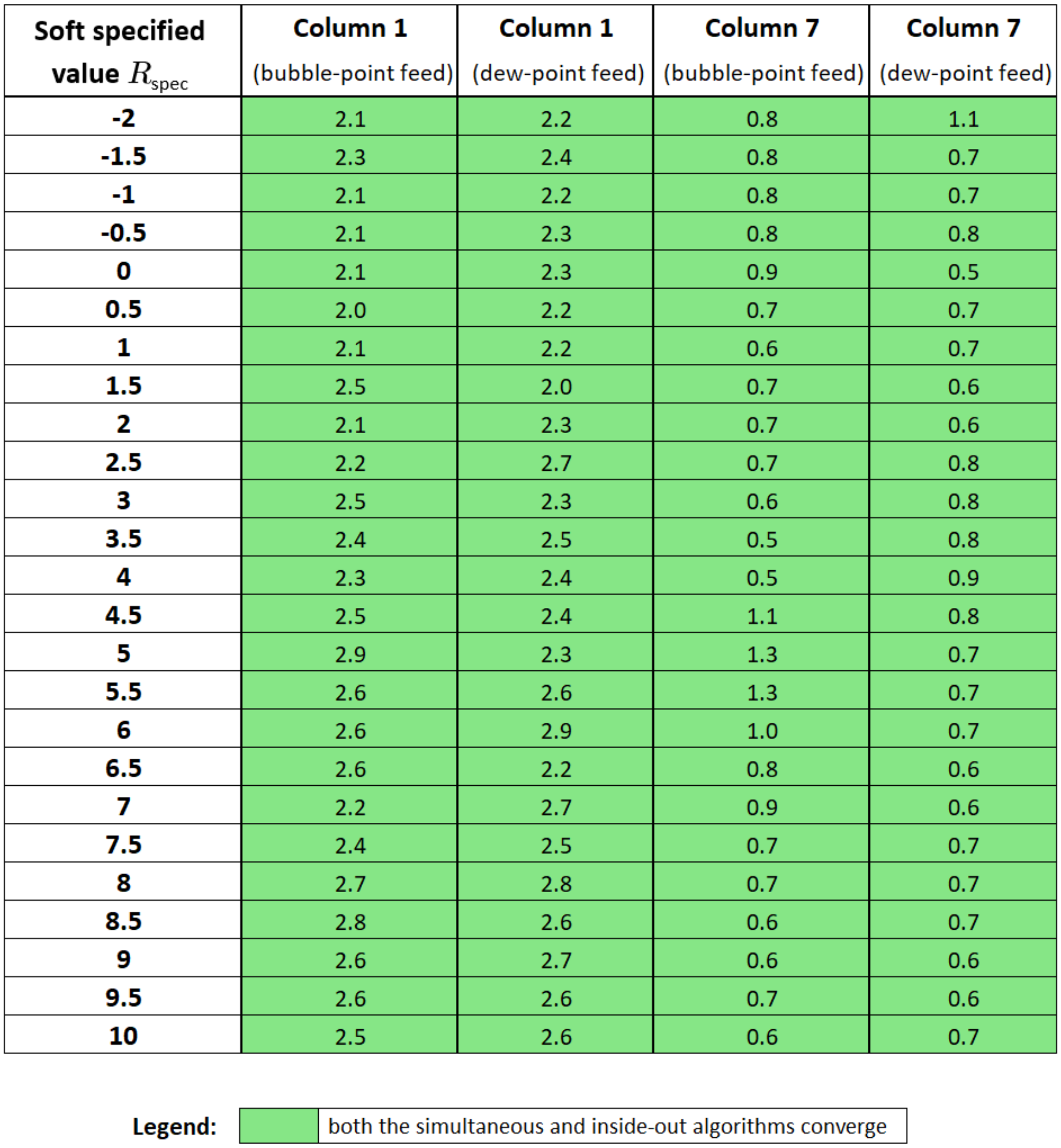
In this section we will compare Methods 1 and 2 using several test cases based on the seven columns of Table 4.2 from Chapter 4. We use the same tolerance value of $\epsilon_{\text{inner}} = 10^{-7}$ for the infinity norm of the inner loop residual and of the full single-soft adaptive model residual, and $\epsilon_{\text{outer}} = 10^{-3}$ for the infinity norm of the outer loop residual. The simultaneous algorithm and the inner loop of the inside-out algorithm are solved with the same equation-solving method from Section 4.4. The outer loop is converged with Anderson acceleration according to Section 5.3.2. In each test case we use the initial guess $\mathbf{X}^0$ obtained according to Section 4.3 for both algorithms, i.e., a "blind" initial guess that does not rely on results of other simulations. We terminate both the simultaneous algorithm and the inner loop after a maximum of 45 iterations, and the outer loop after a maximum of 25 iterations (which is the same default limit in RadFrac).

Our models and equation solving algorithms were coded in MATLAB. We retrieved parameter values for all thermodynamic property correlations from the Aspen Plus V10 database. We used $r_{\text{max}} = 5$ and $r_{\text{min}} = 0.05$ for all test cases except for those involving Column 7, for which we used $r_{\text{max}} = 10$ (see discussion in Section 4.5.5). The fact that the chosen r_{min} value is greater than zero only affects the test cases of Columns 5 and 6, since $W_{L,1}$ is the first flow rate to become zero in both systems instead of an internal flow rate (see Section 4.5.4).

We will use two main metrics of performance: convergence reliability (i.e., the ability to converge to the correct/expected solution from a "blind" initial guess within the imposed iteration limit), and convergence speed (i.e., total running time until convergence). Our MATLAB implementation of the simultaneous and inside-out algorithms has not been optimized for speed, as that would have been outside of the scope of this thesis. Therefore, to compare convergence speed we report only the ratio of running times between the inside-out and the simultaneous algorithms.

Figures 5.1, 5.2, 5.3 and 5.4 present the results of 584 test case simulations involving Columns 1 through 7 from Table 4.2, using both the simultaneous and inside-out algorithms to converge the single-soft model. In Figures 5.1 and 5.2 we vary the soft reflux ratio specification from -2 to 10 in 0.5 increments for Columns 1 and 7, each of which is given either a bubble-point or a dew-point feed stream. In Figures 5.3 and 5.4 we vary the soft product purity specification (which is either $x_{1,1}$ or $x_{N,1}$, depending on the column) from 0 to 1 in 0.1 increments for Columns 2 through 6, using several different feed compositions for each of the columns. The chosen hard specification for each column is stated in Table 4.2. Figures 5.1 and 5.3 depict the test cases in which each algorithm converges, and present the ratio of simulation times for the inside-out/simultaneous algorithms for those test cases in which both of them converge. Figures 5.2 and 5.4 show the value of the soft specification variable λ (either R or a product purity) obtained at the single-soft model solution for each λ_{spec} value, highlighting the corresponding maximum ($\lambda = \lambda_{r_{\text{max}}}$)

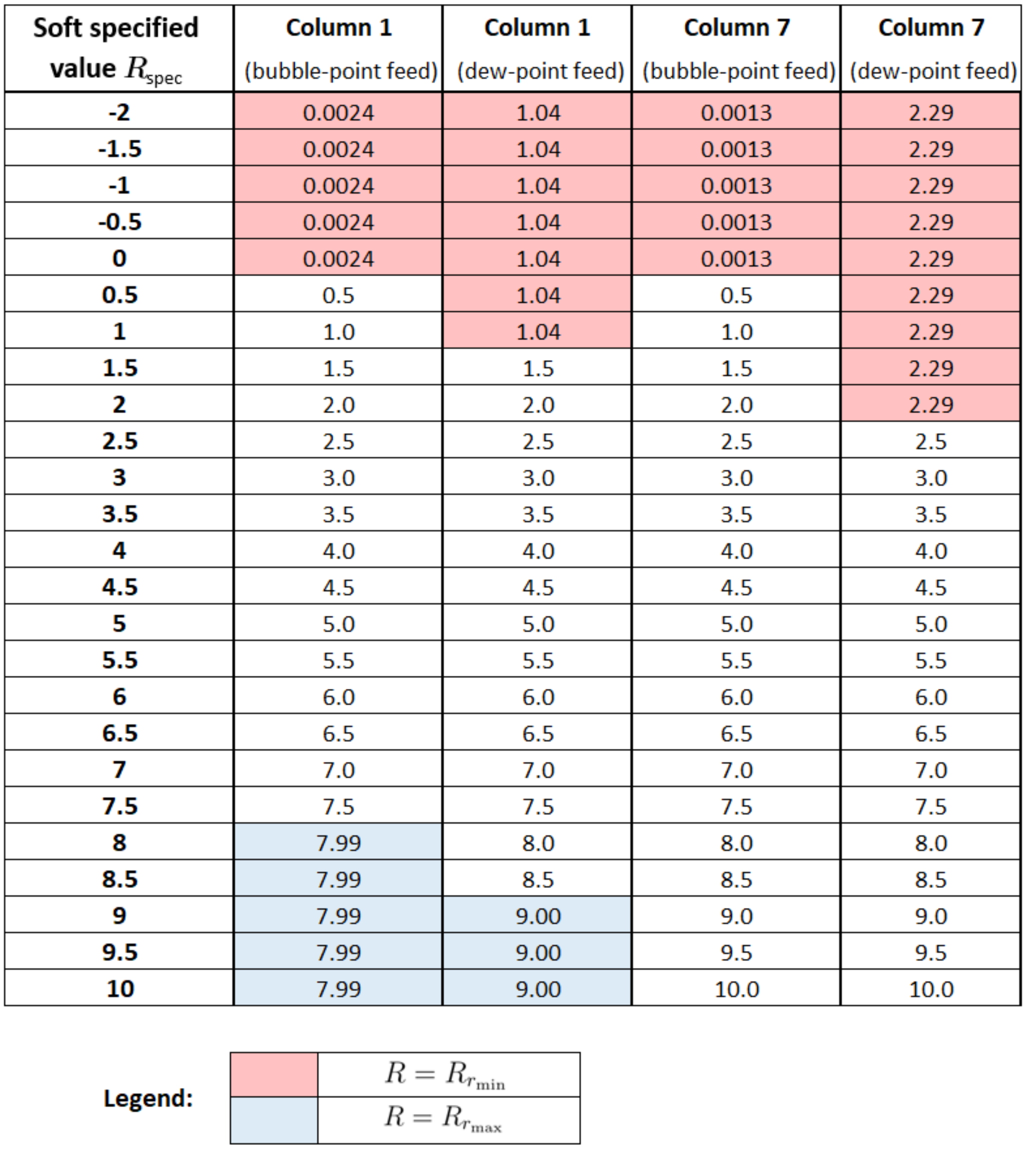
and minimum ($\lambda = \lambda_{r_{\min}}$) flow rate solutions (see Section 4.2.1).



Soft specified value R_{spec}	**Column 1** (bubble-point feed)	**Column 1** (dew-point feed)	**Column 7** (bubble-point feed)	**Column 7** (dew-point feed)
-2	2.1	2.2	0.8	1.1
-1.5	2.3	2.4	0.8	0.7
-1	2.1	2.2	0.8	0.7
-0.5	2.1	2.3	0.8	0.8
0	2.1	2.3	0.9	0.5
0.5	2.0	2.2	0.7	0.7
1	2.1	2.2	0.6	0.7
1.5	2.5	2.0	0.7	0.6
2	2.1	2.3	0.7	0.6
2.5	2.2	2.7	0.7	0.8
3	2.5	2.3	0.6	0.8
3.5	2.4	2.5	0.5	0.8
4	2.3	2.4	0.5	0.9
4.5	2.5	2.4	1.1	0.8
5	2.9	2.3	1.3	0.7
5.5	2.6	2.6	1.3	0.7
6	2.6	2.9	1.0	0.7
6.5	2.6	2.2	0.8	0.6
7	2.2	2.7	0.9	0.6
7.5	2.4	2.5	0.7	0.7
8	2.7	2.8	0.7	0.7
8.5	2.8	2.6	0.6	0.7
9	2.6	2.7	0.6	0.6
9.5	2.6	2.6	0.7	0.6
10	2.5	2.6	0.6	0.7

Legend: 🟩 both the simultaneous and inside-out algorithms converge

Figure 5.1: Ratio of simulation times for the inside-out/simultaneous algorithms for each test case with a soft R specification.



Soft specified value R_{spec}	Column 1 (bubble-point feed)	Column 1 (dew-point feed)	Column 7 (bubble-point feed)	Column 7 (dew-point feed)
-2	🟥 0.0024	🟥 1.04	🟥 0.0013	🟥 2.29
-1.5	🟥 0.0024	🟥 1.04	🟥 0.0013	🟥 2.29
-1	🟥 0.0024	🟥 1.04	🟥 0.0013	🟥 2.29
-0.5	🟥 0.0024	🟥 1.04	🟥 0.0013	🟥 2.29
0	🟥 0.0024	🟥 1.04	🟥 0.0013	🟥 2.29
0.5	0.5	🟥 1.04	0.5	🟥 2.29
1	1.0	🟥 1.04	1.0	🟥 2.29
1.5	1.5	1.5	1.5	🟥 2.29
2	2.0	2.0	2.0	🟥 2.29
2.5	2.5	2.5	2.5	2.5
3	3.0	3.0	3.0	3.0
3.5	3.5	3.5	3.5	3.5
4	4.0	4.0	4.0	4.0
4.5	4.5	4.5	4.5	4.5
5	5.0	5.0	5.0	5.0
5.5	5.5	5.5	5.5	5.5
6	6.0	6.0	6.0	6.0
6.5	6.5	6.5	6.5	6.5
7	7.0	7.0	7.0	7.0
7.5	7.5	7.5	7.5	7.5
8	🟦 7.99	8.0	8.0	8.0
8.5	🟦 7.99	8.5	8.5	8.5
9	🟦 7.99	🟦 9.00	9.0	9.0
9.5	🟦 7.99	🟦 9.00	9.5	9.5
10	🟦 7.99	🟦 9.00	10.0	10.0

Legend:

🟥	$R = R_{r_{\min}}$
🟦	$R = R_{r_{\max}}$

Figure 5.2: Value of R at the single-soft model solution for each test case with a soft R specification.

		Soft product purity specified value λ_{spec}										
	z_1 value	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Column 2 (bubble-point feed)	0.35	2.0	2.2	2.1	2.1	1.5	2.1	2.0	1.9	2.6	🟦	🟦
	0.45	1.7	2.2	1.9	2.2	1.9	1.7	2.0	2.3	2.1	🟦	🟦
	0.55	2.1	2.3	2.4	2.4	2.6	2.4	1.8	1.7	2.0	1.2	1.3
	0.65	1.7	1.8	2.0	2.0	2.0	1.8	1.6	2.2	2.1	1.5	1.2
	0.75	1.8	2.0	2.1	2.1	2.2	1.8	1.7	1.5	🟦	1.5	1.7
	0.85	2.1	2.1	2.0	2.3	1.9	2.1	2.1	2.2	2.0	1.8	0.9
	0.9	1.7	1.6	1.8	1.9	1.7	1.8	1.6	1.6	1.9	1.6	1.3
Column 2 (dew-point feed)	0.35	2.4	2.3	2.2	2.6	1.5	1.7	🟦	1.8	2.5	1.4	1.6
	0.45	2.3	1.9	1.9	2.2	1.9	🟦	2.5	1.8	2.3	1.6	1.4
	0.55	1.6	1.5	1.5	1.7	1.5	1.8	1.4	2.5	2.5	1.4	1.3
	0.65	0.9	1.0	0.9	0.9	1.0	0.9	0.9	1.8	2.4	1.8	1.5
	0.75	2.3	2.0	2.0	2.3	1.9	2.3	2.0	2.2	1.9	1.0	1.2
	0.85	2.2	2.3	1.9	2.5	2.1	2.4	2.2	2.0	2.4	2.1	1.0
	0.9	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	0.7
Column 3	0.4	2.2	3.5	3.7	3.7	2.0	2.3	1.9	1.8	1.8	1.6	1.7
	0.45	3.3	3.5	4.2	3.7	🟦	1.6	1.4	1.3	1.3	1.4	1.4
	0.5	3.6	4.0	3.6	4.0	🟦	1.7	1.7	1.7	1.5	1.7	1.6
	0.55	4.0	3.8	4.1	3.9	🟦	1.3	🟦	🟦	🟦	🟦	🟦
	0.6	4.0	3.7	3.7	3.6	2.0	1.4	1.2	1.2	1.3	1.3	1.5
	0.65	3.7	4.0	3.9	3.9	1.7	1.7	1.2	1.1	1.3	1.2	1.1
	0.7	4.1	4.3	3.6	3.6	2.0	1.7	1.8	🟦	1.3	2.0	1.9
Column 4	0.2	2.2	2.3	1.2	1.9	2.2	2.1	2.3	2.1	1.3	2.2	1.9
	0.25	2.2	2.3	2.2	2.3	🟦	2.1	1.9	2.0	1.3	2.0	2.0
	0.3	1.9	2.3	2.0	🟦	0.6	1.8	2.0	2.0	1.3	1.9	1.9
	0.35	0.3	0.3	0.3	0.3	🟦	2.5	2.2	2.2	1.3	2.3	2.1
	0.4	🟦	🟦	🟦	🟦	🟦	2.4	2.1	2.1	1.3	2.3	2.1
	0.42	2.4	2.0	2.1	2.4	2.1	🟦	🟦	🟦	1.3	🟦	🟦
Column 5	0.32	🟦	🟦	🟦	🟦	4.0	4.0	3.9	3.8	1.3	4.0	3.9
	0.35	1.4	1.4	1.8	4.1	3.4	3.5	3.5	3.6	1.3	3.6	3.5
	0.4	🟦	🟦	🟦	🟦	1.4	1.4	1.4	1.4	1.3	1.5	1.5
	0.45	1.0	1.0	1.3	1.6	🟦	0.8	0.8	0.8	1.3	0.8	0.8
	0.5	🟦	🟦	🟦	1.8	1.3	2.1	2.1	2.0	1.3	2.2	2.2
	0.55	🟦	🟦	🟦	1.7	1.5	1.4	1.4	1.3	1.3	1.4	1.4
	0.6	🟦	🟦	🟦	🟦	1.7	2.0	1.3	1.3	1.3	1.4	1.3
	0.65	🟦	🟦	🟦	2.4	1.8	2.2	1.8	🟦	🟦	🟦	🟦
	0.7	🟦	🟦	🟦	3.9	1.6	2.2	0.6	🟦	🟦	🟦	🟦
Column 6	0.2	2.8	2.8	4.8	17.0	3.2	🟧	🟦	🟦	🟦	🟦	🟦
	0.25	3.0	3.1	2.9	2.9	2.7	🟦	🟦	🟦	🟦	🟦	🟦
	0.3	2.3	2.2	2.2	3.3	2.8	🟦	🟦	🟦	🟦	🟦	🟦
	0.35	2.5	2.6	2.7	2.7	0.7	3.1	🟦	🟦	🟦	🟦	🟦
	0.4	3.2	3.2	3.2	3.1	3.5	🟦	🟦	🟦	🟦	🟦	🟦
	0.45	3.4	3.4	3.6	3.5	3.5	2.8	🟦	🟦	🟦	🟦	🟦
	0.5	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦

Legend:		
	🟩	both algorithms converge
	🟦	only the inside-out algorithm converges
	🟧	only the simultaneous algorithm converges

Figure 5.3: Ratio of simulation times for the inside-out/simultaneous algorithms for each test case with a soft purity specification.

		Soft product purity specified value λ_{spec}										
	z_1 value	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Column 2 (bubble-point feed)	0.35	0.568	0.568	0.568	0.568	0.568	0.568	0.6	0.7	0.8	0.9	0.908
	0.45	0.652	0.652	0.652	0.652	0.652	0.652	0.652	0.7	0.8	0.9	0.912
	0.55	0.717	0.717	0.717	0.717	0.717	0.717	0.717	0.717	0.8	0.9	0.917
	0.65	0.771	0.771	0.771	0.771	0.771	0.771	0.771	0.771	0.8	0.9	0.921
	0.75	0.817	0.817	0.817	0.817	0.817	0.817	0.817	0.817	0.817	0.9	0.925
	0.85	0.879	0.879	0.879	0.879	0.879	0.879	0.879	0.879	0.879	0.9	0.930
	0.9	0.932	0.932	0.932	0.932	0.932	0.932	0.932	0.932	0.932	0.932	0.932
Column 2 (dew-point feed)	0.35	0.696	0.696	0.696	0.696	0.696	0.696	0.696	0.7	0.8	0.9	0.906
	0.45	0.626	0.626	0.626	0.626	0.626	0.626	0.626	0.7	0.8	0.9	0.910
	0.55	0.563	0.563	0.563	0.563	0.563	0.563	0.6	0.7	0.8	0.9	0.914
	0.65	0.668	0.668	0.668	0.668	0.668	0.668	0.668	0.7	0.8	0.9	0.919
	0.75	0.774	0.774	0.774	0.774	0.774	0.774	0.774	0.774	0.8	0.9	0.924
	0.85	0.879	0.879	0.879	0.879	0.879	0.879	0.879	0.879	0.879	0.9	0.929
	0.9	0.932	0.932	0.932	0.932	0.932	0.932	0.932	0.932	0.932	0.932	0.932
Column 3	0.4	0.359	0.359	0.359	0.359	0.369	0.369	0.369	0.369	0.369	0.369	0.369
	0.45	0.359	0.359	0.359	0.359	0.4	0.412	0.412	0.412	0.412	0.412	0.412
	0.5	0.359	0.359	0.359	0.359	0.4	0.430	0.430	0.430	0.430	0.430	0.430
	0.55	0.359	0.359	0.359	0.359	0.4	0.452	0.452	0.452	0.452	0.452	0.452
	0.6	0.359	0.359	0.359	0.359	0.4	0.479	0.479	0.479	0.479	0.479	0.479
	0.65	0.359	0.359	0.359	0.359	0.4	0.5	0.509	0.509	0.509	0.509	0.509
	0.7	0.359	0.359	0.359	0.359	0.4	0.5	0.545	0.545	0.545	0.545	0.545
Column 4	0.2	0.395	0.395	0.395	0.395	0.4	0.434	0.434	0.434	0.434	0.434	0.434
	0.25	0.410	0.410	0.410	0.410	0.410	0.436	0.436	0.436	0.436	0.436	0.436
	0.3	0.421	0.421	0.421	0.421	0.421	0.438	0.438	0.438	0.438	0.438	0.438
	0.35	0.430	0.430	0.430	0.430	0.430	0.440	0.440	0.440	0.440	0.440	0.440
	0.4	0.437	0.437	0.437	0.437	0.437	0.442	0.442	0.442	0.442	0.442	0.442
	0.42	0.442	0.442	0.442	0.442	0.442	0.443	0.443	0.443	0.443	0.443	0.443
Column 5	0.32	0.283	0.283	0.283	0.284	0.284	0.284	0.284	0.284	0.284	0.284	0.284
	0.35	0.283	0.283	0.283	0.3	0.316	0.316	0.316	0.316	0.316	0.316	0.316
	0.4	0.284	0.284	0.284	0.3	0.369	0.369	0.369	0.369	0.369	0.369	0.369
	0.45	0.284	0.284	0.284	0.3	0.4	0.421	0.421	0.421	0.421	0.421	0.421
	0.5	0.284	0.284	0.284	0.3	0.4	0.474	0.474	0.474	0.474	0.474	0.474
	0.55	0.284	0.284	0.284	0.3	0.4	0.5	0.527	0.527	0.527	0.527	0.527
	0.6	0.284	0.284	0.284	0.3	0.4	0.5	0.579	0.579	0.579	0.579	0.579
	0.65	0.284	0.284	0.284	0.3	0.4	0.5	0.6	0.632	0.632	0.632	0.632
	0.7	0.284	0.284	0.284	0.3	0.4	0.5	0.6	0.684	0.684	0.684	0.684
Column 6	0.2	0.210	0.210	0.210	0.3	0.4	0.5	0.578	0.578	0.578	0.578	0.578
	0.25	0.263	0.263	0.263	0.3	0.4	0.5	0.578	0.578	0.578	0.578	0.578
	0.3	0.316	0.316	0.316	0.316	0.4	0.5	0.578	0.578	0.578	0.578	0.578
	0.35	0.368	0.368	0.368	0.368	0.4	0.5	0.577	0.577	0.577	0.577	0.577
	0.4	0.421	0.421	0.421	0.421	0.421	0.5	0.577	0.577	0.577	0.577	0.577
	0.45	0.473	0.473	0.473	0.473	0.473	0.5	0.577	0.577	0.577	0.577	0.577
	0.5	0.526	0.526	0.526	0.526	0.526	0.526	0.577	0.577	0.577	0.577	0.577

Legend:		
	🟥	$\lambda = \lambda_{r_{\min}}$
	🟦	$\lambda = \lambda_{r_{\max}}$

Figure 5.4: Value of the purity specification λ at the single-soft model solution for each test case with a soft λ specification.

Convergence reliability

First we analyze the convergence reliability of each method. The simultaneous and inside-out algorithms each fail to converge in 128/584 (22%) and 1/584 (0.17%) test cases, respectively. However, both methods converge for all test cases in Figure 5.1 using $\lambda = R$. This is compatible with our general expectations, since this type of column specification (R together with D) is the most numerically well-behaved one (see discussion in Section 4.5.3), even for highly non-ideal systems such as Column 7.

In Figure 5.3 we can distinguish two main patterns of convergence failure for the simultaneous algorithm. Firstly, these failures tend to concentrate at/around purity specification values λ_{spec} that lead the single-soft model to reset λ to the maximum flow rate solution $\lambda_{r_{\max}}$. For instance, we observe this in the test cases of Columns 5 and 6 with $\lambda_{\text{spec}} < \lambda_{r_{\max}} = 0.284$ and $\lambda_{\text{spec}} > \lambda_{r_{\max}} = 0.578$, respectively. This same type of convergence failure also happens for the benzene-toluene Column 2 ($\lambda_{r_{\max}} = 0.93$) with a bubble-point feed, $z_1 = 0.35$ or 0.45, and $\lambda_{\text{spec}} = 0.9$ or 1. Numerical difficulties near $\lambda_{r_{\max}}$ are to be expected given the infinite discontinuity in flow rates that happens at near-pure or near-azeotrope product purity specifications (see Sections 4.5.2 and 4.5.3). Though the single-soft adaptive model eliminates the actual discontinuity, the model equations can still become ill-conditioned around $\lambda_{r_{\max}}$ (see the steep slope in Figures 4.13 and 4.16). Nevertheless, with the inside-out algorithm we can reliably obtain solutions that are extremely close to an azeotrope pinch point. For example, for Column 3 ($\lambda = x_{1,1}$) we successfully converge to the $\lambda_{r_{\max}} = 0.359$ solution whereas the azeotropic composition of component 1 is 0.3585. Moreover, the inside-out algorithm can handle quite unreasonable user-specified values of λ_{spec} (even mole fractions < 0 or > 1, though not shown in Figures 5.2 and 5.4), and still return a “nearest best” MESH-feasible solution.

The second pattern in which the simultaneous algorithm may fail to converge is associated with the scenario of two specifications being simultaneously infeasible. The single-soft adaptive model is only capable of resetting a single specification if it happens to be infeasible due to flow rates going out of bounds. However, as illustrated in Sections 4.5.2 and 4.5.3, there are sets of column specifications in which two variables are infeasible simultaneously; in such cases, the single-soft model cannot reach an alternative, feasible solution. In particular, as exemplified in Figure 4.12, the feed composition must lie in between the hard purity specification and the range of possible values for the soft purity specification in order for the single-soft model to have a solution. The last row of test cases for Columns 2 (dew-point feed), 4 and 6 corresponds to a z_1 feed composition that is close to the feasibility limit imposed by the relevant azeotrope or near-pure component composition in each column. Though z_1 is not technically infeasible in these test cases, the proximity to a scenario with two infeasible specifications is enough to prevent the simultaneous (yet not the inside-out) algorithm from converging.

Interestingly, the simultaneous algorithm can sometimes fail to converge even for an extremely ideal mixture such as the benzene-toluene Column 2 due to either of the two mechanisms discussed above, while our inside-out algorithm remains reliable. The inside-out algorithm was originally developed by Boston and Sullivan Jr to handle the numerical difficulties associated with non-ideal mixture thermodynamics. The greater convergence reliability of our inside-out algorithm even for ideal systems indicates that this method can also be useful in overcoming other types of numerical difficulties, such as ill-conditioning issues near an infinite discontinuity.

The only test case in both Figures 5.1 and 5.3 for which the inside-out (but not the simultaneous) algorithm failed to converge corresponds to the diethylamine-methanol Column 6 with $z_1 = 0.2$ and $\lambda_{\text{spec}} = x_{N,1,\text{spec}} = 0.5$. However, we were able to converge this test case by using successive substitution to solve the outer loop instead of Anderson acceleration, and reducing the α^* value in Algorithm 2 from 0.3 to 0.2. This example illustrates the importance of step size control for the outer loop when converging particularly non-ideal systems. In such cases, acceleration techniques during the first outer loop iterations might preclude convergence of the inner loop when evaluating the outer loop function.

Convergence speed

Next, we compare algorithm speed. On average, for the test cases in Figures 5.1 and 5.3 in which both algorithms converge, the inside-out method is 2 times slower than the simultaneous method, with a standard deviation of approximately 1. The inside-out algorithm is slower for almost every test case of Columns 2 through 6, while it is faster in 90% of the test cases for Column 7. For systems without side products, the full set of model equations has size $2NN_c + 3N + 1$, while the inner loop of the inside-out method has size N. However, the latter must be repeatedly solved within the outer loop, for usually no more than about 8 outer loop iterations in our experience. Therefore, we can expect to have a break-even point between the running times of both algorithms as we increase the number of components N_c. When dealing with a binary mixture ($N_c = 2$), which is the case for Columns 2 through 6, the full set of equations is 7 times bigger than the inner loop. On the other hand, for Column 7 the former is already 13 times larger than the latter, which leads to the inside-out method algorithm being the fastest one.

Convergence under multiple steady states

In a second series of test cases we now revisit Example 5 from Section 4.5.5, which consists of Column 7 as specified in Table 4.2 using $D = 0.3F_s$ as the hard specification, $\lambda = x_{1,1}$ as the soft specification, and $r_{\max} = 10$. As seen in Figure 4.18, the MESH model exhibits multiple steady states for most of the narrow range $0.272 < x_{1,1} \leq 0.382$ of feasible $x_{1,1}$ values: two feasible states and a third, infeasible one with negative flow rates. As discussed in Section 4.5.5, the most adequate (though not obvious) value of α for Equation 4.6 in this example is equal to 1. Using said value, the single-soft adaptive model retains a very similar bifurcation curve to that of the MESH model except that the third state, if present, is feasible and corresponds to the minimum flow rate solution $\lambda = \lambda_{r_{\min}}$. Therefore, in this example we can compare not only the convergence reliability and speed of Methods 1 and 2 but also which of the multiple solutions each method converges to.

Figures 5.5 and 5.6 present the normalized flow rate V_8/F_s at the solutions that the simultaneous and inside-out algorithms converge to, respectively, when using $\alpha = -1$ for Column 7. Figures 5.7 and 5.8 present the same results using $\alpha = 1$ for each method. In each of the four figures, we vary $x_{1,1,\text{spec}}$ from 0.25 to 0.45 in 0.005 increments. Finally, Figure 5.9 presents the running time ratio of the inside-out/simultaneous algorithms at $x_{1,1,\text{spec}}$ values for which both methods converge, for $\alpha = 1$ and $\alpha = -1$.

In this example we could expect numerical issues not only due to the multiplicity of MESH solutions but also due to the narrow window between $\lambda_{r_{\min}} \approx 0.336$ and $\lambda_{r_{\max}} \approx 0.317$. For $\alpha = 1$, we see in Figures 5.5 and 5.6 that the inside-out algorithm converges for all test cases, while the simultaneous algorithm fails at 2/41 test cases. Moreover, the inside-out algorithm is able to reach an "intermediate" MESH solution when two of them exist, while the simultaneous algorithm is biased towards reaching the $\lambda_{r_{\min}}$ solution. The latter constitutes an artificially introduced third feasible steady state in this example.

In Figures 5.7 and 5.8 we can analyze how the convergence properties of the single-soft model change by choosing a less adequate value of α. As discussed in Section 4.5.5, we would choose $\alpha = -1$ for Column 7 based on the general strategy of Table 4.1. However, this yields a set of solutions that is counter-intuitive regarding the direction in which the method resets infeasible $\lambda = x_{1,1}$ values. This alternative α value compromises the convergence reliability of both the simultaneous and inside-out methods, although the latter still outperforms the former both in the fraction of successfully converged test points and in being able to reach intermediate feasible solutions. In Figure 5.9 we see that the inside-out method still remains faster than the simultaneous one for most test cases in this multicomponent ($N_c = 5$) column with either $\alpha = 1$ and $\alpha = -1$.

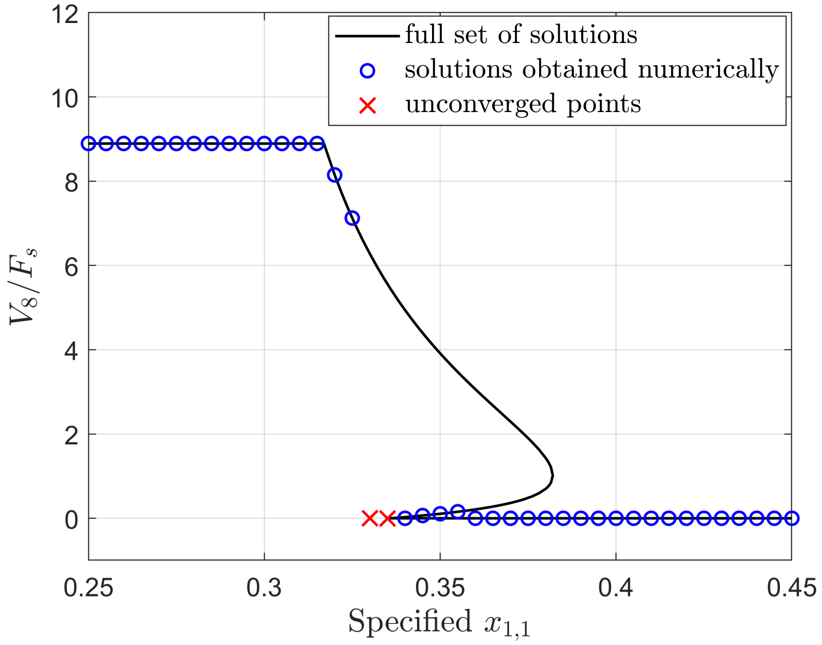


Figure 5.5: Converged solutions in terms of V_8/F_s versus $x_{1,1,\text{spec}}$ for Column 7 using the simultaneous algorithm and $\alpha = 1$ in Equation 4.6.

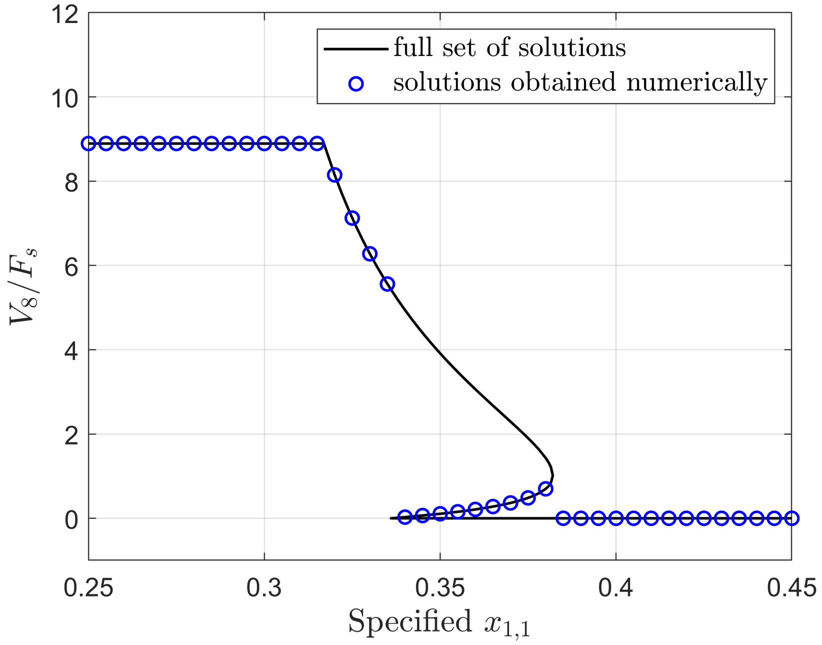


Figure 5.6: Converged solutions in terms of V_8/F_s versus $x_{1,1,\text{spec}}$ for Column 7 using the inside-out algorithm and $\alpha = 1$ in Equation 4.6.

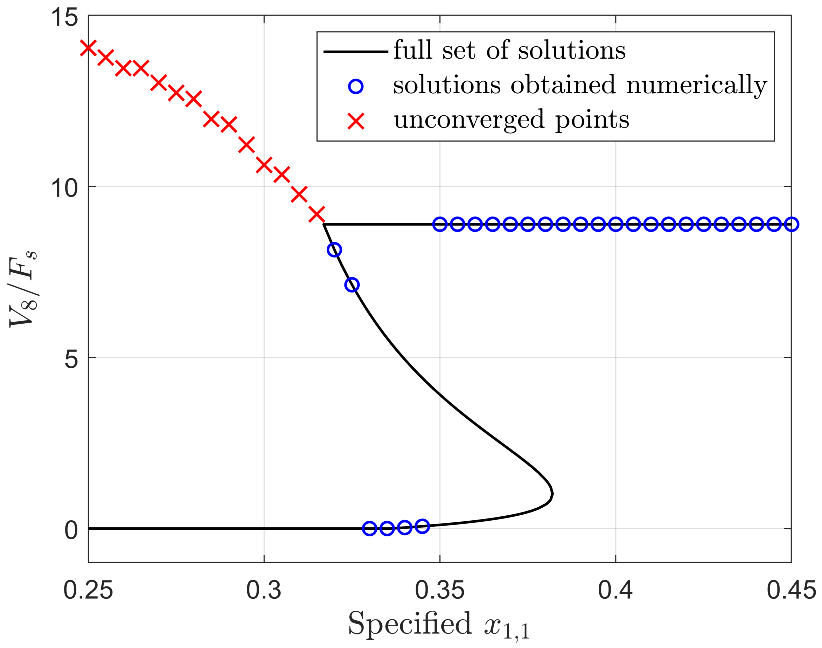


Figure 5.7: Converged solutions in terms of V_8/F_s versus $x_{1,1,\text{spec}}$ for Column 7 using the simultaneous algorithm and $\alpha = -1$ in Equation 4.6.

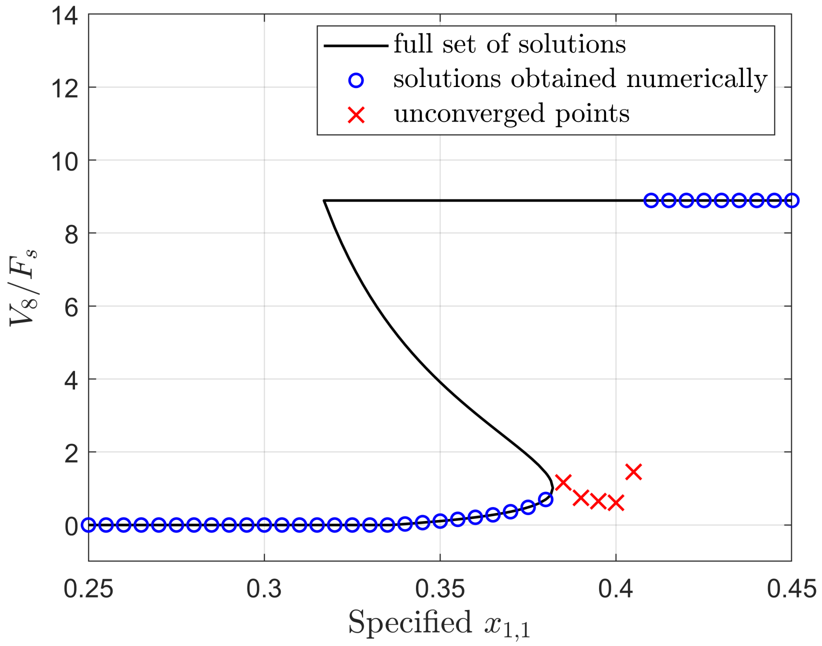


Figure 5.8: Converged solutions in terms of V_8/F_s versus $x_{1,1,\text{spec}}$ for Column 7 using the inside-out algorithm and $\alpha = -1$ in Equation 4.6.

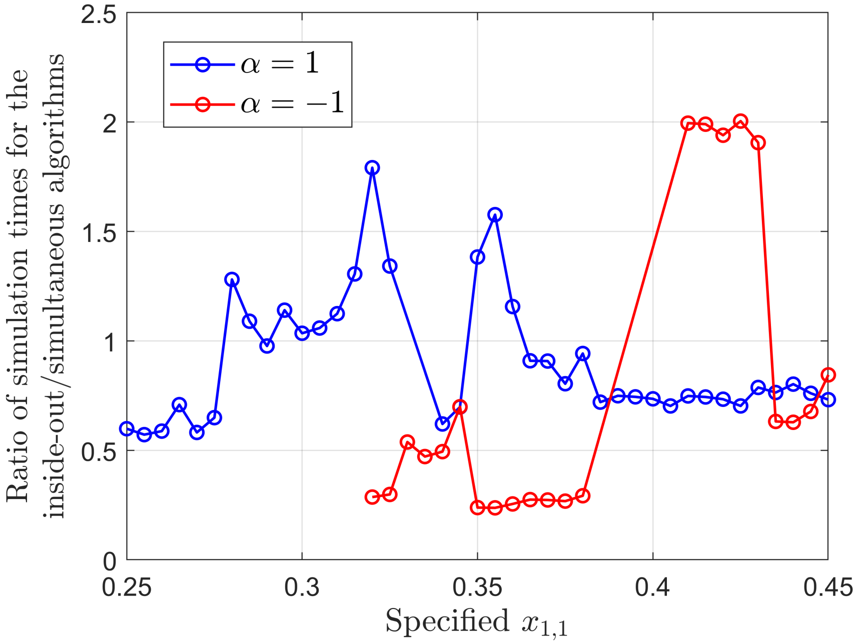


Figure 5.9: Ratio of simulation times for the inside-out/simultaneous algorithms versus specified $x_{1,1}$ using $\alpha = \pm 1$ in Equation 4.6.

5.4.2. RadFrac versus the single-soft adaptive model

In this section we compare the performance of the single-soft inside-out algorithm (Method 2) to that of Aspen Plus' RadFrac. In all test cases we keep all convergence options and algorithms in RadFrac at their default values, and always purge any previous results before simulating each test case so as to not override the internal initialization procedure of RadFrac. We refer the reader to Section 4.1.1 of Chapter 4 for a description of RadFrac's standard algorithm and commonly encountered convergence error messages.

As an interpreted language, MATLAB is considerably slower than Fortran, which is the compiled language used by Aspen Plus. Moreover, as previously stated, our MATLAB implementation was not optimized for speed. Therefore, we refrain from comparing the absolute running times of our algorithms with Aspen Plus' RadFrac and instead compare only convergence reliability. Another limitation in comparing model performance is that, unlike our single-soft adaptive model, RadFrac cannot possibly return a feasible solution when any of the column specifications are infeasible. That fact already stands in favor of the single-soft model. Therefore, in this section we focus on investigating test cases with feasible column specifications for which RadFrac nonetheless fails to converge.

We now describe the test cases performed for Columns 3 and 7 from Table 4.2 and report the results obtained, which we will discuss afterwards.

Column 3

For this column, the range of MESH feasible $\lambda = x_{1,1}$ values is $0.3585 < x_{1,1} \leq 0.530 = \lambda_{r_{\min}}$, where the lower bound corresponds to a minimum-boiling azeotrope. When using $r_{\max} = 5$ we have $\lambda_{r_{\max}} = 0.359$, therefore in this case the achievable range of values with the single-soft model is $0.359 \leq x_{1,1} \leq 0.530$. We simulated both RadFrac and the single-soft inside-out algorithm with $\lambda = x_{1,1}$ by varying $x_{1,1,\text{spec}}$ from 0.35 to 0.55 in 0.005 increments. In RadFrac, we specify $D = 0.3F_s$ in the standard Setup tab and create a design specification to enforce $x_{1,1,\text{spec}}$ values by varying R. RadFrac converges only for $0.37 \leq x_{1,1,\text{spec}} \leq 0.52$, while our Method 2 converges successfully for all test cases.

Column 7

First, we simulate Column 7 in RadFrac by specifying $D = 0.3F_s$ and a range of R values greater than zero, corresponding to the same type of test cases performed for our adaptive model in the last column of Figure 5.1. RadFrac converges successfully for all test cases with $R \geq 2.29$, even as far as $R = 100,000$; $R \leq 2.285$ values lead to a dry column error message. We can approach the minimum allowed value $R \approx 2.287$ quite closely without needing to supply a specific initial guess.

In a second series of tests, we kept $D = 0.3F_s$ and specified $x_{1,1}$ values ranging from 0.3 to 0.38 in 0.01 increments, using R as the manipulated variable. RadFrac only converged to a solution in 1/21 test cases, for $x_{1,1,\text{spec}} = 0.34$. We obtained a dry column error for $x_{1,1,\text{spec}} < 0.31$ and the "RadFrac not converged in 25 outside loop iterations" error message for all other specifications. This stands in contrast to the excellent performance of our Method 2 in Figure 5.6 and even to its sub-optimal outcomes in Figure 5.8.

The RadFrac model can enforce R, D specifications very rapidly and effectively, regardless of mixture non-ideality. This is understandable given that RadFrac's inside-out algorithm structure is optimized for R, D specifications, and that flow rate values do not exhibit infinite discontinuities with respect to R. The only type of issue we may encounter in general with R, D specifications in RadFrac consists of dry column errors. However, the latter do not seem to stem from failure in converging the model equations, but simply from the built-in internal safeguard in RadFrac that shuts down intermediate calculations when any $L_j, V_j < 10^{-5} F_s$. On the other hand, in our algorithms we allow for flow rates to assume any values during iterations, including negative ones. Further, our adaptive models always enforce $L_j, V_j \geq 0$ for all internal flow rates at the solution, and are able to converge to minimal flow rate solutions in which some $L_j, V_j = 0$.

Aside from dry column errors, the other main type of failure we can encounter during single-column simulation in Aspen Plus is related to product purity specifications near an azeotrope pinch point. In the test cases of Column 3, RadFrac fails for $x_{1,1,\text{spec}} < 0.37$ despite the problem remaining feasible as we approach the azeotropic composition $x_{\text{azeo},1} \approx 0.3585$ from above. In this scenario we usually encounter the generic "RadFrac not converged in 25 outside loop iterations" error message, which does not provide any insight to the user about the nature of the issue. In contrast, with our Method 2 we can converge to maximum flow rate solutions for which $\lambda_{r_{\max}}$ is much closer to the azeotropic composition. Interestingly, RadFrac also performs poorly in the Column 7 test cases with an $x_{1,1}$ specification even when we are not close to an azeotrope.

We could speculate that RadFrac's sometimes poor performance with purity specifications might be related to its inside-out algorithm structure. In the latter, this type of specification must be numerically enforced indirectly through a third, middle loop, given that the inner loop is designed to accept R, D specifications. This indirectness might jeopardize RadFrac's ability to converge near an azeotrope, or when multiple steady states in terms of L_j, V_j values exist with respect to a product purity but not to an R specification.

The latter holds for our Column 7 example, whose bifurcation diagram with respect to R is analogous to Figure 4.5.

5.5. Robust distillation simulation for flowsheet calculations

When converging a flowsheet with recycle streams, the standard procedure is to use the solution from the previous flowsheet pass as the initial guess for each equipment in the current flowsheet iteration. Therefore, we can expect to have a reasonably good initial guess in most flowsheet passes, except the very first one, to simulate any distillation columns that might be present. The single-column simulation test cases from Section 5.4.1 demonstrate that the inside-out algorithm is considerably more reliable to converge the single-soft adaptive model than the simultaneous algorithm, which is mostly useful in the absence of a good initial guess. However, the former method is slower than the latter for systems with few components. Therefore, it would make sense, in terms of computational efficiency, to resort to the inside-out algorithm only in case of failure of the simultaneous algorithm, which is more likely to occur during the first flowsheet pass. Another point to consider is that, as the compositions of the column feed streams are varied during intermediate flowsheet iterations, we may encounter conditions in which both column specifications are infeasible. In this case, to obtain an alternative MESH-feasible solution we must resort to the double-soft adaptive model (see Sections 4.5.2 and 4.5.6). The latter can also be converged with either the simultaneous or the inside-out algorithms.

In this section we propose a four-tier distillation modeling strategy for converging flowsheets with recycle streams. To simulate each distillation column we make use of up to four different simulation methods in the follow hierarchical order:

1. The single-soft adaptive model converged with the simultaneous algorithm;

2. The single-soft adaptive model converged with the inside-out algorithm;
3. The double-soft adaptive model converged with the simultaneous algorithm;
4. The double-soft adaptive model converged with the inside-out algorithm.

That is, we move on to the next simulation method if the current one fails to converge. We will refer to Methods 2 through 4 as advanced simulation methods, given that they are only required when the single-soft simultaneous algorithm fails.

5.6. Flowsheet simulation test cases

We now test the four-tier distillation modeling strategy from Section 5.5 to simulate two flowsheets for performing pressure-swing distillation: one for ethanol-benzene (Flowsheet 1) and the second one for diethylamine-methanol purification (Flowsheet 2). In this type of process, two distillation columns operating with distinct pressures are used to circumvent an azeotrope and obtain two high-purity products from a binary mixture. One of the outputs of the second column is recycled as a feed stream to the first column, hence we must converge the flowsheet by iteratively adjusting the variables of the recycle tear stream.

We use the same tolerance values and equation solving methods from Section 5.4.1 for the simultaneous and inside-out algorithms, whether they are being applied to the single-soft or double-soft models. In the first flowsheet pass we use the initial guess $\mathbf{X}^0$ obtained according to Section 4.3, while subsequent passes use the converged solution of the preceding flowsheet iteration. The termination criteria for each of the four simulation methods influences the computational efficiency of the four-tier strategy. We terminate both the simultaneous algorithm and the inner loop after a maximum of 45 iterations, and the outer loop after a maximum of 25 iterations. We also terminate each equation-solving task early if the norm residual grows beyond a certain threshold, if oscillations or stagnation are detected, or if the iterate or function residual becomes undefined or cannot be evaluated. As another strategy to improve computational efficiency, we start from the double-soft model with the simultaneous algorithm for a given column if a double-soft model had to be used for 3 or more previous flowsheet passes in that respective column.

To simulate both flowsheet examples, we used a tolerance value of $\epsilon = 10^{-3}$ for the infinity norm of the tear (i.e., recycle) stream residual, which includes discrepancies in flow rate and mole fraction values. Since the pressure and vapor fraction of the tear stream are known constants in both flowsheets, we did not include the tear stream temperature as a convergence variable. For all test cases in each flowsheet, we initialized the tear stream with the fresh feed composition ($\mathbf{z} = (2/3, 1/3)$ for ethanol-benzene and $\mathbf{z} = (0.5, 0.5)$ for diethylamine-methanol) and a flow rate of zero.

We converge the flowsheet tear stream using the Anderson acceleration algorithm of Zhang et al. [91]. Our four-tier distillation model is not only nonsmooth in general, but also potentially discontinuous since we might switch from the single-soft to the double-soft model during an intermediate flowsheet iteration. In turn, that makes the flowsheet residual function also potentially discontinuous. However, the convergence theorem for the Zhang et al. algorithm (Theorem 4.1 in [91]) requires the function to be non-expansive, which implies continuity. Regardless of the applicability of said convergence theorem to our flowsheet residual function, we have nonetheless been able to converge our test cases successfully with this algorithm.

We also simulate test cases for both flowsheets in Aspen Plus V10 using fully feasible specifications for each column. As in Section 5.4.2, we keep the default convergence options and algorithms in RadFrac, and always purge any previous results before simulating each flowsheet test case.

5.6.1. Flowsheet 1: ethanol-benzene pressure-swing distillation

We consider the ethanol-benzene pressure-swing distillation flowsheet from Example 11.5 of the textbook [79], which is schematized in Figure 5.10 and henceforth referred to as

Flowsheet 1. The first and second columns in the flowsheet, which we refer to as Columns A and B, correspond to Columns 3 (without the recycle stream) and 4 in Table 4.2, respectively. The bifurcation diagram of Column A with respect to $x_{1,1,\mathrm{spec}}$ was analyzed in Section 4.5.3. We describe the liquid phase with the NRTL model and the vapor phase is treated as ideal. The thermodynamic models used by the authors in [79] were not disclosed. We take ethanol to be component $i = 1$. The mixture tends to form a minimum-boiling azeotrope at the top of both columns, with composition $\mathbf{z}_{\mathrm{azeo}} = (0.3585, 0.6415)$ at Column A's top tray pressure of 0.3 bar, and composition $\mathbf{z}_{\mathrm{azeo}} = (0.449, 0.551)$ at Column B's top tray pressure of 1.06 bar.

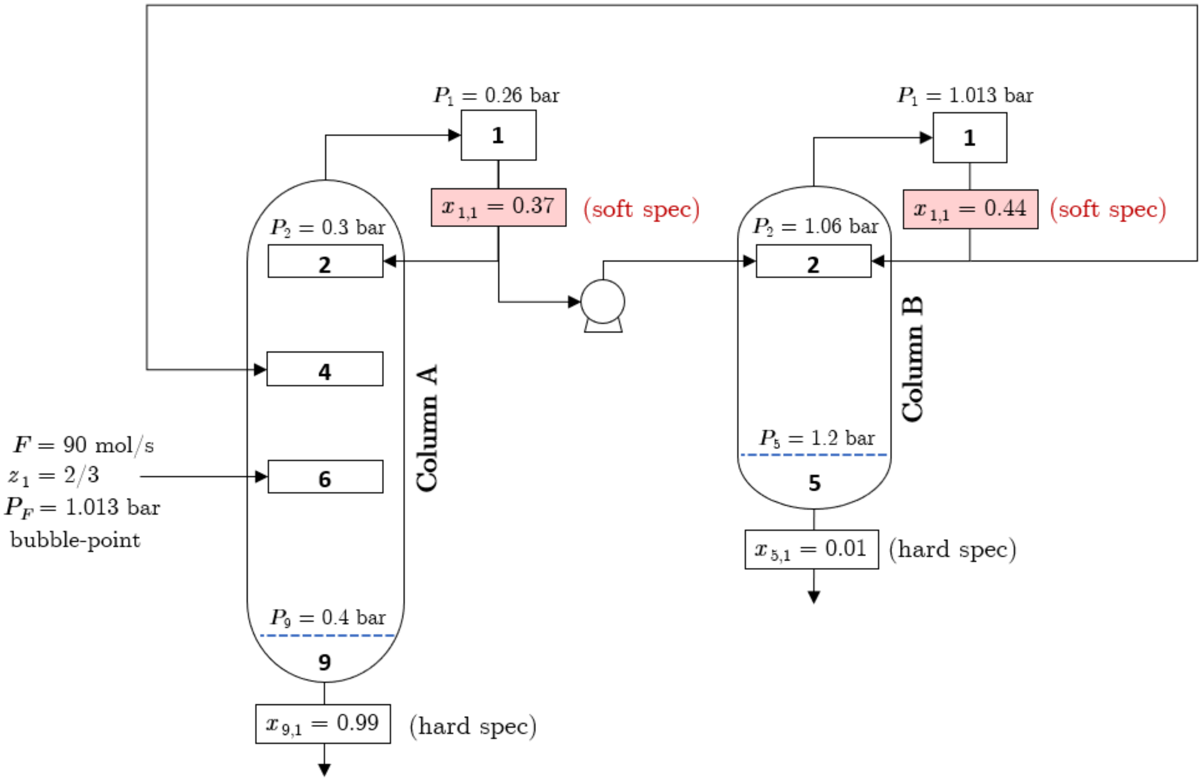


Figure 5.10: The ethanol-benzene pressure-swing distillation flowsheet from Example 11.5 of [79] (Flowsheet 1).

In [79], the two main specifications are $x_{1,1} = 0.37$, $x_{N,1} = 0.99$ for Column A, and $x_{1,1} = 0.44$, $x_{N,1} = 0.01$ for Column B. Using $x_{1,1}$ soft specifications, our flowsheet simulation method converges to a fully feasible solution that agrees with that of [79] (e.g., we obtain a recycle stream flow rate value of 152.7 mol/s compared to 152.9 mol/s in [79]). The only advanced distillation simulation method required is the single-soft inside-out method, which is used in the first flowsheet pass for Column A.

To explore a wider parameter space range and analyze the robustness of our simulation method to infeasible specifications, we set $\lambda = x_{1,1}$ as a soft specification and vary $x_{1,1,\text{spec}}$ from 0 to 1 in 0.1 increments for each column, making up a total of 121 test cases. Figures 5.11 and 5.12 present the $x_{1,1}$ values at the obtained flowsheet solutions for Columns A and B, respectively, for each combination of $x_{1,1,\text{spec}}$ values. Figure 5.13 presents the total number of flowsheet iterations needed to converge each test case, and Figure 5.14 presents the number of flowsheet passes in which an advanced distillation method was needed to converge Column B. No advanced method was required for Column A. For all test cases, the hard specifications $x_{N,1} = 0.99$ for Column A and $x_{N,1} = 0.01$ for Column B were satisfied at the final flowsheet solution.

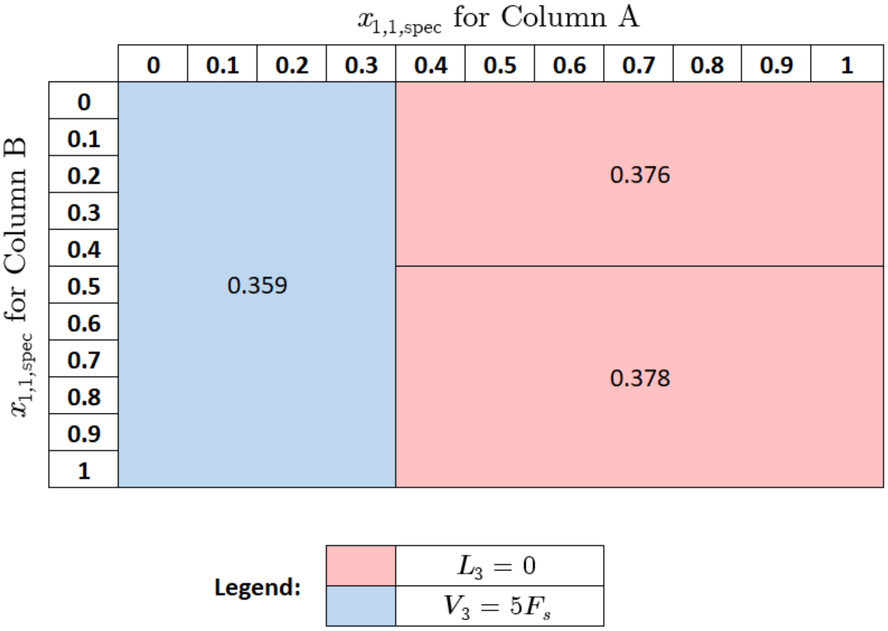
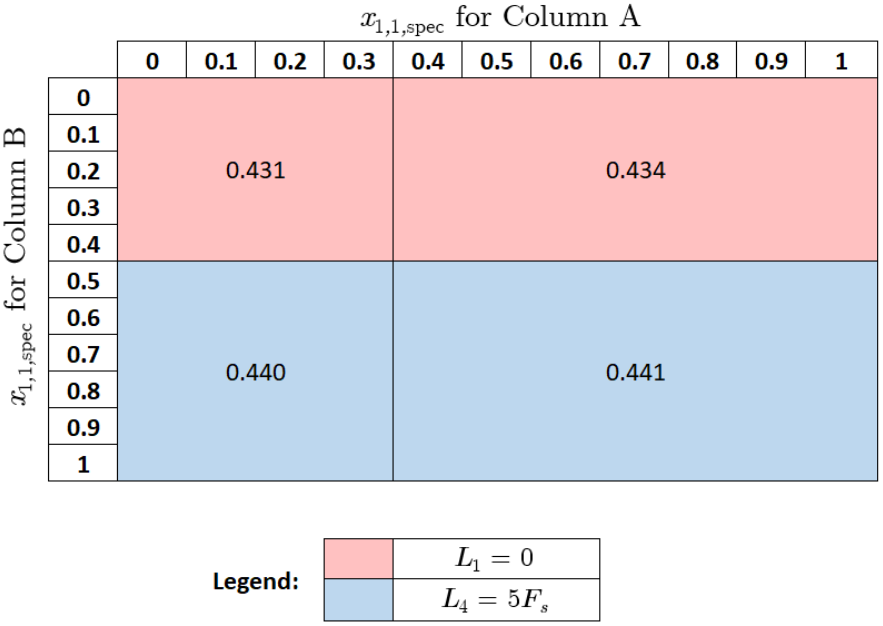


Figure 5.11: $x_{1,1}$ value at the solution for Column A in Flowsheet 1.

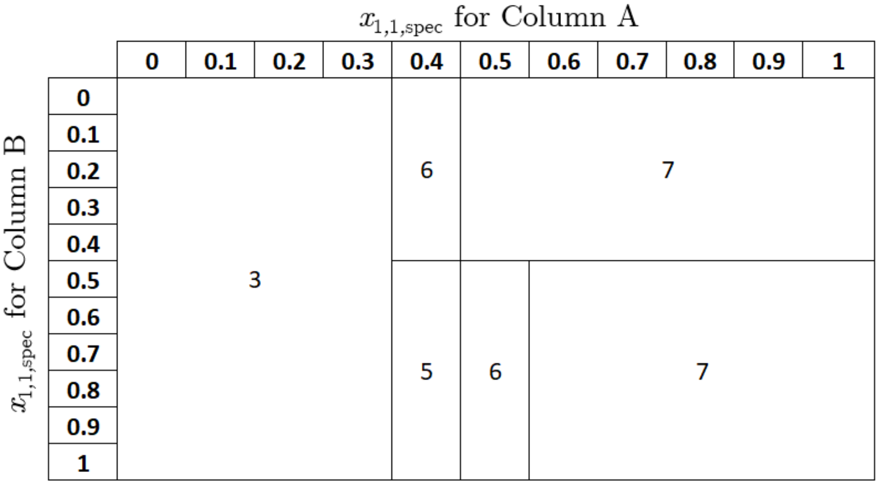


		$x_{1,1,\text{spec}}$ for Column A										
		0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
$x_{1,1,\text{spec}}$ for Column B	0	0.431				0.434						
	0.1											
	0.2											
	0.3											
	0.4											
	0.5	0.440				0.441						
	0.6											
	0.7											
	0.8											
	0.9											
	1											

Legend:

🟥	$L_1 = 0$
🟦	$L_4 = 5F_s$

Figure 5.12: $x_{1,1}$ value at the solution for Column B in Flowsheet 1.



		$x_{1,1,\text{spec}}$ for Column A										
		0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
$x_{1,1,\text{spec}}$ for Column B	0	3				6	7					
	0.1											
	0.2											
	0.3											
	0.4											
	0.5					5	6	7				
	0.6											
	0.7											
	0.8											
	0.9											
	1											

Figure 5.13: Total number of iterations to converge Flowsheet 1.

In Figures 5.11 and 5.12 we only see 4 mainly distinct flowsheet solutions, in which each column operates at either its maximum or minimum flow rate solution. In both columns the latter corresponds to an internal flow rate being equal to zero, and therefore the positive $r_{\min} = 0.05$ value used does not impact $\lambda_{r_{\min}}$. We can conclude that every tested value of $x_{1,1,\text{spec}}$ is infeasible for both columns, though we are able to specify the hard

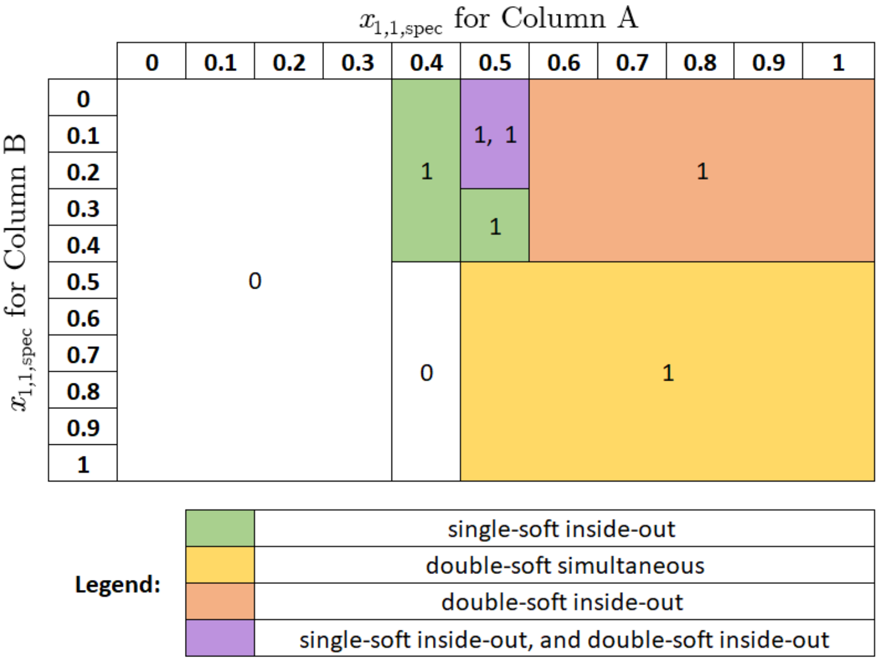


Figure 5.14: Number of flowsheet iterations in which each advanced distillation model was needed for Column B in Flowsheet 1.

$x_{N,1,\text{spec}}$ values at every solution. Despite that, with the single-soft model we are able to obtain a MESH-feasible flowsheet solution in all test cases. Further, we observe extremely narrow ranges of feasible $x_{1,1}$ values for both columns: $0.3585 < x_{1,1} \leq 0.376$ and $0.431 \leq x_{1,1} < 0.449$ using the azeotropic compositions for Columns A and B, respectively. With our single-soft adaptive model using $r_{\max} = 5$, the reachable values are $0.359 \leq x_{1,1} \leq 0.376$ for Column A and $0.431 \leq x_{1,1} \leq 0.441$ for Column B.

Though the purity limits corresponding to the azeotropic compositions are fixed for a given pressure, the $\lambda_{r_{\min}}$ value is a function of other column specifications (e.g., see Figure 4.15). At first glance we could assume that the feasible ranges of $x_{1,1}$ values are narrow due to the small number of stages in each column ($N_A = 9$ and $N_B = 5$); however, we observe the opposite trend. For example, if we set $N_A = 15$, $N_B = 12$, introduce the fresh and recycle streams to Column A at Stages 10 and 6, and keep the feed stream to Column B at Stage 2, we obtain slightly narrower feasible ranges of $0.3585 \leq x_{1,1} \leq 0.374$ for Column A and $0.438 \leq x_{1,1} \leq 0.449$ for Column B. In general, the $\lambda_{r_{\min}}$ and $\lambda_{r_{\max}}$ values grow closer as the feed composition approaches a near-pure component or azeotropic composition until they eventually coincide (e.g., see Figure 4.11 for the ideal benzene-toluene Column 2). We observe this behavior in Figures 5.15 and 5.16, which present the type of MESH solution for Columns A (without the recycle stream) and B for each pair of $x_{1,1}$, z_1 values, keeping the hard specification $x_{N,1,\text{spec}}$ constant for each column.

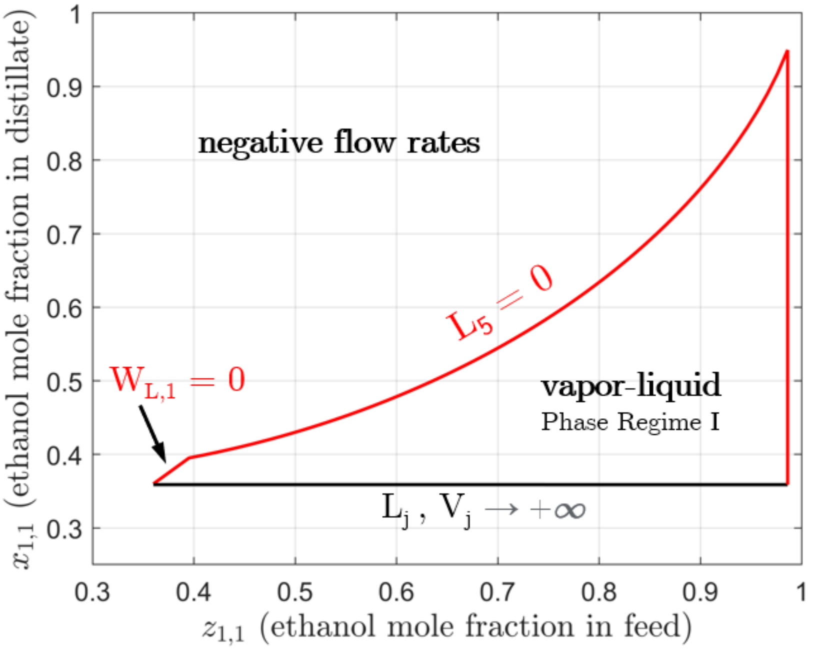


Figure 5.15: Type of MESH model solution for each pair of $x_{1,1}$ and z_1 values for Column A without the recycle stream (i.e., Column 3 in Table 4.2).

As discussed in Section 4.5.2, the composite feed composition must lie between the top and bottom product purities in order to be feasible. In Figures 5.15 and 5.16, the two feasibility limits for the composite z_1 correspond to the azeotropic composition $x_{\text{azeo},1}$ and the hard specification $x_{N,1,\text{spec}}$. In Figure 5.15 we see that the fresh feed composition $z_{\text{fresh},1} = 2/3$ already determines a somewhat narrow range for $x_{1,1}$ in Column A. The composition $x_{1,1}$ of Column B's distillate, which is the second feed to Column A, is necessarily < 0.45. The composite feed of Column A in the converged flowsheet tends to be much closer to the latter and thus determines an even narrower feasible range of purities for its distillate. Any achievable $x_{1,1}$ value for Column A, which corresponds to z_1

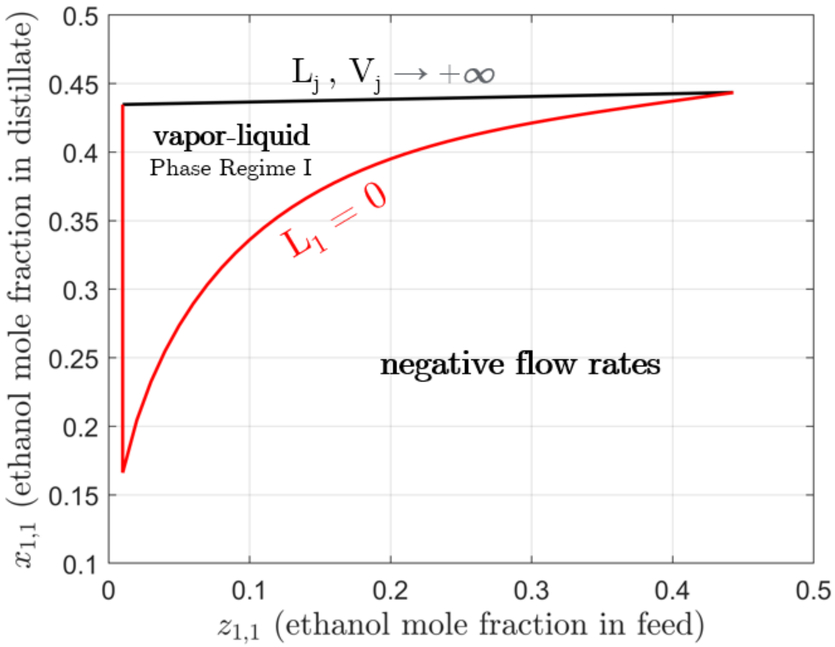


Figure 5.16: Type of MESH model solution for each pair of $x_{1,1}$ and z_1 values for Column B.

for Column B, also determines an extremely narrow feasible range for $x_{1,1}$ in Column B as seen in Figure 5.16. Therefore, in order to obtain wider feasible $x_{1,1}$ ranges in Flowsheet 1 we could increase the fresh feed composition $z_{\text{fresh},1}$, and/or increase the pressure difference between the two columns so as to introduce a wider separation between the azeotropes.

In general we may need to use one or more of the advanced distillation simulation methods to converge more challenging sets of flowsheet specifications, especially during the first flowsheet iteration(s) and in the absence of a good initial guess for the tear stream. In Figure 5.14 we see that we have to utilize a double-soft model once for Column B, in this case during the first flowsheet pass, whenever we specify $x_{1,1,\text{spec}} \geq 0.5$ in Column A. Given our initial guess of zero for the tear stream flow rate, the feasibility range for Column A in the first flowsheet pass corresponds to Figure 5.15 with $z_1 = 2/3$. In this figure we see that $x_{1,1,\text{spec}} \geq 0.5$ in Column A leads to a composition value of $0.5 \leq x_{1,1} \leq 0.52$ for the Column B feed, which is infeasible as seen in Figure 5.16. Therefore, to solve Column B in the first flowsheet pass we must resort to the double-soft model. In all subsequent iterations the tear stream has a non-zero flow rate, which in this example is enough to make the feed to Column B feasible. Therefore, we can conclude that the double-soft model can act as a buffer against poor initial guesses for the tear stream, while the inside-out algorithm provides robustness against poor initial guesses for the state of each column in the first flowsheet iteration.

Test cases in Aspen Plus

First, we use Aspen Plus to converge Flowsheet 1 with the fully-feasible specification values from [79], i.e., $x_{1,1} = 0.37$, $x_{N,1} = 0.99$ for Column A and $x_{1,1} = 0.44$, $x_{N,1} = 0.01$ for Column B. If we do not provide any initialization for the tear stream variables, which corresponds to guessing a tear stream flow rate $F_{\text{tear}} = 0$, we obtain a "Column not in mass balance" convergence error message for both Columns A and B. As previously stated, our simulation method converges this set of specifications successfully even when guessing $F_{\text{tear}} = 0$.

In our Aspen Plus test cases we have observed that F_{tear} is the pivotal variable when providing an initial guess for the tear stream. This way, to determine how good of an initial guess Aspen Plus requires to converge this and other test cases in this chapter, we initialize the tear stream with the correct final values for all its variables (composition, temperature, pressure) except F_{tear}, whose value we increase starting from zero. With this strategy, we are only able to converge to the final solution $F_{\text{tear}} = 152.7$ mol/s if we provide an initial guess with $F_{\text{tear}} \geq 167$ mol/s. Interestingly, initializing the flowsheet with the correct final state of the tear stream does not allow Aspen Plus to converge.

If we change the top product specification of Column A to $x_{1,1} = 0.36$, Aspen Plus is unable to converge the flowsheet regardless of the initial guess provided for the tear stream (including its correct state). Further, the simulation also fails if we first initialize the whole flowsheet with the converged values corresponding to $x_{1,1} = 0.37$. In contrast, our simulation method is able to converge to solutions in which the top product of Column A is even closer to the azeotrope, e.g., to the minimum flow rate solution with $x_{1,1} = 0.359$, starting from $F_{\text{tear}} = 0$.

5.6.2. Flowsheet 2: diethylamine-methanol pressure-swing distillation

In this section we study the pressure-swing distillation flowsheet from Figure 3a of [92] (Flowsheet 2) for the binary system diethylamine-methanol, which forms a pressure-sensitive maximum-boiling azeotrope. The flowsheet configuration and specified parameters are presented in Figure 5.17. The first and second columns (Columns A and B) correspond to Columns 5 and 6 in Table 4.2, respectively. We take diethylamine as component $i = 1$. Section 4.5.4 presented the bifurcation diagram of Column A with respect to $x_{N,1,\text{spec}}$. As in [92], the UNIQUAC activity model is used to describe the liquid phase, while the vapor phase is treated as ideal. We impose a total pressure drop of 0.3 atm for each of the columns as suggested by Iqbal et al. in [40] for the same flowsheet, since the pressure drop was not reported by Zhang et al. in [92]. The mixture tends to form a maximum-boiling azeotrope at the bottom of each column, with composition $\mathbf{z}_{\text{azeo}} = (0.285, 0.715)$ at Column A's reboiler pressure of 1.1 atm, and composition $\mathbf{z}_{\text{azeo}} = (0.579, 0.421)$ at Column B's reboiler pressure of 10.3 atm.

In [92] the authors report only the final flowsheet solution without disclosing which two main specifications were set for each column. Since we are interested in converging product purity specifications, we use their solution values of $x_{1,1} = 0.996$, $x_{N,1} = 0.3$ for Column A and $x_{1,1} = 0.004$, $x_{N,1} = 0.54$ for Column B as our specifications. With the latter our method converges in 2 flowsheet iterations, without requiring advanced distillation simulation methods, to a solution for which $R = 3.903$ and $R = 1.275$ for Columns A and B, and $F_{\text{tear}} = 61.7$ kmol/h. The corresponding solution values reported in [92] are $R = 3.608$, $R = 1.316$, and $F_{\text{tear}} = 65.07$ kmol/h.

Next, we set $\lambda = x_{N,1}$ as a soft specification in both columns and simulate 121 test cases with our modeling strategy by varying $x_{N,1,\text{spec}}$ from 0 to 1 in 0.1 increments for each column. Figures 5.18 and 5.19 present the $x_{N,1}$ values at the obtained flowsheet solutions

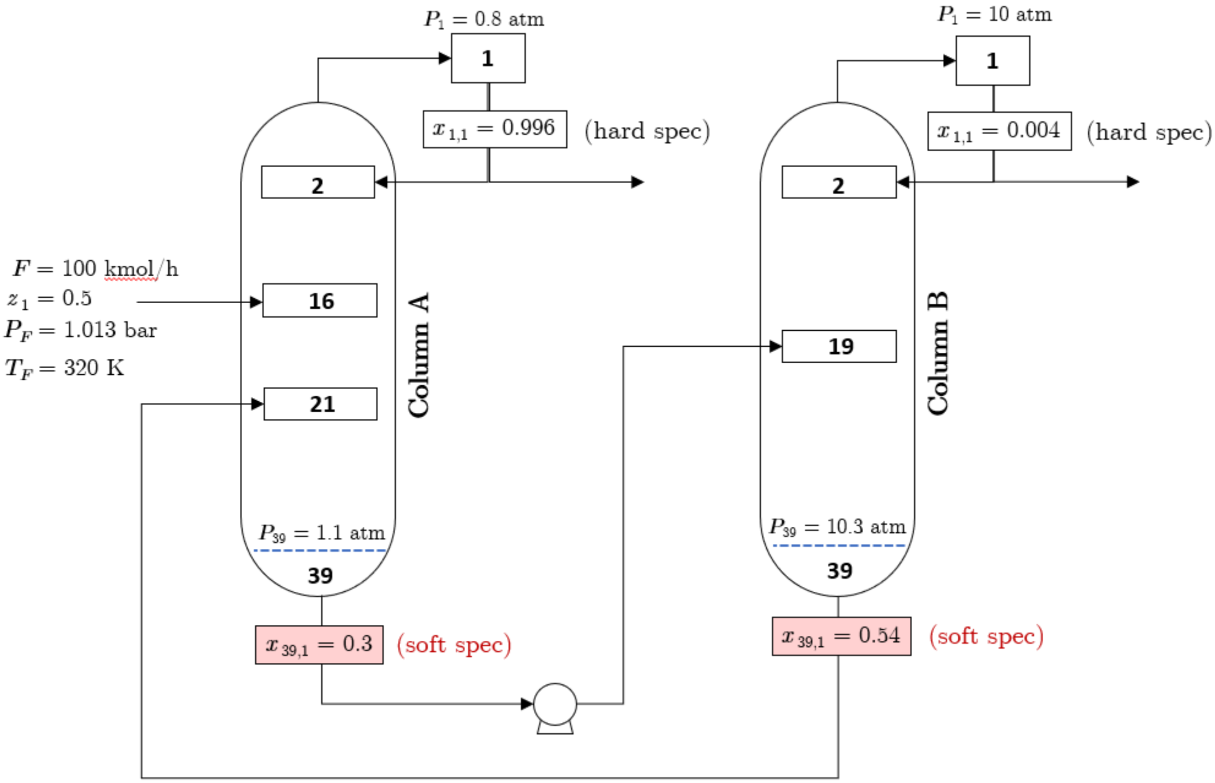
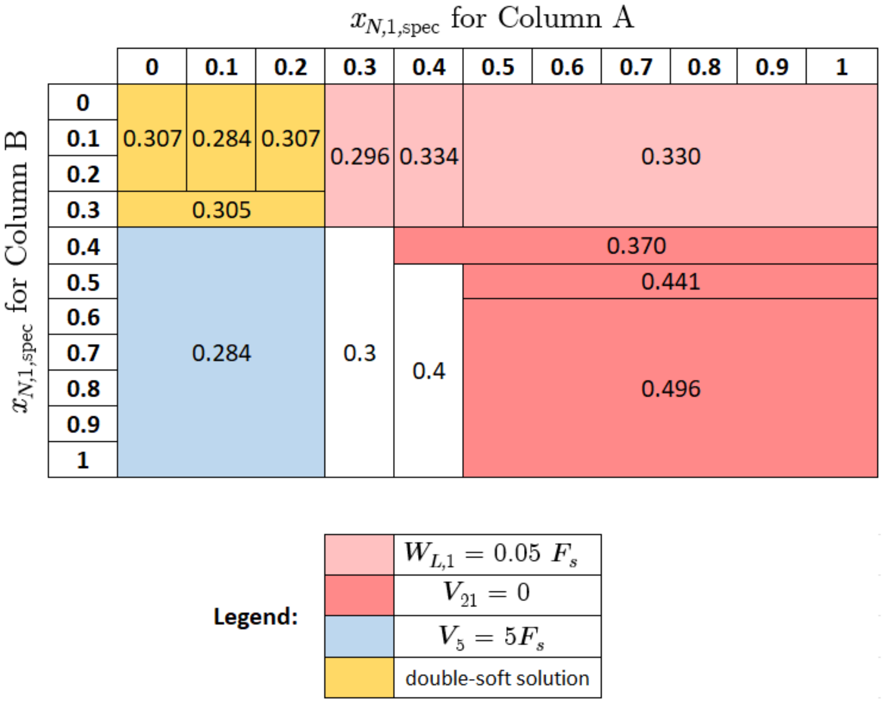


Figure 5.17: The diethylamine-methanol pressure-swing distillation flowsheet from Figure 3a of [92] (Flowsheet 2).

for Columns A and B, respectively. While the hard specification $x_{1,1} = 0.004$ for Column B was satisfied at all test cases, Figure 5.20 shows the final $x_{1,1}$ values for Column A, which were not equal to 0.996 for 12/121 of the test cases. Figure 5.21 shows the total number of flowsheet iterations needed to converge each test case. Finally, Figures 5.22 and 5.23 present the number of times an advanced distillation method was required for Columns A and B, respectively.

In Figures 5.18 and 5.19 we see a wider range of feasible $x_{N,1}$ values compared to Figures 5.11 and 5.12 for Flowsheet 1. Though each column exhibits a constant $\lambda_{r_{\max}}$ value, we observe distinct $\lambda_{r_{\min}}$ values depending on the test case. As previously discussed, $\lambda_{r_{\min}}$ can be a strong function of the feed composition, thus the state of Column A at the flowsheet solution influences the $\lambda_{r_{\min}}$ value for Column B and vice versa. Though the combined region of feasible $x_{N,1}$ values for both columns is not exactly rectangular,



$x_{N,1,\text{spec}}$ for Column A

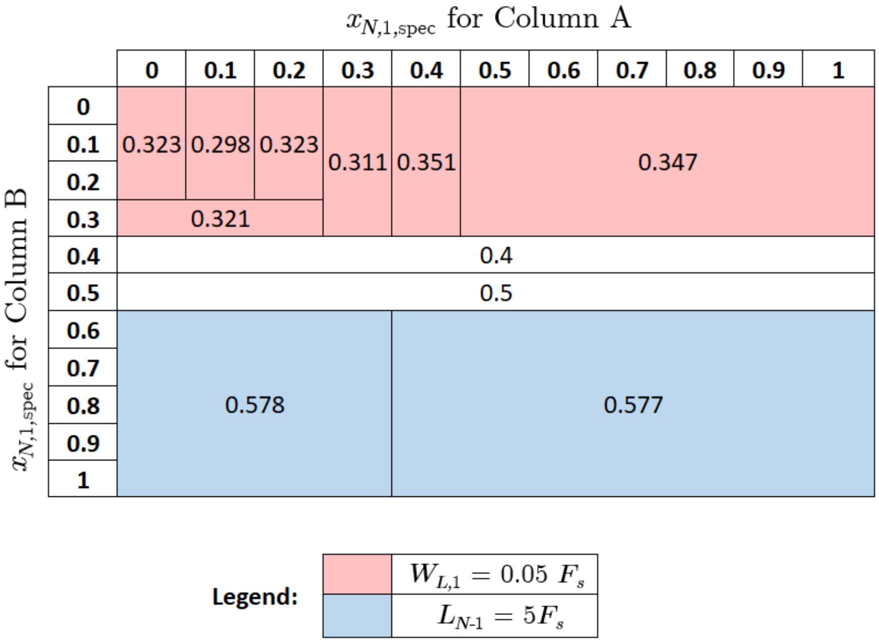
$x_{N,1,\text{spec}}$ for Column B

<table>
<tr><th></th><th>0</th><th>0.1</th><th>0.2</th><th>0.3</th><th>0.4</th><th>0.5</th><th>0.6</th><th>0.7</th><th>0.8</th><th>0.9</th><th>1</th></tr>
<tr><th>0</th><td rowspan="3">🟨 0.307</td><td rowspan="3">🟨 0.284</td><td rowspan="3">🟨 0.307</td><td rowspan="4">▨ 0.296</td><td rowspan="4">▨ 0.334</td><td rowspan="4" colspan="6">▨ 0.330</td></tr>
<tr><th>0.1</th></tr>
<tr><th>0.2</th></tr>
<tr><th>0.3</th><td colspan="3">🟨 0.305</td></tr>
<tr><th>0.4</th><td rowspan="7" colspan="3">🟦 0.284</td><td rowspan="7">0.3</td><td colspan="7">🟥 0.370</td></tr>
<tr><th>0.5</th><td rowspan="6">0.4</td><td colspan="6">🟥 0.441</td></tr>
<tr><th>0.6</th><td rowspan="5" colspan="6">🟥 0.496</td></tr>
<tr><th>0.7</th></tr>
<tr><th>0.8</th></tr>
<tr><th>0.9</th></tr>
<tr><th>1</th></tr>
</table>

Legend:

▨	$W_{L,1} = 0.05\ F_s$
🟥	$V_{21} = 0$
🟦	$V_5 = 5F_s$
🟨	double-soft solution

Figure 5.18: $x_{N,1}$ value at the solution for Column A in Flowsheet 2.



$x_{N,1,\text{spec}}$ for Column A

$x_{N,1,\text{spec}}$ for Column B

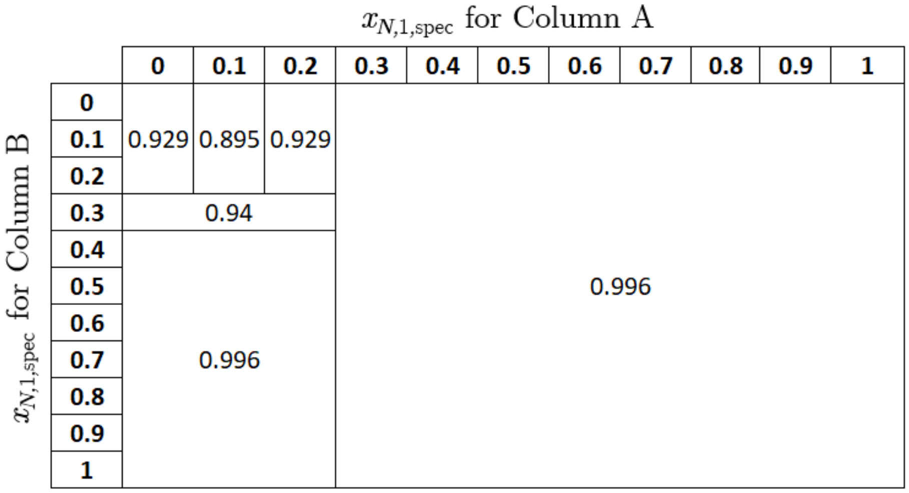
<table>
<tr><th></th><th>0</th><th>0.1</th><th>0.2</th><th>0.3</th><th>0.4</th><th>0.5</th><th>0.6</th><th>0.7</th><th>0.8</th><th>0.9</th><th>1</th></tr>
<tr><th>0</th><td rowspan="3">▨ 0.323</td><td rowspan="3">▨ 0.298</td><td rowspan="3">▨ 0.323</td><td rowspan="4">▨ 0.311</td><td rowspan="4">▨ 0.351</td><td rowspan="4" colspan="6">▨ 0.347</td></tr>
<tr><th>0.1</th></tr>
<tr><th>0.2</th></tr>
<tr><th>0.3</th><td colspan="3">▨ 0.321</td></tr>
<tr><th>0.4</th><td colspan="11">0.4</td></tr>
<tr><th>0.5</th><td colspan="11">0.5</td></tr>
<tr><th>0.6</th><td rowspan="5" colspan="4">🟦 0.578</td><td rowspan="5" colspan="7">🟦 0.577</td></tr>
<tr><th>0.7</th></tr>
<tr><th>0.8</th></tr>
<tr><th>0.9</th></tr>
<tr><th>1</th></tr>
</table>

Legend:

▨	$W_{L,1} = 0.05\ F_s$
🟦	$L_{N\text{-}1} = 5F_s$

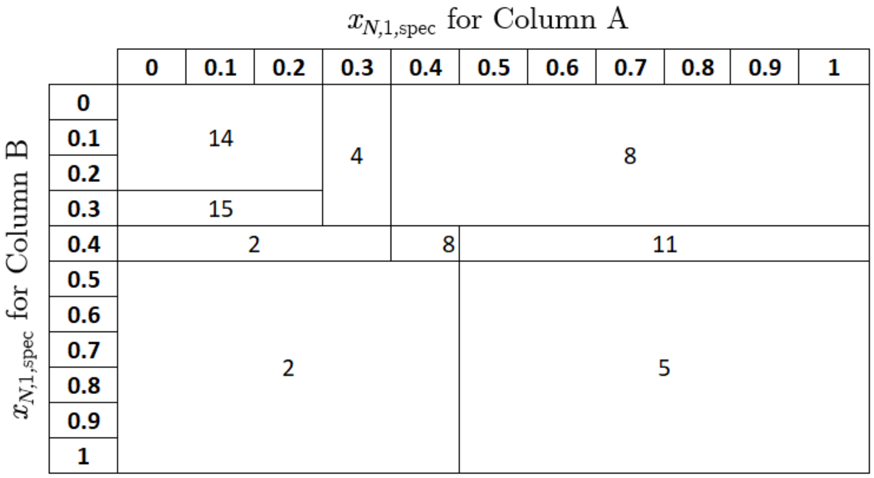
Figure 5.19: $x_{N,1}$ value at the solution for Column B in Flowsheet 2.

these range from 0.284 to 0.496 in Column A and from 0.298 to 0.578 in Column B. In Figures 5.18 and 5.19 we see essentially three types of flowsheet solutions with respect to



		$x_{N,1,\text{spec}}$ for Column A										
		0	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1**
$x_{N,1,\text{spec}}$ for Column B	**0**	0.929	0.895	0.929	0.996							
	0.1											
	0.2											
	0.3	0.94										
	0.4	0.996										
	0.5											
	0.6											
	0.7											
	0.8											
	0.9											
	1											

Figure 5.20: $x_{1,1}$ value at the solution for Column A in Flowsheet 2.

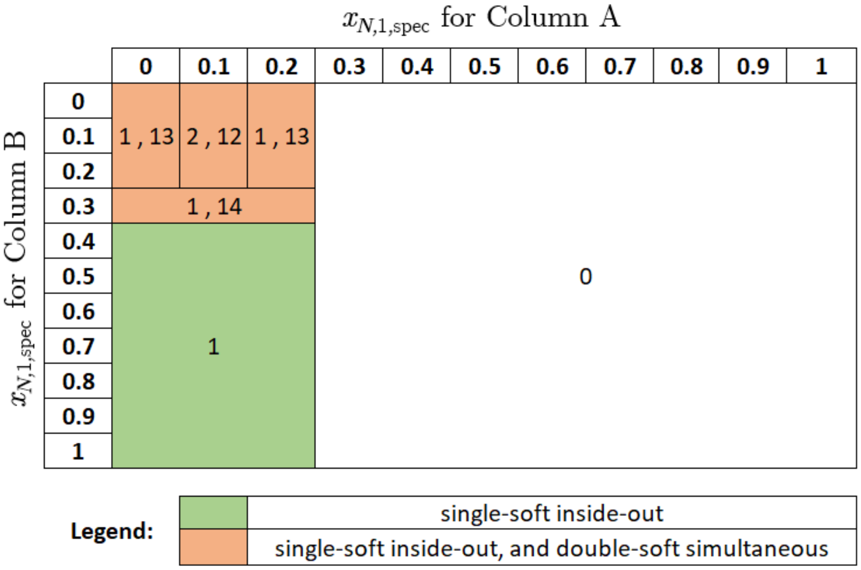


		$x_{N,1,\text{spec}}$ for Column A										
		0	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1**
$x_{N,1,\text{spec}}$ for Column B	**0**	14			4	8						
	0.1											
	0.2											
	0.3	15										
	0.4	2				8	11					
	0.5	2					5					
	0.6											
	0.7											
	0.8											
	0.9											
	1											

Figure 5.21: Total number of iterations to converge Flowsheet 2.

the model that is enforced for each of the columns: feasible/feasible, single-soft/feasible, and single-soft/double-soft. Column A exhibits two types of minimum flow rate solutions: one for which $W_{L,1} = r_{\min}F_s = 0.05F_s$ and one in which the internal flow rate V_{21} is equal to zero, while Column B exhibits only the former type.

The test cases with $0 \leq x_{N,1,\text{spec}} \leq 0.2$ for Column A and $0 \leq x_{N,1,\text{spec}} \leq 0.3$ for Column B lead to a double-soft solution for Column A, in which $x_{1,1,\text{spec}} = 0.996$ is not enforced, and to a single-soft $\lambda_{r_{\min}}$ solution for Column B. However, in Figure 5.20 we see that the $x_{1,1}$ values at these double-soft solutions are still somewhat close to the specified



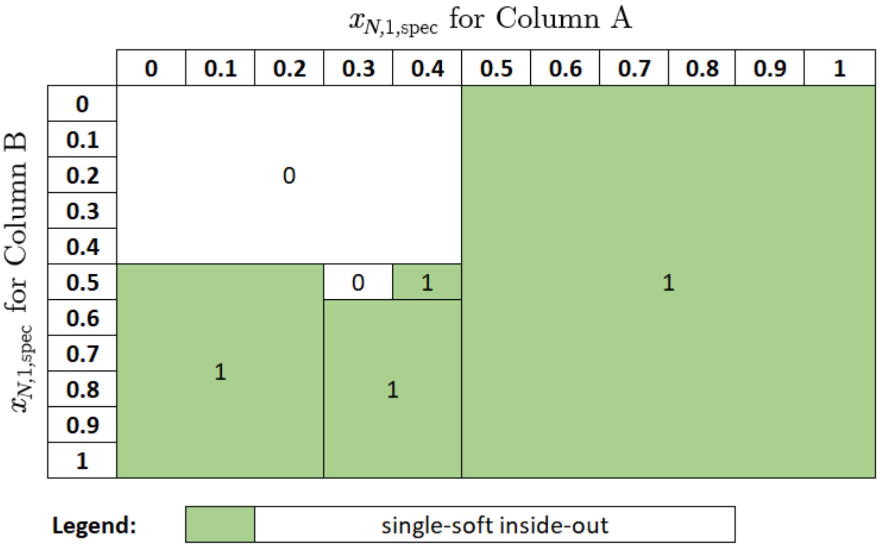
$x_{N,1,\text{spec}}$ for Column A

$x_{N,1,\text{spec}}$ for Column B	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0	1 , 13	2 , 12	1 , 13	0							
0.1											
0.2											
0.3	1 , 14										
0.4	1										
0.5											
0.6											
0.7											
0.8											
0.9											
1											

Legend:

🟩	single-soft inside-out
🟧	single-soft inside-out, and double-soft simultaneous

Figure 5.22: Number of flowsheet iterations in which each advanced distillation model was needed for Column A in Flowsheet 2.



$x_{N,1,\text{spec}}$ for Column A

$x_{N,1,\text{spec}}$ for Column B	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0	0					1					
0.1											
0.2											
0.3											
0.4											
0.5	1			0	1						
0.6				1							
0.7											
0.8											
0.9											
1											

Legend:

🟩	single-soft inside-out

Figure 5.23: Number of flowsheet iterations in which each advanced distillation model was needed for Column B in Flowsheet 2.

value of 0.996. As discussed in Section 4.5.6, the double-soft model has a two-dimensional continuum of solutions due to the removal of two specification equations. The model might converge to any of these solutions depending on the initial guess and its closeness to a MESH-feasible state. In the context of converging a flowsheet we simulate each column with a sequence of progressively more accurate initial guesses, which allows us to steer the double-soft model in the direction of approximately enforcing desired values for one or more specifications. At each of the double-soft solutions, Column A is neither at the minimum nor the maximum flow rate solutions, though for some test cases it is close to the former or to both of them simultaneously. This last scenario ($\lambda_{r_{\max}} \approx \lambda_{r_{\min}}$) corresponds to the composite feed composition of Column A being close to its feasibility limit determined by the relevant azeotrope.

The fact that we obtained flowsheet solutions in which the double-soft model ended up being enforced for Column A does not necessarily mean that no flowsheet solution exists in which the single-soft model is enforced instead. As previously mentioned, with our four-tier strategy the flowsheet pass function becomes discontinuous once we switch from the single to the double-soft models in any of the columns. Also, the final double-soft solution obtained depends on the path described by the flowsheet iterations. Since low values of $x_{N,1,\text{spec}}$ tend to lead to maximum and minimum flow rate solutions in Columns A and B, respectively, we could expect to be able to obtain a single-soft/single-soft solution with $x_{N,1} \approx 0.284$ in Column A and $x_{N,1} \approx 0.3$ in Column B instead of the double-soft/single-soft solutions.

To investigate these double-soft test cases further, we attempted to re-simulate them by setting $r_{\min} = 0$ for both columns; however, that resulted in the four-tier modeling strategy failing to converge Column B in an intermediate flowsheet iteration. If instead we decrease $r_{\min} = 0$ only for Column A and keep $r_{\min} = 0.05$ for Column B, we are able to successfully converge to different flowsheet solutions. By specifying $0 \leq x_{N,1,\text{spec}} \leq 0.2$ in both Columns A and B we obtain the same single-soft/single-soft solution in which $x_{N,1} = 0.284 = \lambda_{r_{\max}}$ and $W_{L,1} = 0.048F_s$ for Column A, and $x_{N,1} = 0.298 = \lambda_{r_{\min}}$ for Column B. Since $W_{L,1} = 0.048F_s < 0.05F_s$ at the Column A solution, we can be confident that no single-soft/single-soft solution exists for the original test cases, in which we enforce $W_{L,1} \geq 0.05F_s$ for both columns. If we specify $0 \leq x_{N,1,\text{spec}} \leq 0.2$ in Column A and $x_{N,1,\text{spec}} = 0.3$ in Column B, we obtain a single-soft/feasible solution, in which

$x_{N,1} = 0.284 = \lambda_{r_{\max}}$ and $W_{L,1} = 0.053F_s$ for Column A and $x_{N,1} = 0.3$ for Column B. This solution would satisfy the $W_{L,1} \geq 0.05F_s$ requirement in the original test cases, therefore it would be the most adequate one for the latter instead of the double-soft/single-soft solutions obtained in Figures 5.22 and 5.23. These examples illustrate that there is a compromise in choosing a value of $r_{\min}$. On the one hand, a zero or near-zero value might preclude model convergence and allow external product flow rates $W_{L,1}, L_N$ to be zero or near-zero, which is undesirable whether these streams are final products or inputs to other equipment. On the other hand, higher $r_{\min}$ values might change $\lambda_{r_{\min}}$ enough so that double-soft solutions end up being obtained instead of single-soft ones.

In Figures 5.22 and 5.23 we see that, for most test cases, the (single-soft) inside-out algorithm was required to converge at least one of the columns in one (in most cases, the first) flowsheet iteration. This is expected given that the columns are converged starting from a "blind" initial guess in the first flowsheet pass. Subsequent passes start from the previously converged solutions and thus the columns can be succcessfully and rapidly solved with the simultaneous algorithm. In Flowsheet 1 we observed test cases in which the double-soft model was required only in the first flowsheet iteration due to a feed composition being temporarily infeasible. On the other hand, for the Flowsheet 2 test cases in which the double-soft model was required for Column A, this same model had to be utilized in almost all flowsheet iterations up to the last one, yielding a double-soft final solution for Column A. For these same test cases we observe an elevated number of flowsheet iterations in Figure 5.21. This makes sense due to "looseness" of the double-soft model, which allows the column state to vary in between flowsheet iterations much more than the single-soft model does. Consequently, more flowsheet passes are needed to stabilize the state of Column A.

Test cases in Aspen Plus

With the original specifications $x_{1,1} = 0.996$, $x_{N,1} = 0.3$ for Column A and $x_{1,1} = 0.004$, $x_{N,1} = 0.54$ for Column B, Aspen Plus converges Flowsheet 2 without any provided initial guess for the tear stream, i.e., with an initial guess of $F_{\text{tear}} = 0$. If we change only the bottom product specification for Column B to $x_{N,1} = 0.577$, we are no longer able to converge Flowsheet 2 without an initial guess and we get a "RadFrac not converged in 25 outer loop iterations" error message for Column B. Next, as done for Flowsheet 1, we initialize the tear stream with the correct values for all variables except F_{tear}. We are only able to converge the flowsheet to the correct solution, for which $F_{\text{tear}} = 53.4$ kmol/h, if we provide an initial guess with $45 \leq F_{\text{tear}} \leq 52$ kmol/h or $60 \leq F_{\text{tear}} \leq 65$ kmol/h. Note that providing the actual solution value for all tear stream variables as an initial guess results in convergence failure.

We are able to converge the flowsheet for $x_{N,1} = 0.578$ with our modeling strategy by setting $r_{\max} = 7$ for Column B. During the flowsheet computations we only need to use the single-soft inside-out method once, for Column B. Conversely, Aspen Plus does not converge the flowsheet in this case regardless of the initial guess for the tear stream state, including if we provide all the correct final values as determined with our model. We are also unable to converge the flowsheet in Aspen Plus even if we slowly increase $x_{N,1}$ from 0.577 to 0.578 in 0.0001 increments while retaining previous simulation results to initialize the flowsheet.

5.7. Conclusions

In this chapter we have presented a nonsmooth version of the inside-out method to simulate distillation columns, which is suitable for converging both the single-soft and double-soft adaptive models from Chapter 4. As illustrated with several test cases for the single-soft model, the inside-out algorithm is significantly more reliable to converge the model equations from an *ab initio* starting point compared to the simultaneous method of solving the whole set of equations at once. However, the former algorithm is slower than the latter for mixtures with few components. We have also developed a four-tier modeling strategy to simulate distillation columns when converging flowsheets with recycle streams. This strategy makes use of a sequence of up to four distillation simulation methods, combining the single-soft and double-soft models and the inside-out and simultaneous algorithms in order of increasing reliability and decreasing speed. We have demonstrated that this flowsheeting strategy is reliable in converging two pressure-swing distillation processes even under infeasible (and unreasonable) column specifications and using a poor initial guess for the tear stream state (i.e., zero flow rate). In particular, the double-soft model allows us to proceed with flowsheet calculations even when two column specifications are infeasible during intermediate iterations. Additionally, we presented both single-column and flowsheet test cases in which our distillation simulation methods outperform Aspen Plus' Radfrac model, despite the process specifications being fully feasible.

Chapter 6

Lipschitz and Piecewise-differentiable Rank Theorems

In this chapter we present a piecewise-differentiable (PC^r) Rank Theorem and extend a previously stated Lipschitz Rank Theorem, with the goal of characterizing the level sets of nonsmooth functions $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$. When the appropriate conditions are satisfied by the generalized derivatives of $\mathbf{f}$, the Rank Theorems allow us to express a given level set $\mathbf{f}^{-1}(\mathbf{c}) \subset \mathbb{R}^n$ locally as the graph of a nonsmooth function, within a homeomorphic transformation of the same class as $\mathbf{f}$. We define PC^r and Lipschitz submersions, immersions and maps of constant rank in terms of the most general conditions under which the corresponding Rank Theorems are applicable. Moreover, we develop sufficient conditions that are more easily verifiable for practical applications and relate them to existing full-rank conditions from the literature.

6.1. Introduction

The present chapter together with Chapter 7 aim to characterize the local structure of the level sets $\mathbf{f}^{-1}(\mathbf{c}) \subset \mathbb{R}^n$ of locally Lipschitz continuous (Lipschitz for short) or piecewise-differentiable (PC^r) functions $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$. In Chapter 7 we will use the Rank Theorems developed in the present chapter to characterize these level sets as nonsmooth manifolds, according to the definitions thereof that we will introduce, and provide example applications.

The (smooth) Rank Theorem, a traditional result in differential topology, is based on the Inverse Function Theorem and guarantees that the level set of a constant rank function between manifolds is itself an embedded submanifold of the domain. In this chapter we will present Lipschitz and PC^r Rank Theorems (Theorem 6.4.2) whose results and proof structure are analogous to the "Euclidean version" of the smooth Rank Theorem. In Chapter 7 we will extend these theorems to the case of functions between nonsmooth manifolds.

A Euclidean Lipschitz Rank Theorem has been presented before in the literature [5] and there was an attempt to generalize it in [17]; Clarke's Inverse Function Theorem was used as a basis in both instances. Ours is the first PC^r Rank Theorem to the best of our knowledge, and it relies on the PC^r Inverse Function Theorem from [66, 72]. Unlike the smooth case, there isn't a single clear choice of how to define a Lipschitz or PC^r function of "constant rank". Our Definition 6.3.2 is phrased with general and fairly abstract conditions, relying on the existence of homeomorphisms that can transform the function into the format required within the Rank Theorem proof. This choice of definition can be justified by the fact that equivalent or even sufficient rank conditions involving only the generalized derivatives of the function don't seem to exist. In particular, we discuss this in detail for the Lipschitz case in order to clarify a mistake in [17]. Nevertheless, more concrete sufficient conditions can be stated for Lipschitz functions under specific cases.

We will also present concrete and verifiable sufficient rank conditions for a special class of PC^r functions, which can be applied to the nonsmooth distillation model from [19] as detailed in Chapter 7.

Given that every PC^r function is locally Lipschitz continuous, Lipschitz Rank Theorems are also applicable to PC^r functions. However, the PC^r Rank Theorem is useful in its own right because the Lipschitz assumptions are both too restrictive and likely impractical to verify for PC^r functions. Further, as discussed in Chapter 7, the Lipschitz Rank Theorem applied to a PC^r function will only allow us to conclude that its level set is a Lipschitz manifold, not necessarily a PC^r manifold.

6.2. Background concepts and notation

We use brackets [ ] to include text that can replace its previous counterpart, according to the relevant context, and parentheses ( ) to include text that can be added to the sentence. Neighborhoods are taken to be open sets in the relevant topology. We defined the projections $\boldsymbol{\pi}^n_m : \mathbb{R}^n \to \mathbb{R}^m$, $\boldsymbol{\rho}^n_{n-m} : \mathbb{R}^n \to \mathbb{R}^{n-m}$ and the inclusion $\boldsymbol{\iota}^m_n : \mathbb{R}^m \to \mathbb{R}^n$ in Definition 2.2.9. The range and null space of a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ are denoted $\mathcal{R}(\mathbf{A})$ and $\mathcal{N}(\mathbf{A})$, respectively. The orthogonal complement of a subspace $V \subset W$ of a vector space W is denoted $V^{\perp}$. The identity map on a set A is denoted id_A. Slices of open subsets of $\mathbb{R}^n$ were presented in Definition 2.2.7. We refer the reader to Chapter 2 for definitions of all the other background concepts used in this chapter.

For ease of notation, in this chapter and in Chapter 7 we will call a function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ simply "Lipschitz" or Lipschitz at $\mathbf{x}_0 \in \mathbb{R}^n$ if it is locally Lipschitz continuous [at $\mathbf{x}_0$] in the standard sense, and we will call it "globally Lipschitz" on a set $U \subset \mathbb{R}^n$ if it is Lipschitz continuous on U in the standard sense.

In this chapter and in Chapter 7, $\mathcal{G}$ stands for a generic category or "class" of functions; in this thesis we will be considering $\mathcal{G} = C^r, PC^r$, Lipschitz.

6.2.1. Homeomorphisms and Inverse Function Theorems

The concept of local $\mathcal{G}$ homeomorphisms and the existing C^r, PC^r and Lipschitz Inverse Function Theorems were described in Section 2.2. We now summarize the main results of that section which will be relevant within this chapter.

We can state conditions for $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ to be a local $\mathcal{G}$ homeomorphism at $\mathbf{x}_0 \in \mathbb{R}^n$ using an appropriately chosen $\mathcal{G}$ Inverse Function Theorem. For $\mathcal{G} = C^r$, invertibility of $\mathbf{Jf}(\mathbf{x}_0)$ is a necessary and sufficient condition. For $\mathcal{G} =$ Lipschitz, Clarke regularity at $\mathbf{x}_0$ is a sufficient condition, while Kummer's necessary and sufficient conditions are stated in terms of the generalized "Thibault" derivative. Clarke's condition, though much more commonly used in the literature, is not a necessary condition even in the case of piecewise-linear functions. Nevertheless, in this chapter we have chosen to use Clarke's rather than Kummer's Inverse Function Theorem within the Lipschitz Rank Theorem proof. This allows us to state rank conditions in terms of the more standard and less abstract Clarke generalized derivatives, at the cost of these conditions being less general to some extent.

For $\mathcal{G} = PC^r$, a necessary and sufficient condition is that $\mathbf{f}$ be coherently oriented at $\mathbf{x}_0$ and $d\mathbf{f}_{\mathbf{x}_0} : \mathbb{R}^n \to \mathbb{R}^n$ be invertible, while a sufficient condition is that $\mathbf{f}$ be completely coherently oriented at $\mathbf{x}_0$. For PC^r functions the necessary and sufficient PC^r conditions are equivalent to Kummer's condition. The advantage of the former is that it utilizes the B-subdifferential, a finite set with a concrete representation for PC^r functions in terms of essentially active C^r functions. As with Lipschitz functions, Clarke regularity is a sufficient but not necessary condition.

6.3. Submersions, immersions, and maps of constant rank

We start by recalling the usual C^r constant rank, submersion and immersion definitions to provide the rationale behind our corresponding Lipschitz and PC^r definitions. We say the C^r function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ is a C^r map of constant rank $k \leq m, n$ around $\mathbf{x}_0 \in \mathbb{R}^n$ if there exists a neighborhood $U \subset \mathbb{R}^n$ of $\mathbf{x}_0$ such that $\mathbf{Jf}(\mathbf{x}) \in \mathbb{R}^{m\times n}$ has rank k for all $\mathbf{x} \in U$. We say $\mathbf{f}$ is a C^r submersion [immersion] at $\mathbf{x}_0$ if $\mathbf{Jf}(\mathbf{x}_0) \in \mathbb{R}^{m\times n}$ is full-row rank and $m \leq n$ [full-column rank and $m \geq n$].

To prove the C^r Rank Theorem, first we must transform $\mathbf{f}$ such that the same $k \times k$ submatrix of the Jacobian matrix stays invertible. To this end, we can use the following proposition.

Proposition 6.3.1. $\mathbf{f}$ is a C^r map of constant rank k around $\mathbf{x}_0$ if and only if there exist permutation linear homeomorphisms $\mathbf{g}_1 : \mathbb{R}^n \to \mathbb{R}^n$, $\mathbf{g}_2 : \mathbb{R}^m \to \mathbb{R}^m$ such that the leading $k\times k$ submatrix of $\mathbf{J}(\mathbf{g}_2\circ\mathbf{f}\circ\mathbf{g}_1)(\mathbf{x}) \in \mathbb{R}^{m\times n}$ is invertible for every $\mathbf{x}$ on some neighborhood $V_1 \subset \mathbb{R}^n$ of $\tilde{\mathbf{x}}_0 = {\mathbf{g}_1}^{-1}(\mathbf{x}_0)$. Moreover, a C^r submersion [immersion] at $\mathbf{x}_0$ is a C^r map of constant rank $k = m$ [$k = n$] around $\mathbf{x}_0$.

Proof. The results follow from the lower semicontinuity of the rank, the fact $\mathbf{f}$ is at least C^1, and the invertibility of the Jacobians of $\mathbf{g}_1, \mathbf{g}_2$. □

Given this equivalence, we do not need to assume the existence of the homeomorphisms $\mathbf{g}_1, \mathbf{g}_2$ to prove the C^r Rank Theorem. Instead, we use the far more concrete condition in terms of the rank of $\mathbf{Jf}(\mathbf{x})$. However, in the Lipschitz and PC^r cases we do not have such equivalent and concrete conditions solely in terms of the generalized derivatives of $\mathbf{f}$. For this reason, our nonsmooth constant rank definitions rely on the *existence* of homeomorphisms $\mathbf{g}_1, \mathbf{g}_2$ that can transform $\mathbf{f}$ into the required format within the Rank Theorem proof. Later in this section, we discuss several more concrete conditions that are sufficient (but not necessary) for a Lischitz or PC^r function to have constant rank in this sense.

In the following three definitions, let $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ be a PC^r [Lipschitz] function at $\mathbf{x}_0 \in \mathbb{R}^n$.

Definition 6.3.2 (*PC^r* **/ Lipschitz constant rank map).** We say $\mathbf{f}$ is a PC^r [Lipschitz] map of constant rank $k \leq m, n$ around $\mathbf{x}_0$ with respect to the PC^r [Lipschitz] homeomorphisms $\mathbf{g}_1 : \mathbb{R}^n \to \mathbb{R}^n$, $\mathbf{g}_2 : \mathbb{R}^m \to \mathbb{R}^m$ if there exists a neighborhood $V \subset \mathbb{R}^n$ of $\tilde{\mathbf{x}}_0 = {\mathbf{g}_1}^{-1}(\mathbf{x}_0)$ such that the following holds for the function $\mathbf{F} = \mathbf{g}_2 \circ \mathbf{f} \circ \mathbf{g}_1$:

PC^r case: (1) every matrix in $\partial^B \mathbf{F}(\mathbf{x})$ has rank k for every $\mathbf{x} \in V$,

(2) $\boldsymbol{\pi}^m_k \circ \mathbf{F}$ is coherently oriented with respect to the first k variables at $\tilde{\mathbf{x}}_0$,

(3) $d(\boldsymbol{\pi}^m_k \circ \mathbf{F})_{\tilde{\mathbf{x}}_0}(\cdot, \mathbf{v}) : \mathbb{R}^k \to \mathbb{R}^k$ is invertible $\forall \mathbf{v} \in \mathbb{R}^{n-k}$.

Lipschitz case: every matrix in $\partial \mathbf{F}(\mathbf{x})$ has rank k and its leading $k \times k$ submatrix is invertible for every $\mathbf{x} \in V$.

Definition 6.3.3 (*PC^r* **/ Lipschitz submersion).** We say $\mathbf{f}$ is a PC^r [Lipschitz] submersion at $\mathbf{x}_0$ with respect to the PC^r [Lipschitz] homeomorphism $\mathbf{g}_1 : \mathbb{R}^n \to \mathbb{R}^n$ if $m \leq n$ and the following holds for the function $\mathbf{F} = \mathbf{f} \circ \mathbf{g}_1$, where $\tilde{\mathbf{x}}_0 = {\mathbf{g}_1}^{-1}(\mathbf{x}_0)$:

PC^r case: $\mathbf{F}$ is coherently oriented with respect to the first m variables at $\tilde{\mathbf{x}}_0$, and $d\mathbf{F}_{\tilde{\mathbf{x}}_0}(\cdot, \mathbf{v}) : \mathbb{R}^m \to \mathbb{R}^m$ is invertible $\forall \mathbf{v} \in \mathbb{R}^{n-m}$.

Lipschitz case: $\mathbf{F}$ is Clarke regular with respect to the first m variables at $\tilde{\mathbf{x}}_0$.

Definition 6.3.4 (*PC^r* **/ Lipschitz immersion).** We say $\mathbf{f}$ is a PC^r [Lipschitz] immersion at $\mathbf{x}_0$ with respect to the PC^r [Lipschitz] homeomorphism $\mathbf{g}_2 : \mathbb{R}^m \to \mathbb{R}^m$ if $m \geq n$ and the following holds for the function $\mathbf{F} = \mathbf{g}_2 \circ \mathbf{f}$:

PC^r case: $\boldsymbol{\pi}^m_n \circ \mathbf{F}$ is coherently oriented at $\mathbf{x}_0$ and $d(\boldsymbol{\pi}^m_n \circ \mathbf{F})_{\mathbf{x}_0} : \mathbb{R}^n \to \mathbb{R}^n$ is invertible.

Lipschitz case: $\boldsymbol{\pi}^m_n \circ \mathbf{F}$ is Clarke regular at $\mathbf{x}_0$.

We say $\mathbf{f}$ is a $\mathcal{G}$ map of constant rank k [$\mathcal{G}$ immersion] [$\mathcal{G}$ submersion] if it is so around [at] every $\mathbf{x}_0 \in \mathbb{R}^n$.

Remark 6.3.5. In the previous three definitions, the homeomorphisms $\mathbf{g}_1, \mathbf{g}_2$ need only be local; that is, they may be of the form $\mathbf{g}_1 : V_1 \to W_1$ and $\mathbf{g}_2 : V_2 \to W_2$, with $V_1, W_1 \in \mathbb{R}^n$ and $V_2, W_2 \in \mathbb{R}^m$ open, and $\mathbf{x}_0 \in W_1$, $\mathbf{f}(\mathbf{x}_0) \in V_2$.

Proposition 6.3.6 shows how the constant rank conditions become equivalent to the submersion/immersion definitions in the full rank case. Therefore, for our purposes we can focus on nonsmooth Rank Theorems without having to consider Submersion and Immersion Theorems separately.

Proposition 6.3.6. Let $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ be a PC^r/Lipschitz function. $\mathbf{f}$ is a PC^r/Lipschitz submersion [immersion] at $\mathbf{x}_0$ w.r.t. $\mathbf{g}_1$ [$\mathbf{g}_2$] if and only if $\mathbf{f}$ is a PC^r/Lipschitz map of constant rank m [n] around $\mathbf{x}_0$ w.r.t. $\mathbf{g}_1, \mathrm{id}_{\mathbb{R}^m}$ [$\mathrm{id}_{\mathbb{R}^n}, \mathbf{g}_2$].

Proof. The converse statements follow immediately from the definitions, so we consider only the direct statements. Let $k = m$ [$k = n$] and $\mathbf{F}$ be the function described in Definition 6.3.3 [Definition 6.3.4]. In the immersion case, for convenience of notation let $\tilde{\mathbf{x}}_0 = \mathbf{x}_0$.

<u>PC^r case</u>: Let $\{\mathbf{F}_{(i)} : V \to \mathbb{R}^m : \ i \in I\}$ be a set of essentially active selection functions for $\mathbf{F}$ at $\tilde{\mathbf{x}}_0$, where $V \subset V_1$ is a neighborhood of $\tilde{\mathbf{x}}_0$. From the coherent orientation assumption, every matrix in $\partial^B \mathbf{F}(\tilde{\mathbf{x}}_0)$ has full rank k, thus each $\mathbf{JF}_{(i)}(\tilde{\mathbf{x}}_0) \in \partial^B \mathbf{F}(\tilde{\mathbf{x}}_0)$ is full rank. By lower-semicontinuity of the rank and continuous differentiability of the finitely many $\mathbf{F}_{(i)}$, we can shrink the neighborhood V such that $\mathbf{JF}_{(i)}(\mathbf{x})$ stays full rank $\forall \mathbf{x} \in V$ and $\forall i \in I$. Since $\{\mathbf{F}_{(i)} : \ i \in I\}$ is a valid set of selection functions $\forall \mathbf{x} \in V$, $\partial^B \mathbf{F}(\mathbf{x}) \subset \left\{\mathbf{JF}_{(i)}(\mathbf{x}) : \ i \in I\right\}$ and thus every matrix in $\partial^B \mathbf{F}(\mathbf{x})$ has full rank k $\forall \mathbf{x} \in V$.

<u>Lipschitz case</u>: By assumption every matrix in $\partial \mathbf{F}(\tilde{\mathbf{x}}_0)$ has full rank k. Since the Clarke Jacobian of a Lipschitz function is upper semicontinuous (Proposition 2.6.2 c) in [22]) and given the lower semicontinuity of the rank, we can find a neighborhood $V \subset V_1$ of $\tilde{\mathbf{x}}_0$ such that every matrix in $\partial \mathbf{F}(\mathbf{x})$ has full rank k for every $\mathbf{x} \in V$.

Both cases: Letting $\mathbf{g}_2 = id_{\mathbb{R}^m}$ $[\mathbf{g}_1 = id_{\mathbb{R}^n}]$, we satisfy Definition 6.3.2 with $\mathbf{F} = \mathbf{g}_2 \circ \mathbf{F}$ $[\mathbf{F} = \mathbf{F} \circ \mathbf{g}_1]$ and $k = m$ $[k = n]$. □

Proposition 6.3.7 addresses the relationship between the PC^r and Lipschitz rank conditions in the case of a PC^r function. The converse of this result is not true, e.g., if all matrices in the B-subdifferential have positive determinants then one of their convex combinations might have determinant equal to zero. Therefore, the PC^r rank conditions are more general than the Lipschitz ones for PC^r functions.

Proposition 6.3.7. Let $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ be a PC^r function. If $\mathbf{f}$ is a Lipschitz map of constant rank k [submersion] [immersion] around [at] $\mathbf{x}_0 \in \mathbb{R}^n$ with respect to PC^r homeomorphisms, then $\mathbf{f}$ is a PC^r map of constant rank k [submersion] [immersion] around [at] $\mathbf{x}_0$ with respect to the same homeomorphisms.

Proof. Let $\mathbf{f}$ have constant rank k according to the Lipschitz Definition 6.3.2, where $\mathbf{g}_1, \mathbf{g}_2$ are PC^r. Condition (1) of the PC^r definition holds since $\partial^B \mathbf{F}(\mathbf{x}) \subset \partial \mathbf{F}(\mathbf{x})$. The fact that all leading $k \times k$ submatrices in $\partial(\boldsymbol{\pi}_k^m \circ \mathbf{F})(\tilde{\mathbf{x}}_0)$ are invertible is sufficient for invertibility of the Lipschitz function $d(\boldsymbol{\pi}_k^m \circ \mathbf{F})_{\tilde{\mathbf{x}}_0}(\cdot, \mathbf{v})$ and thus condition (3) holds (see proof of Proposition 6.3.18). Finally, suppose $\mathbf{M}_1, \mathbf{M}_2$ are leading $k \times k$ submatrices in $\partial^B(\boldsymbol{\pi}_k^m \circ \mathbf{F})(\tilde{\mathbf{x}}_0)$ such that $\det(\mathbf{M}_1)\det(\mathbf{M}_2) < 0$. By continuity of the determinant, there must exist $\lambda \in (0, 1)$ such that $\mathbf{M} = \lambda \mathbf{M}_1 + (1 - \lambda)\mathbf{M}_2$ is a leading $k \times k$ submatrix in $\partial(\boldsymbol{\pi}_k^m \circ \mathbf{F})(\tilde{\mathbf{x}}_0)$ and $\det(\mathbf{M}) = 0$, a contradiction. Therefore, condition (2) of the constant rank PC^r definition holds with the same $\mathbf{F}$. The result for a submersion/immersion then follows from Proposition 6.3.6. □

6.3.1. Sufficient rank conditions for Lipschitz functions

In order to introduce sufficient conditions for a Lipschitz function to have constant rank, first we consider the following definitions from [24] which relate to our Lipschitz submersion definition. Note that analogous definitions could also be stated for the full column rank case, relating to Lipschitz immersions.

Definition 6.3.8 (FRA and SFRA). Let $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ be Lipschitz at $\mathbf{x}_0 \in \mathbb{R}^n$ and $m \leq n$. The full rank assumption (FRA) is said to hold at $\mathbf{x}_0$ if every matrix in $\partial\mathbf{f}(\mathbf{x}_0)$ has full row rank m. The strong full rank assumption (SFRA) is said to hold at $\mathbf{x}_0$ if there exists an m-dimensional subspace $E \subset \mathbb{R}^n$ such that

$$\mathcal{N}(\mathbf{A}) \cap E = \{\mathbf{0}\}, \quad \forall \mathbf{A} \in \partial\mathbf{f}(\mathbf{x}_0). \tag{6.1}$$

Remark 6.3.9. The SFRA implies the FRA, given that the dimension of each $\mathcal{N}(\mathbf{A})$ is at least $n - m$ from the fact $m \leq n$ and at most $n - m$ from Equation 6.1. However, we know the converse is not true from the following counterexample presented by Izmailov in [41]:

$$\mathcal{A} = \operatorname{conv}\left\{ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} -0.2 & 0.8 & 0.2 \\ 0 & -0.5 & 0.4 \end{bmatrix}, \begin{bmatrix} 0.7 & -0.7 & -0.6 \\ -0.8 & 0.1 & 0.6 \end{bmatrix} \right\}. \tag{6.2}$$

The set $\mathcal{A}$ satisfies FRA but not SFRA (see Appendix A for a detailed demonstration), and it must correspond to the Clarke Jacobian of some Lipschitz function according to [7]. In [24] the authors hypothesize the equivalence of FRA and SFRA for PC^1 functions ("full-rank conjecture"), but prove the result only for min-type functions $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ (i.e., each component f_i is the minimum of two C^1 functions) with $m \leq 3$.

Now we introduce the following definitions as natural extensions of the FRA and SFRA for non-full row rank cases.

Definition 6.3.10 (CRA and SCRA). Let $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ be Lipschitz at $\mathbf{x}_0 \in \mathbb{R}^n$ and $k \leq m, n$. The constant rank assumption (CRA) is said to hold at $\mathbf{x}_0$ with rank k if there exists a neighborhood $U \subset \mathbb{R}^n$ of $\mathbf{x}_0$ such that every matrix in $\partial\mathbf{f}(\mathbf{x})$ has rank k for every $\mathbf{x} \in U$. The strong constant rank assumption (SCRA) is said to hold at $\mathbf{x}_0$ with rank k if, additionally, there exist k-dimensional subspaces $E \subset \mathbb{R}^n$ and $H \subset \mathbb{R}^m$ and a neighborhood $U \subset \mathbb{R}^n$ of $\mathbf{x}_0$ such that

$$\mathcal{N}(\mathbf{A}) \cap E = \{\mathbf{0}\} \text{ and } \mathcal{N}(\mathbf{A}^{\mathrm{T}}) \cap H = \{\mathbf{0}\}, \quad \forall \mathbf{A} \in \partial \mathbf{f}(\mathbf{x}),\ \mathbf{x} \in U. \tag{6.3}$$

Remark 6.3.11. Due to Lemma 6.3.12, the SCRA is equivalent to the existence of an $(n-k)$-dimensional subspace $E^{\perp} \subset \mathbb{R}^n$ and an $(m-k)$-dimensional subspace $H^{\perp} \subset \mathbb{R}^m$ such that

$$\mathcal{R}(\mathbf{A}^{\mathrm{T}}) \cap E^{\perp} = \{\mathbf{0}\} \text{ and } \mathcal{R}(\mathbf{A}) \cap H^{\perp} = \{\mathbf{0}\}, \quad \forall \mathbf{A} \in \partial \mathbf{f}(\mathbf{x}),\ \mathbf{x} \in U. \tag{6.4}$$

Lemma 6.3.12. *Let $V, W \subset \mathbb{R}^n$ be subspaces such that $\dim(V) + \dim(W) = n$.*

$V \cap W = \{\mathbf{0}\}$ if and only if $V^{\perp} \cap W^{\perp} = \{\mathbf{0}\}$.

Proof. Let $\mathbf{v} \in V^{\perp} \cap W^{\perp}$ and $Z = \operatorname{span}\{\mathbf{v}\} \subset \mathbb{R}^n$. Since every vector in V and in W is orthogonal to $\mathbf{v}$ and Z has at most dimension 1, then $V, W \subset Z^{\perp} \subset \mathbb{R}^n$. Given that $\dim(V)+\dim(W) = n$ and $V \cap W = \{\mathbf{0}\}$, we must have $\dim(Z^{\perp}) \geq n$, therefore $Z^{\perp} = \mathbb{R}^n$ and $\mathbf{v} = \mathbf{0}$. □

The next proposition establishes that the CRA/SCRA indeed generalize the FRA/SFRA in the full row rank case.

Proposition 6.3.13. *Let $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ be Lipschitz at $\mathbf{x}_0 \in \mathbb{R}^n$ and $m \leq n$.*

FRA [SFRA] holds at $\mathbf{x}_0$ if and only if CRA [SCRA] holds at $\mathbf{x}_0$ with full row rank m.

Proof. The CRA⇒FRA and SCRA⇒SFRA statements follow directly from the definitions; in the latter, SCRA must hold with $H = \mathbb{R}^n$ given that CRA holds with full row rank. FRA⇒CRA follows from upper semicontinuity of the Clarke Jacobian and lower semicontinuity of the rank.

From SFRA (see Remark 6.3.11) there exists an m-dimensional subspace $E \subset \mathbb{R}^n$ such that $\mathcal{R}(\mathbf{A}^{\mathrm{T}}) \cap E^{\perp} = \{\mathbf{0}\}$ for every $\mathbf{A} \in \partial f(\mathbf{x}_0)$. Given $\mathbf{A} \in \partial f(\mathbf{x}_0)$, the minimal angle

$\theta \in [0, \pi/2]$ between $\mathcal{R}(\mathbf{A}^{\mathrm{T}})$ and $E^{\perp}$ is a continuous function of $\mathbf{A}$ because its cosine corresponds to the 2-norm of the matrix $\mathbf{P}_{\mathcal{R}(\mathbf{A}^{\mathrm{T}})}\mathbf{P}_{E^{\perp}}$, where $\mathbf{P}_V$ denotes the orthogonal projection matrix onto the subspace $V \subset \mathbb{R}^n$ (see [39]). Moreover, $\theta > 0$ if and only if $\mathcal{R}(\mathbf{A}^{\mathrm{T}}) \cap E^{\perp} = \{\mathbf{0}\}$. Then, given the upper semicontinuity of the Clarke Jacobian, we can find a neighborhood $U \subset \mathbb{R}^n$ of $\mathbf{x}_0$ where Equation 6.4 holds with $H = \mathbb{R}^m$ and thus the SCRA applies with $k = m$.

□

Propositions 6.3.16 and 6.3.17 show that the SCRA [SFRA] is equivalent to the function being a Lipschitz map of constant rank [submersion] in the specific case where the homeomorphisms are linear and orthogonal. On the other hand, our Lipschitz constant rank and submersion definitions allow for the homeomorphisms to be only Lipschitz.

Lemma 6.3.14. Let $\mathbf{A} \in \mathbb{R}^{m \times n}$, $k \leq m, n$, and $\mathbf{v}_1, \ldots, \mathbf{v}_k \in \mathbb{R}^n$ be a basis of the subspace $E \subset \mathbb{R}^n$. $\mathcal{N}(\mathbf{A}) \cap E = \{\mathbf{0}\}$ if and only if the vectors $\mathbf{A}\mathbf{v}_1, \ldots, \mathbf{A}\mathbf{v}_k \in \mathbb{R}^m$ are linearly independent.

Proof. Considering the direct statement, let $\alpha_i \in \mathbb{R}$ such that

$$\sum_{i=1}^{k} \alpha_i \mathbf{A}\mathbf{v}_i = \mathbf{A}\left(\sum_{i=1}^{k} \alpha_i \mathbf{v}_i\right) = \mathbf{0}.$$

Then $\mathbf{v} = \sum_{i=1}^{k} \alpha_i \mathbf{v}_i$ belongs to $\mathcal{N}(\mathbf{A}) \cap E$, therefore $\mathbf{v} = \mathbf{0}$. From linear independence of the $\mathbf{v}_i$ it follows that all $\alpha_i = 0$. Now for the converse statement, let $\mathbf{v} \in \mathcal{N}(\mathbf{A}) \cap E$. There exist $\alpha_1, \ldots, \alpha_k \in \mathbb{R}$ such that $\mathbf{v} = \sum_{i=1}^{k} \alpha_i \mathbf{v}_i$ and $\mathbf{A}\mathbf{v} = \sum_{i=1}^{k} \alpha_i \mathbf{A}\mathbf{v}_i = \mathbf{0}$. From linear independence of the $\mathbf{A}\mathbf{v}_i$ it follows that all $\alpha_i = 0$ and $\mathbf{v} = \mathbf{0}$. □

Lemma 6.3.15. Let $\mathbf{A} \in \mathbb{R}^{m \times n}$, $k \leq n$, and $\mathbf{P} \in \mathbb{R}^{m \times m}$ be invertible. The first k columns of $\mathbf{A}$ are linearly independent if and only if the first k columns of $\mathbf{PA}$ are linearly independent.

Proof. Let $\mathbf{a}_i \in \mathbb{R}^m$ denote the i-th column of $\mathbf{A}$ and $c_1, \ldots, c_k \in \mathbb{R}$. The result follows from

$$\textstyle\sum_{i=1}^{k} c_i(\mathbf{P}\mathbf{a}_i) = \mathbf{0} \iff \mathbf{P}\left(\sum_{i=1}^{k} c_i\mathbf{a}_i\right) = \mathbf{0} \iff \sum_{i=1}^{k} c_i\mathbf{a}_i = \mathbf{0}. \tag{6.5}$$

□

Proposition 6.3.16. Let $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ be Lipschitz at $\mathbf{x}_0 \in \mathbb{R}^n$. SCRA holds at $\mathbf{x}_0$ with rank $k \leq m, n$ if and only if $\mathbf{f}$ is a Lipschitz map of constant rank k around $\mathbf{x}_0$ with respect to orthogonal linear homeomorphisms $\mathbf{g}_1 : \mathbb{R}^n \to \mathbb{R}^n$, $\mathbf{g}_2 : \mathbb{R}^m \to \mathbb{R}^m$.

Proof. Let SCRA hold according to Definition 6.3.10. Let $\mathbf{v}_1, \ldots, \mathbf{v}_k \in \mathbb{R}^n$ and $\mathbf{v}_{k+1}, \ldots, \mathbf{v}_n \in \mathbb{R}^n$ be orthonormal bases for E and $E^{\perp}$, respectively, and $\mathbf{y}_1, \ldots, \mathbf{y}_k \in \mathbb{R}^m$ and $\mathbf{y}_{k+1}, \ldots, \mathbf{y}_m \in \mathbb{R}^m$ be orthonormal bases for H and $H^{\perp}$, respectively. Further, let $\mathbf{P}_1 \in \mathbb{R}^{n\times n}$ be the matrix whose j-th column is $\mathbf{v}_j \in \mathbb{R}^n$, and $\mathbf{P}_2 \in \mathbb{R}^{m\times m}$ be the matrix whose i-th row is $\mathbf{y}_i \in \mathbb{R}^m$. Define the orthogonal linear homeomorphisms $\mathbf{g}_1 : \mathbb{R}^n \to \mathbb{R}^n$, $\mathbf{g}_1(\mathbf{x}) = \mathbf{P}_1\mathbf{x}$, and $\mathbf{g}_2 : \mathbb{R}^m \to \mathbb{R}^m$, $\mathbf{g}_2(\mathbf{x}) = \mathbf{P}_2\mathbf{x}$, and let $\mathbf{F} = \mathbf{g}_2 \circ \mathbf{f} \circ \mathbf{g}_1$.

Letting $\mathbf{x} \in U$ and $\mathbf{A} \in \partial\mathbf{f}(\mathbf{x})$, we can apply Lemma 6.3.14 to show that the first k columns of $\mathbf{A}\mathbf{P}_1 \in \mathbb{R}^{m\times n}$ are linearly independent. Because $\mathbf{P}_1 \in \mathbb{R}^{n\times n}$ is invertible, $\mathcal{R}(\mathbf{A}\mathbf{P}_1) = \mathcal{R}(\mathbf{A})$, thus $\mathcal{N}((\mathbf{A}\mathbf{P}_1)^{\mathrm{T}}) = \mathcal{N}(\mathbf{A}^{\mathrm{T}})$. Then $\mathcal{N}((\mathbf{A}\mathbf{P}_1)^{\mathrm{T}}) \cap H = \{\mathbf{0}\}$ and we can apply Lemma 6.3.14 to see that the first k rows of $\mathbf{P}_2\mathbf{A}\mathbf{P}_1 \in \mathbb{R}^{m\times n}$ are linearly independent. From Lemma 6.3.15 we can conclude $\mathbf{P}_2\mathbf{A}\mathbf{P}_1$ has rank k with invertible leading $k \times k$ submatrix for every $\mathbf{A} \in \partial\mathbf{f}(\mathbf{x})$ and every $\mathbf{x} \in U$. Given that $\mathbf{P}_1, \mathbf{P}_2$ are invertible we have that $\partial\mathbf{F}(\mathbf{x}) = \mathbf{P}_2\partial\mathbf{f}(\mathbf{g}_1(\mathbf{x}))\mathbf{P}_1$ for every $\mathbf{x} \in V$, where $V = \mathbf{g}_1{}^{-1}(U) \subset \mathbb{R}^n$ is a neighborhood of $\tilde{\mathbf{x}}_0 = \mathbf{g}_1{}^{-1}(\mathbf{x}_0)$ (see Proposition 2.1.3). Then $\mathbf{f}$ is a Lipschitz map of constant rank k around $\mathbf{x}_0$ with respect to $\mathbf{g}_1, \mathbf{g}_2$.

For the converse, let $\mathbf{f}$ and $\mathbf{F}$ be according to Definition 6.3.2 and $\mathbf{g}_1(\mathbf{x}) = \mathbf{M}_1\mathbf{x}$, $\mathbf{g}_2(\mathbf{x}) = \mathbf{M}_2\mathbf{x}$, where $\mathbf{v}_j \in \mathbb{R}^n$ denotes the j-th column of $\mathbf{M}_1$ and $\mathbf{y}_i \in \mathbb{R}^m$ denotes the i-th row of $\mathbf{M}_2$. Further, let $U = \mathbf{g}_1(V)$, $E = \operatorname{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ and $H = \operatorname{span}\{\mathbf{y}_1, \ldots, \mathbf{y}_k\}$.

Letting $\mathbf{x} \in U$ and $\mathbf{A} \in \partial\mathbf{f}(\mathbf{x})$, we have that $\mathbf{M}_2\mathbf{A}\mathbf{M}_1 \in \partial\mathbf{F}(\mathbf{g}_1{}^{-1}(\mathbf{x}))$ has rank k with leading $k \times k$ submatrix invertible. Then CRA follows from invertibility of $\mathbf{M}_1, \mathbf{M}_2$.

Lemma 6.3.15 gives that $\mathbf{A}\mathbf{v}_1, \ldots, \mathbf{A}\mathbf{v}_k$ (the first k columns of $\mathbf{A}\mathbf{M}_1$) and $\mathbf{A}_1^{\mathrm{T}}\mathbf{y}_1, \ldots, \mathbf{A}_k^{\mathrm{T}}\mathbf{y}_k$ (the first k rows of $\mathbf{M}_2\mathbf{A}$) are linearly independent. Then Lemma 6.3.14 shows that Equation 6.3 holds. □

Proposition 6.3.17. Let $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ be Lipschitz at $\mathbf{x}_0 \in \mathbb{R}^n$ and $m \leq n$. SFRA holds at $\mathbf{x}_0$ if and only if $\mathbf{f}$ is a Lipschitz submersion at $\mathbf{x}_0$ with respect to an orthogonal linear homeomorphism $\mathbf{g}_1 : \mathbb{R}^n \to \mathbb{R}^n$.

Proof. From Proposition 6.3.13 we know that SFRA holds at $\mathbf{x}_0$ if and only if SCRA holds at $\mathbf{x}_0$ with $H = \mathbb{R}^n$. From Proposition 6.3.16 the latter is equivalent to $\mathbf{f}$ being a Lipschitz map of constant rank m around $\mathbf{x}_0$ w.r.t. $\mathbf{g}_1$ and $\mathbf{g}_2 = id_{\mathbb{R}^m}$, where $\mathbf{g}_1$ is a linear orthogonal homeomorphism. Finally, this is equivalent to $\mathbf{f}$ being a Lipschitz submersion at $\mathbf{x}_0$ w.r.t. $\mathbf{g}_1$ from Proposition 6.3.6. □

6.3.2. Sufficient rank conditions for PC^r functions

The following proposition provides a sufficient condition for $\mathbf{f}$ to be a PC^r map of constant rank k, an immersion ($k = n$), or a submersion ($k = m$), in terms of complete coherent orientation of the transformed function $\mathbf{F} = \mathbf{g}_2 \circ \mathbf{f} \circ \mathbf{g}_1$. As with the PC^r Inverse Function Theorem, complete coherent orientation implies both coherent orientation and invertibility of the relevant piecewise-affine map.

Proposition 6.3.18. Let $V_1 \subset \mathbb{R}^n$ be open, $\mathbf{F} : V_1 \to \mathbb{R}^m$ be a PC^r function, and $k \leq m, n$. If $\boldsymbol{\pi}_k^m \circ \mathbf{F}$ is completely coherently oriented with respect to the first k variables at $\tilde{\mathbf{x}}_0 \in V_1$, then $d(\boldsymbol{\pi}_k^m \circ \mathbf{F})_{\tilde{\mathbf{x}}_0}(\cdot, \mathbf{v}) : \mathbb{R}^k \to \mathbb{R}^k$ is invertible for every $\mathbf{v} \in \mathbb{R}^{n-k}$.

Proof. Let $\mathbf{v} \in \mathbb{R}^{n-k}$ and $\mathbf{h}_{\mathbf{v}} = d(\boldsymbol{\pi}_k^m \circ \mathbf{F})_{\tilde{\mathbf{x}}_0}(\cdot, \mathbf{v})$. For every $\mathbf{u} \in \mathbb{R}^k$ we have

$$\mathbf{h}_{\mathbf{v}}(\mathbf{u}) = d(\boldsymbol{\pi}_k^m \circ \mathbf{F})_{\tilde{\mathbf{x}}_0}(\mathbf{u}, \mathbf{v}) \in \left\{ \mathbf{M} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} : \mathbf{M} \in \partial^B(\boldsymbol{\pi}_k^m \circ \mathbf{F})(\tilde{\mathbf{x}}_0) \right\} \tag{6.6}$$

$$\subset \left\{ \mathbf{M}_1\mathbf{u} + \mathbf{M}_2\mathbf{v} : \mathbf{M}_1 \in \boldsymbol{\pi}_k^B(\boldsymbol{\pi}_k^m \circ \mathbf{F})(\tilde{\mathbf{x}}_0),\ \mathbf{M}_2 \in \boldsymbol{\rho}_{n-k}^B(\boldsymbol{\pi}_k^m \circ \mathbf{F})(\tilde{\mathbf{x}}_0) \right\}. \tag{6.7}$$

Therefore, $\mathbf{h}_\mathbf{v}$ is continuous and piecewise-affine, and $\partial^B \mathbf{h}_\mathbf{v}(\mathbf{u})$ is a subset of $\boldsymbol{\pi}_k^B(\boldsymbol{\pi}_k^m \circ \mathbf{F})(\tilde{\mathbf{x}}_0)$ for every $\mathbf{u} \in \mathbb{R}^k$. Then $\mathbf{h}_\mathbf{v}$ is completely coherently oriented everywhere and thus invertible (Corollary 19 in [72]). □

Theorem 6.3.19 gives concrete and more easily verifiable sufficient conditions for certain PC^r functions, with at most two essentially active functions, to be PC^r submersions or maps of constant rank. This result can be applied to the distillation model from [19], as will be detailed in Chapter 7.

Theorem 6.3.19 (*PC^r* **maps with 2 selection functions**)**.** Suppose the PC^r function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ has a selection set at $\mathbf{x}_0 \in \mathbb{R}^n$ composed of two C^r functions $\mathbf{f}_{(1)}, \mathbf{f}_{(2)} : U \to \mathbb{R}^m$, where $U \subset \mathbb{R}^n$ is a neighborhood of $\mathbf{x}_0$. Moreover, suppose $\mathbf{Jf}_{(1)}(\mathbf{x}_0) - \mathbf{Jf}_{(2)}(\mathbf{x}_0)$ has rank 1.

(a) Let $m = n - 1$. If both $\mathbf{Jf}_{(1)}(\mathbf{x}_0)$ and $\mathbf{Jf}_{(2)}(\mathbf{x}_0)$ have full row rank and their null spaces do not coincide, then $\mathbf{f}$ is a PC^r submersion at $\mathbf{x}_0$ with respect to an orthogonal linear homeomorphism.

(b) Let $m = n$ and suppose the C^r functions $\mathbf{f}_{(1)}, \mathbf{f}_{(2)}$ have constant rank $n - 1$ on U. If the 1-dimensional null spaces of $\mathbf{Jf}_{(1)}(\mathbf{x}_0)$ and $\mathbf{Jf}_{(2)}(\mathbf{x}_0)$ do not coincide, and the 1-dimensional left null spaces of $\mathbf{Jf}_{(1)}(\mathbf{x}_0)$ and $\mathbf{Jf}_{(2)}(\mathbf{x}_0)$ do not coincide, then $\mathbf{f}$ is a PC^r map of constant rank $n-1$ around $\mathbf{x}_0$ with respect to orthogonal linear homeomorphisms.

Proof. Both cases: Let $\mathbf{M} = \mathbf{Jf}_{(1)}(\mathbf{x}_0) - \mathbf{Jf}_{(2)}(\mathbf{x}_0)$. For matrices $\mathbf{A}, \mathbf{B}$ of the same dimensions such that $\mathcal{N}(\mathbf{A}) \cap \mathcal{N}(\mathbf{B}) = \{\mathbf{0}\}$ it holds that

$$\mathbf{x} \in \mathcal{N}(\mathbf{A}),\ \mathbf{x} \neq \mathbf{0} \Rightarrow \mathbf{Ax} = \mathbf{0},\ \mathbf{Bx} \neq \mathbf{0} \Rightarrow \mathbf{x} \notin \mathcal{N}(\mathbf{A} - \mathbf{B}), \tag{6.8}$$

therefore

$$\mathcal{N}(\mathbf{Jf}_{(i)}(\mathbf{x}_0)) \ \cap\ \mathcal{N}(\mathbf{M}) = \{\mathbf{0}\}, \quad i = 1, 2. \tag{6.9}$$

Let $\mathbf{v}_1, \ldots, \mathbf{v}_{n-1} \in \mathbb{R}^n$ be an orthonormal basis of $\mathcal{N}(\mathbf{M})$ and $\mathbf{v}_n \in \mathbb{R}^n$ be an orthonormal basis of $\mathcal{R}(\mathbf{M}^{\mathrm{T}})$, and $\mathbf{P}_1 \in \mathbb{R}^{n \times n}$ be the matrix whose j-th column is $\mathbf{v}_j$. Then we apply Lemma 6.3.14 and conclude the first $n-1$ columns of $\mathbf{Jf}_{(i)}(\mathbf{x}_0)\mathbf{P}_1$ are linearly independent for $i = 1, 2$. Define the orthogonal linear homeomorphism $\mathbf{g}_1 : \mathbb{R}^n \to \mathbb{R}^n$, $\mathbf{g}_1(\mathbf{x}) = \mathbf{P}_1\mathbf{x}$, and let $V = \mathbf{g}_1{}^{-1}(U)$ and $\tilde{\mathbf{x}}_0 = \mathbf{g}_1{}^{-1}(\mathbf{x}_0)$.

Case (a): By composition, $\mathbf{F} = \mathbf{f} \circ \mathbf{g}_1$ is a PC^r function with selection set

$$\left\{\mathbf{f}_i \circ \mathbf{g}_1 : V \to \mathbb{R}^{n-1},\ i = 1, 2\right\} \tag{6.10}$$

at $\tilde{\mathbf{x}}_0$. Applying the Chain Rule to each $\mathbf{f}_i \circ \mathbf{g}_1$ at $\tilde{\mathbf{x}}_0$,

$$\partial^B \mathbf{F}(\tilde{\mathbf{x}}_0) \subset \left\{\mathbf{Jf}_{(1)}(\mathbf{x}_0)\mathbf{P}_1,\ \ \mathbf{Jf}_{(2)}(\mathbf{x}_0)\mathbf{P}_1\right\}. \tag{6.11}$$

Since $\mathbf{M}\left[\mathbf{v}_1, \ldots, \mathbf{v}_{n-1}\right] = \mathbf{0}$, the first $n-1$ columns of $\mathbf{Jf}_{(1)}(\mathbf{x}_0)\mathbf{P}_1$ and $\mathbf{Jf}_{(2)}(\mathbf{x}_0)\mathbf{P}_1$ coincide. Then $\mathbf{F}$ is completely coherently oriented w.r.t. the first $n-1$ variables at $\tilde{\mathbf{x}}_0$, and by Proposition 6.3.18 $\mathbf{f}$ is a PC^r submersion at $\mathbf{x}_0$ w.r.t. $\mathbf{g}_1$.

Case (b): Since $\mathbf{P}_1$ is invertible, the column spaces and therefore also the left null spaces of $\mathbf{Jf}_{(i)}(\mathbf{x}_0)$ and $\mathbf{Jf}_{(i)}(\mathbf{x}_0)\mathbf{P}_1$ coincide, thus

$$\mathcal{N}((\mathbf{Jf}_{(i)}(\mathbf{x}_0)\mathbf{P}_1)^{\mathrm{T}})\ \cap\ \mathcal{N}(\mathbf{M}^{\mathrm{T}}) = \{\mathbf{0}\}, \quad i = 1, 2. \tag{6.12}$$

Let $\mathbf{y}_1, \ldots, \mathbf{y}_{n-1} \in \mathbb{R}^n$ be an orthonormal basis of $\mathcal{N}(\mathbf{M}^{\mathrm{T}})$ and $\mathbf{y}_n \in \mathbb{R}^n$ be an orthonormal basis of $\mathcal{R}(\mathbf{M})$, and $\mathbf{P}_2 \in \mathbb{R}^{m \times m}$ be the matrix whose i-th row is $\mathbf{y}_j$. Then we apply Lemma 6.3.14 using $\mathbf{A} = (\mathbf{Jf}_{(i)}(\mathbf{x}_0)\mathbf{P}_1)^{\mathrm{T}}$ to see that the first $n-1$ rows of $\mathbf{P}_2\mathbf{Jf}_{(i)}(\mathbf{x}_0)\mathbf{P}_1$ are linearly independent for $i = 1, 2$. Applying Lemma 6.3.15 we conclude that the leading $(n-1) \times (n-1)$ submatrix of $\mathbf{P}_2\mathbf{Jf}_{(i)}(\mathbf{x}_0)\mathbf{P}_1$ is invertible for $i = 1, 2$. Since $\mathbf{M}\left[\mathbf{v}_1, \ldots, \mathbf{v}_{n-1}\right] = \mathbf{0}$ and $\left[\mathbf{y}_1, \ldots, \mathbf{y}_{n-1}\right]^{\mathrm{T}} \mathbf{M} = \mathbf{0}^{\mathrm{T}}$, the leading $(n-1) \times (n-1)$

submatrices of $\mathbf{P}_2\mathbf{Jf}_{(1)}(\mathbf{x}_0)\mathbf{P}_1$ and $\mathbf{P}_2\mathbf{Jf}_{(2)}(\mathbf{x}_0)\mathbf{P}_1$ coincide:

$$\Big[\mathbf{y}_1,\ldots,\mathbf{y}_{n-1}\Big]^{\mathrm{T}}\mathbf{Jf}_{(1)}(\mathbf{x}_0)\Big[\mathbf{v}_1,\ldots,\mathbf{v}_{n-1}\Big]=\Big[\mathbf{y}_1,\ldots,\mathbf{y}_{n-1}\Big]^{\mathrm{T}}\mathbf{Jf}_{(2)}(\mathbf{x}_0)\Big[\mathbf{v}_1,\ldots,\mathbf{v}_{n-1}.\Big] \tag{6.13}$$

Define the orthogonal linear homeomorphism $\mathbf{g}_2 : \mathbb{R}^n \to \mathbb{R}^n$, $\mathbf{g}_2(\mathbf{x}) = \mathbf{P}_2\mathbf{x}$. By composition, the function $\mathbf{F} = \mathbf{g}_2 \circ \mathbf{f} \circ \mathbf{g}_1 : V \to \mathbb{R}^n$ is PC^r with selection set

$$\{\mathbf{g}_2 \circ \mathbf{f}_i \circ \mathbf{g}_1 : V \to \mathbb{R}^n,\ i = 1, 2\} \tag{6.14}$$

at $\tilde{\mathbf{x}}_0$. Applying the Chain Rule to each $\mathbf{g}_2 \circ \mathbf{f}_i \circ \mathbf{g}_1$ at every $\mathbf{x} \in V$ gives

$$\partial^B\mathbf{F}(\mathbf{x}) \subset \left\{\mathbf{P}_2\mathbf{Jf}_{(1)}(\mathbf{g}_1(\mathbf{x}))\mathbf{P}_1,\quad \mathbf{P}_2\mathbf{Jf}_{(2)}(\mathbf{g}_1(\mathbf{x}))\mathbf{P}_1\right\},\quad \mathbf{x} \in V. \tag{6.15}$$

Then, by Equation 6.13, $\boldsymbol{\pi}^n_{n-1} \circ\ \mathbf{F} : V \to \mathbb{R}^{n-1}$ is completely coherently oriented with respect to the first $n-1$ variables at $\tilde{\mathbf{x}}_0$. Further, every matrix in $\left\{\partial^B\mathbf{F}(\mathbf{x}) :\ \mathbf{x} \in V\right\}$ has rank $n-1$, given that $\mathbf{P}_1, \mathbf{P}_2$ are invertible and $\mathbf{f}_{(1)}, \mathbf{f}_{(2)}$ have constant rank $n-1$ on $U = \mathbf{g}_1(V)$. By Proposition 6.3.18 we conclude $\mathbf{f}$ is a PC^r map of constant rank $n-1$ around $\mathbf{x}_0$ with respect to $\mathbf{g}_1, \mathbf{g}_2$. □

6.4. Nonsmooth Rank Theorems

In this section we present a unified and detailed proof of the C^r, Lipschitz, and PC^r Rank Theorems to highlight their common general outline and clarify their distinctions, although only the PC^r proof is truly novel.

The standard C^r Rank Theorem result corresponds only to part (a) of Theorem 6.4.2 and can be found, for instance, in [54]. Parts (b) and (c) are consequences of intermediary results within the proof of (a), and therefore are best presented as separate results to be used in Chapter 7. Moreover, part (b) provides a way to prove the Implicit Function

Theorem directly from the Rank Theorem. Parts (b) and (c) have not been presented for nonsmooth functions in the more general context of *transformed* graphs; in [5] they were presented for Lipschitz functions that do not require any transformation (i.e., the leading $k \times k$ submatrices in the Clarke Jacobian of the original function are invertible).

In [17] Butler et al. presented a Lipschitz Rank Theorem which stated, using the notation from this chapter, that the CRA (Definition 6.3.10) guarantees Theorem 6.4.2(a) holds for Lipschitz functions. This theorem relied on their Proposition 2.1, which corresponds to the statement CRA $\Rightarrow$ SCRA for compact convex subsets of matrices. However, this result is incorrect. The counterexample in Remark 6.3.9 shows that FRA $\not\Rightarrow$ SFRA for compact convex subsets of matrices, therefore CRA $\not\Rightarrow$ SCRA (see Proposition 6.3.13).

We present a corrected Lipschitz Rank Theorem based on the more general concept of a Lipschitz map of constant rank stated in Definition 6.3.2. Our proof also makes use of the fact that the Clarke Jacobian can be taken w.r.t. any measure zero set that includes the non-differentiability set (Equation 2.4). Otherwise, our Lipschitz proof is mostly analogous to the one in [17].

First, we define the concept of $\mathcal{G}$ charts for $\mathbb{R}^n$. Here, $\mathcal{G}$ stands for C^r, PC^r, or Lipschitz.

Definition 6.4.1 (**$\mathcal{G}$ chart for $\mathbb{R}^n$**)**.** We say (U, ϕ) is a $\mathcal{G}$ chart for $\mathbb{R}^n$ around $\mathbf{x}_0$ if $U, \phi(U) \subset \mathbb{R}^n$ are open sets, $\mathbf{x}_0 \in U$, and $\phi : U \to \phi(U)$ is a $\mathcal{G}$ homeomorphism.

Theorem 6.4.2 (**C^r/Lipschitz/PC^r Rank Theorem**)**.** *Let $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ be a $\mathcal{G}$ function, $\mathbf{x}_0 \in \mathbb{R}^n$ and $\mathbf{f}(\mathbf{x}_0) = \mathbf{c}$. Suppose that $\mathbf{f}$ is a $\mathcal{G}$ map of constant rank k around $\mathbf{x}_0$ with respect to the $\mathcal{G}$ homeomorphisms $\mathbf{g}_1 : \mathbb{R}^n \to \mathbb{R}^n$ and $\mathbf{g}_2 : \mathbb{R}^m \to \mathbb{R}^m$, where $k \leq m, n$.*

Then there exist $\mathcal{G}$ charts $(U, \boldsymbol{\phi})$ for $\mathbb{R}^n$ around $\mathbf{x}_0$ and $(V, \boldsymbol{\psi})$ for $\mathbb{R}^m$ around $\mathbf{f}(\mathbf{x}_0)$, with $\mathbf{f}(U) \subset V$, such that:

$$\textit{(a)} \quad \boldsymbol{\psi} \circ \mathbf{f} \circ \boldsymbol{\phi}^{-1} = \boldsymbol{\iota}_m^k \circ \boldsymbol{\rho}_k^n : \boldsymbol{\phi}(U) \to \mathbb{R}^m, \tag{6.16}$$

$$(x_1, \dots, x_k, x_{k+1}, \dots, x_n) \mapsto (x_1, \dots, x_k, 0, \dots, 0).$$

(b) $\boldsymbol{\phi}(U \cap \mathbf{f}^{-1}(\mathbf{c}))$ is an $(n-k)$-slice of $\boldsymbol{\phi}(U) \subset \mathbb{R}^n$, and there exists a $\mathcal{G}$ function $\mathbf{y} : V' \to \mathbb{R}^k$, $V' \subset \mathbb{R}^{n-k}$ open, such that ${\mathbf{g}_1}^{-1}(U \cap \mathbf{f}^{-1}(\mathbf{c})) \subset \mathbb{R}^n$ is (within permutation) the graph of $\mathbf{y}$.

(c) $\boldsymbol{\psi}(\mathbf{f}(U))$ is a k-slice of $\boldsymbol{\psi}(V) \subset \mathbb{R}^m$, and there exists a $\mathcal{G}$ function $\mathbf{z} : U' \to \mathbb{R}^{m-k}$, $U' \subset \mathbb{R}^k$ open, such that $\mathbf{g}_2(\mathbf{f}(U)) \subset \mathbb{R}^m$ is the graph of $\mathbf{z}$.

Proof. **(a)** Let $\mathbf{F} = \mathbf{g}_2 \circ \mathbf{f} \circ \mathbf{g}_1 : V_1 \to \mathbb{R}^m$ be the $\mathcal{G}$ function described in Proposition 6.3.1 for $\mathcal{G} = C^r$ and in Definition 6.3.2 for $\mathcal{G} = PC^r$ or Lipschitz, with $\tilde{\mathbf{x}}_0 = {\mathbf{g}_1}^{-1}(\mathbf{x}_0)$. Next, define the $\mathcal{G}$ function $\mathbf{h} : V_1 \to \mathbb{R}^n$ as

$$\mathbf{h}(\mathbf{x}) = \big(\boldsymbol{\pi}_k^m \circ \mathbf{F}(\mathbf{x}),\ x_{k+1}, \dots, x_n\big). \tag{6.17}$$

$\mathbf{h}$ is differentiable at $\mathbf{x} \in V_1$ if and only if $\boldsymbol{\pi}_k^m \circ \mathbf{F}$ is differentiable at $\mathbf{x} \in V_1$, in which case

$$\mathbf{Jh}(\mathbf{x}) = \begin{bmatrix} \mathbf{A}(\mathbf{x}) & \mathbf{B}(\mathbf{x}) \\ \mathbf{0}_{(n-k)\times k} & \mathbf{I}_{n-k} \end{bmatrix}, \quad \mathbf{J}(\boldsymbol{\pi}_k^m \circ \mathbf{F})(\mathbf{x}) = \begin{bmatrix} \mathbf{A}(\mathbf{x}) & \mathbf{B}(\mathbf{x}) \end{bmatrix}, \tag{6.18}$$

where $\mathbf{A}(\mathbf{x})$ is the leading $k \times k$ submatrix of $\mathbf{J}(\boldsymbol{\pi}_k^m \circ \mathbf{F})(\mathbf{x})$, and thus $\det(\mathbf{Jh}(\mathbf{x})) = \det(\mathbf{A}(\mathbf{x}))$.

$\underline{C^r \text{ case}}$: By the assumptions on $\mathbf{F}$, $\mathbf{A}(\mathbf{x})$ and thus $\mathbf{Jh}(\mathbf{x})$ is invertible $\forall \mathbf{x} \in V_1$.

<u>PC^r case</u>: For any $(\mathbf{u},\mathbf{v}),(\mathbf{w},\mathbf{z}) \in \mathbb{R}^k \times \mathbb{R}^{n-k}$ we have

$$d\mathbf{h}_{\tilde{\mathbf{x}}_0}(\mathbf{u},\mathbf{v}) = \begin{bmatrix} d(\boldsymbol{\pi}_k^m \circ \mathbf{F})_{\tilde{\mathbf{x}}_0}(\mathbf{u},\mathbf{v}) \\ \mathbf{v} \end{bmatrix}, \tag{6.19}$$

$$d\mathbf{h}_{\tilde{\mathbf{x}}_0}(\mathbf{u},\mathbf{v}) = (\mathbf{w},\mathbf{z}) \iff d(\boldsymbol{\pi}_k^m \circ \mathbf{F})_{\tilde{\mathbf{x}}_0}(\mathbf{u},\mathbf{v}) = \mathbf{w} \text{ and } \mathbf{v} = \mathbf{z}. \tag{6.20}$$

For every $\mathbf{z} \in \mathbb{R}^{n-k}$, the function $\mathbf{r}_{\mathbf{z}} = d(\boldsymbol{\pi}_k^m \circ \mathbf{F})_{\tilde{\mathbf{x}}_0}(\cdot,\mathbf{z}) : \mathbb{R}^k \to \mathbb{R}^k$ is invertible by assumption. Therefore $d\mathbf{h}_{\tilde{\mathbf{x}}_0} : \mathbb{R}^n \to \mathbb{R}^n$ is invertible and $(d\mathbf{h}_{\tilde{\mathbf{x}}_0})^{-1}(\mathbf{w},\mathbf{z}) = (\mathbf{r}_{\mathbf{z}}^{-1}(\mathbf{w}),\mathbf{z})$.

In view of Equation 6.18 and the definition of the B-subdifferential, we must have

$$\partial^B \mathbf{h}(\tilde{\mathbf{x}}_0) = \left\{ \begin{bmatrix} \mathbf{M} & \mathbf{N} \\ \mathbf{0}_{(n-k)\times k} & \mathbf{I}_{n-k} \end{bmatrix}, \quad \begin{bmatrix} \mathbf{M} & \mathbf{N} \end{bmatrix} \in \partial^B(\boldsymbol{\pi}_k^m \circ \mathbf{F})(\tilde{\mathbf{x}}_0) \right\}. \tag{6.21}$$

By assumption $\boldsymbol{\pi}_k^m \circ \mathbf{F}$ is coherently oriented w.r.t. the first k variables at $\tilde{\mathbf{x}}_0$, therefore $\mathbf{h}$ is coherently oriented at $\tilde{\mathbf{x}}_0$.

<u>Lipschitz case</u>: In view of Equation 6.21 and given that every leading $k \times k$ submatrix in $\partial \mathbf{F}(\tilde{\mathbf{x}}_0)$ is invertible by assumption,

$$\partial \mathbf{h}(\tilde{\mathbf{x}}_0) = \left\{ \begin{bmatrix} \mathbf{M} & \mathbf{N} \\ \mathbf{0}_{(n-k)\times k} & \mathbf{I}_{n-k} \end{bmatrix}, \quad \begin{bmatrix} \mathbf{M} & \mathbf{N} \end{bmatrix} \in \partial(\boldsymbol{\pi}_k^m \circ \mathbf{F})(\tilde{\mathbf{x}}_0) \right\} \tag{6.22}$$

must contain only invertible matrices, which makes $\mathbf{h}$ Clarke regular at $\tilde{\mathbf{x}}_0$.

<u>All cases</u>: Thus we can apply the C^r/PC^r/Clarke Inverse Function Theorem to $\mathbf{h}$ at $\tilde{\mathbf{x}}_0$ and get neighborhoods $A \subset V_1$ of $\tilde{\mathbf{x}}_0$ and $B \subset \mathbb{R}^n$ of $\mathbf{h}(\tilde{\mathbf{x}}_0)$ such that $\mathbf{h} : A \to B$ is a $\mathcal{G}$ homeomorphism.

By the definition of $\mathbf{h}$, $\boldsymbol{\pi}_k^n \circ \mathbf{h} = \boldsymbol{\pi}_k^m \circ \mathbf{F}$ on A. Then $\boldsymbol{\pi}_k^m \circ \mathbf{F} \circ \mathbf{h}^{-1} = \boldsymbol{\pi}_k^n$ on B and the composite $\mathcal{G}$ function $\mathbf{F} \circ \mathbf{h}^{-1} : B \to \mathbb{R}^m$ is of the form

$$\mathbf{F} \circ \mathbf{h}^{-1}(\mathbf{x}) = \big(x_1, \ldots, x_k,\ \mathbf{G}(\mathbf{x})\big), \tag{6.23}$$

where $\mathbf{G} : B \to \mathbb{R}^{m-k}$ is a $\mathcal{G}$ function. $\mathbf{F} \circ \mathbf{h}^{-1}$ is differentiable at $\mathbf{x} \in B$ if and only if $\mathbf{G}$ is differentiable, in which case

$$\mathbf{J}(\mathbf{F} \circ \mathbf{h}^{-1})(\mathbf{x}) = \begin{bmatrix} \mathbf{I}_k & \mathbf{0}_{k\times(n-k)} \\ \mathbf{C}(\mathbf{x}) & \mathbf{D}(\mathbf{x}) \end{bmatrix}, \quad \mathbf{JG}(\mathbf{x}) = \begin{bmatrix} \mathbf{C}(\mathbf{x}) & \mathbf{D}(\mathbf{x}) \end{bmatrix}, \quad \mathbf{D}(\mathbf{x}) \in \mathbb{R}^{(m-k)\times(n-k)}. \tag{6.24}$$

<u>C^r case</u>: Since $\mathbf{h}^{-1} : B \to A$ is a C^r homeomorphism, by the (converse) C^1 Inverse Function Theorem we know $\mathbf{Jh}^{-1}(\mathbf{x})$ is invertible $\forall \mathbf{x} \in B$. In addition, since $\mathbf{h}^{-1}(B) = A \subset V_1$, then $\mathbf{J}(\mathbf{F} \circ \mathbf{h}^{-1})(\mathbf{x}) = \mathbf{JF}(\mathbf{h}^{-1}(\mathbf{x}))\ \mathbf{Jh}^{-1}(\mathbf{x})$ must have rank k for all $\mathbf{x} \in B$. By inspecting Equation 6.24, the only way this can happen is if $\mathbf{D}(\mathbf{x}) = \mathbf{0}$ and thus $\boldsymbol{\rho}_{n-k}\mathbf{G}(\mathbf{x}) = \{\mathbf{D}(\mathbf{x})\} = \{\mathbf{0}\}$ for all $\mathbf{x} \in B$.

<u>PC^r case</u>: In view of Equation 6.24, for every $\mathbf{x} \in B$ we have

$$\partial^B(\mathbf{F} \circ \mathbf{h}^{-1})(\mathbf{x}) = \left\{ \begin{bmatrix} \mathbf{I}_k & \mathbf{0}_{k\times(n-k)} \\ \mathbf{M} & \mathbf{N} \end{bmatrix} : \quad \begin{bmatrix} \mathbf{M} & \mathbf{N} \end{bmatrix} \in \partial^B\mathbf{G}(\mathbf{x}) \right\}. \tag{6.25}$$

Let $\mathbf{x} \in B$. Since $\mathbf{h}^{-1} : B \to A$ is a PC^r homeomorphism, by the converse of the PC^r Inverse Function Theorem every matrix in $\partial^B\mathbf{h}^{-1}(\mathbf{x})$ has full rank n. Since $\mathbf{h}^{-1}(B) = A \subset V_1$, all matrices in $\partial^B\mathbf{F}(\mathbf{h}^{-1}(\mathbf{x}))$ have rank k for all $\mathbf{x} \in B$. Then, according to Proposition 2.1.5, every matrix in

$$\partial^B(\mathbf{F} \circ \mathbf{h}^{-1})(\mathbf{x}) \subset \left\{\mathbf{M}_1\mathbf{M}_2 : \mathbf{M}_1 \in \partial^B\mathbf{F}(\mathbf{h}^{-1}(\mathbf{x})),\ \mathbf{M}_2 \in \partial^B\mathbf{h}^{-1}(\mathbf{x})\right\}, \tag{6.26}$$

must have rank k. By inspecting Equation 6.25, the only way this can happen is if $\boldsymbol{\rho}^B_{n-k}\mathbf{G}(\mathbf{x}) = \{\mathbf{0}\}$ and thus $\boldsymbol{\rho}_{n-k}\mathbf{G}(\mathbf{x}) = \{\mathbf{0}\}$ for every $\mathbf{x} \in B$.

<u>Lipschitz case</u>: $\mathbf{h}$ is differentiable if and only if $\boldsymbol{\pi}^m_k \circ \mathbf{F}$ is differentiable at $\mathbf{x} \in A \subset V_1$, and by Proposition 2.2.2 in [22] we have $[\mathbf{A}(\mathbf{x})\ \ \mathbf{B}(\mathbf{x})] = \mathbf{J}(\boldsymbol{\pi}^m_k \circ \mathbf{F})(\mathbf{x}) \in \partial(\boldsymbol{\pi}^m_k \circ \mathbf{F})(\mathbf{x})$. Therefore, in this case $\mathbf{A}(\mathbf{x})$ and thus $\mathbf{Jh}(\mathbf{x})$ must be invertible. Since $\mathbf{h} : A \to B$ is a Lipschitz bijection, $\mathbf{h}^{-1}$ is differentiable at $\mathbf{x} \in B$ if and only if $\mathbf{h}$ is differentiable at

$\mathbf{h}^{-1}(\mathbf{x}) \in A$, and in that case $\mathbf{Jh^{-1}}(\mathbf{x}) = \big(\mathbf{Jh}(\mathbf{h}^{-1}(\mathbf{x}))\big)^{-1}$ is also invertible.

Let $\Omega_{\mathbf{F}\circ\mathbf{h}^{-1}} \subset B$ and $\Omega_{\mathbf{h}^{-1}} \subset B$ be the measure zero sets where $\mathbf{F} \circ \mathbf{h}^{-1}$ and $\mathbf{h}^{-1}$ fail to be differentiable. Letting $\mathbf{x} \in B \setminus (\Omega_{\mathbf{F}\circ\mathbf{h}^{-1}} \cup \Omega_{\mathbf{h}^{-1}})$, we can see that $\mathbf{h}^{-1}$ and $\mathbf{F} \circ \mathbf{h}^{-1}$ are differentiable at $\mathbf{x}$, $\mathbf{h}$ is differentiable at $\mathbf{h}^{-1}(\mathbf{x}) \in A$, $\mathbf{Jh}^{-1}(\mathbf{x})$ is invertible, and by the Chain Rule $\mathbf{F} = (\mathbf{F} \circ \mathbf{h}^{-1}) \circ \mathbf{h}$ is differentiable at $\mathbf{h}^{-1}(\mathbf{x})$. From the constant rank assumption, $\mathbf{JF}(\mathbf{h}^{-1}(\mathbf{x})) \in \partial\mathbf{F}(\mathbf{h}^{-1}(\mathbf{x}))$ must have rank k since $\mathbf{h}^{-1}(\mathbf{x}) \in V_1$. Moreover, by the Chain Rule

$$\mathbf{J}(\mathbf{F} \circ \mathbf{h}^{-1})(\mathbf{x}) = \mathbf{JF}(\mathbf{h}^{-1}(\mathbf{x}))\ \mathbf{Jh}^{-1}(\mathbf{x}) = \begin{bmatrix} \mathbf{I}_k & \mathbf{0}_{k\times(n-k)} \\ \mathbf{C}(\mathbf{x}) & \mathbf{D}(\mathbf{x}) \end{bmatrix} \tag{6.27}$$

must have rank k, which is only possible if $\mathbf{D}(\mathbf{x}) = \mathbf{0}$ for every $\mathbf{x} \in B \setminus (\Omega_{\mathbf{F}\circ\mathbf{h}^{-1}} \cup \Omega_{\mathbf{h}^{-1}})$. Given that $\Omega_{\mathbf{F}\circ\mathbf{h}^{-1}} = \Omega_{\mathbf{G}}$ and in view of Equation 2.4, we conclude that $\boldsymbol{\rho}_{n-k}\mathbf{G}(\mathbf{x}) = \{\mathbf{0}\}$ for every $\mathbf{x} \in B$.

All cases: We can shrink $B \subset \mathbb{R}^n$ such that it is a convex neighborhood of $\tilde{\mathbf{x}}_0$ where $\mathbf{G}$ is globally Lipschitz, in which case we shrink $A = \mathbf{h}^{-1}(B)$ accordingly. Given $(\mathbf{u}, \mathbf{v}), (\mathbf{u}, \mathbf{v}') \in B \subset \mathbb{R}^k \times \mathbb{R}^{n-k}$, we can apply the Lipschitz Mean Value Theorem (Proposition 2.6.5 in [22]) to $\mathbf{G}$ and conclude there exist $(\mathbf{c}_1, \mathbf{c}_2) \in L$, $\mathbf{M} \in \boldsymbol{\pi}_k\mathbf{G}(\mathbf{c}_1, \mathbf{c}_2)$ and $\mathbf{N} \in \boldsymbol{\rho}_{n-k}\mathbf{G}(\mathbf{c}_1, \mathbf{c}_2)$ such that

$$\mathbf{G}(\mathbf{u}, \mathbf{v}') - \mathbf{G}(\mathbf{u}, \mathbf{v}) = \mathbf{M}(\mathbf{u} - \mathbf{u}) + \mathbf{N}(\mathbf{v}' - \mathbf{v}) = \mathbf{N}(\mathbf{v}' - \mathbf{v}), \tag{6.28}$$

where $L \subset B$ is the line segment between $(\mathbf{u}, \mathbf{v}), (\mathbf{u}, \mathbf{v}')$. Since $(\mathbf{c}_1, \mathbf{c}_2) \in B$ we must have $\mathbf{N} = \mathbf{0}$, thus $\mathbf{G}(\mathbf{u}, \mathbf{v}) = \mathbf{G}(\mathbf{u}, \mathbf{v}')$ and we conclude $\mathbf{G}$ is not a function of the last $n - k$ variables on B.

Let $U' = \boldsymbol{\pi}_k^n(B) \subset \mathbb{R}^k$, which is an open set, and pick any $\mathbf{v}' \in \boldsymbol{\rho}_{n-k}^n(B)$ to create the $\mathcal{G}$ function $\mathbf{z} : U' \to \mathbb{R}^{m-k}$, $\mathbf{z}(\mathbf{u}) = \mathbf{G}(\mathbf{u}, \mathbf{v}')$. Then $\mathbf{F} \circ \mathbf{h}^{-1}(\mathbf{u}, \mathbf{v}) = (\mathbf{u}, \mathbf{z}(\mathbf{u}))$ for every $(\mathbf{u}, \mathbf{v}) \in B$. Now let $W = U' \times \mathbb{R}^{m-k}$, which is a neighborhood of $\mathbf{F}(\tilde{\mathbf{x}}_0) = \mathbf{F} \circ \mathbf{h}^{-1}(\mathbf{h}(\tilde{\mathbf{x}}_0))$

in $\mathbb{R}^m$, and define the $\mathcal{G}$ function $\mathbf{q} : W \to \mathbb{R}^m$ as $\mathbf{q}(\mathbf{u}, \mathbf{v}) = (\mathbf{u}, \mathbf{v} - \mathbf{z}(\mathbf{u}))$. We can easily check that $\mathbf{q} : W \to W$ is a $\mathcal{G}$ homeomorphism with inverse $\mathbf{q}^{-1}(\mathbf{u}, \mathbf{v}) = (\mathbf{u}, \mathbf{v} + \mathbf{z}(\mathbf{u}))$. Moreover, for $(\mathbf{u}, \mathbf{v}) \in B$,

$$\mathbf{q} \circ \mathbf{F} \circ \mathbf{h}^{-1}(\mathbf{u}, \mathbf{v}) = \mathbf{q}(\mathbf{u}, \mathbf{z}(\mathbf{u})) = \big(\mathbf{u}, \mathbf{0}_{m-k}\big) = \boldsymbol{\iota}_m^k \circ \boldsymbol{\pi}_k^n(\mathbf{u}, \mathbf{v}). \tag{6.29}$$

Let $U = \mathbf{g}_1(A) \subset \mathbb{R}^n$, which is a neighborhood of $\mathbf{x}_0$, and define $\boldsymbol{\phi} = \mathbf{h} \circ \mathbf{g}_1{}^{-1} : U \to B$, which is a $\mathcal{G}$ homeomorphism between open subsets of $\mathbb{R}^n$ by composition; note $\boldsymbol{\phi}(U) = B$. Let $V = \mathbf{g}_2{}^{-1}(W) \subset \mathbb{R}^m$, which is a neighborhood of $\mathbf{f}(\mathbf{x}_0) = \mathbf{g}_2{}^{-1} \circ \mathbf{F}(\tilde{\mathbf{x}}_0)$, and let $\boldsymbol{\psi} = \mathbf{q} \circ \mathbf{g}_2 : V \to W$, which is a $\mathcal{G}$ homeomorphism by composition. Since $\mathbf{F} \circ \mathbf{h}^{-1}(B) \subset W$, then $\mathbf{f}(U) = \mathbf{f} \circ \mathbf{g}_1 \circ \mathbf{h}^{-1}(B) \subset \mathbf{g}_2{}^{-1}(W) = V$, and we conclude

$$\boldsymbol{\psi} \circ \mathbf{f} \circ \boldsymbol{\phi}^{-1} = \mathbf{q} \circ \mathbf{g}_2 \circ \mathbf{f} \circ \mathbf{g}_1 \circ \mathbf{h}^{-1} = \mathbf{q} \circ \mathbf{F} \circ \mathbf{h}^{-1} = \boldsymbol{\iota}_m^k \circ \boldsymbol{\pi}_k^n : \boldsymbol{\phi}(U) \to \mathbb{R}^m. \tag{6.30}$$

□

Proof. **(b)** Let $S = \mathbf{f}^{-1}(\mathbf{c})$, and note that $\boldsymbol{\psi}(\mathbf{c}) = \boldsymbol{\psi} \circ \mathbf{f} \circ \boldsymbol{\phi}^{-1}(\boldsymbol{\phi}(\mathbf{x}_0)) = \boldsymbol{\iota}_m^k \circ \boldsymbol{\pi}_k^n(\boldsymbol{\phi}(\mathbf{x}_0))$ must be of the form $\boldsymbol{\psi}(\mathbf{c}) = (\mathbf{d}, \mathbf{0})$, where $\mathbf{d} \in \mathbb{R}^k$. For $\mathbf{x} \in \boldsymbol{\phi}(U)$, we have that

$$(x_1, \ldots, x_k) = \mathbf{d} \Leftrightarrow \boldsymbol{\iota}_m^k \circ \boldsymbol{\pi}_k^n(\mathbf{x}) = \boldsymbol{\psi} \circ \mathbf{f} \circ \boldsymbol{\phi}^{-1}(\mathbf{x}) = \boldsymbol{\psi}(\mathbf{c}) \Leftrightarrow \mathbf{x} \in \boldsymbol{\phi}(S), \tag{6.31}$$

therefore $\boldsymbol{\phi}(U \cap S)$ is the following $(n-k)$ slice of $\boldsymbol{\phi}(U) \subset \mathbb{R}^n$:

$$\boldsymbol{\phi}(U \cap S) = \left\{(x_1, \ldots, x_n) \in \boldsymbol{\phi}(U) : (x_1, \ldots, x_k) = \mathbf{d}\right\}. \tag{6.32}$$

Since $U = \mathbf{g}_1(A)$, then $\mathbf{h}^{-1}(\boldsymbol{\phi}(U \cap S)) = \mathbf{g}_1{}^{-1}(U \cap S) \subset A$. From the definition of $\mathbf{h}$ in Equation 6.17, the $\mathcal{G}$ function $\mathbf{h}^{-1} : B \to A$ must be of the form $\mathbf{h}^{-1}(\mathbf{u}, \mathbf{v}) = (\mathbf{s}(\mathbf{u}, \mathbf{v}), \mathbf{v})$, where $\mathbf{s} : B \to \mathbb{R}^k$ is a $\mathcal{G}$ function. The set $V' = \boldsymbol{\rho}_{n-k}^n(\boldsymbol{\phi}(U \cap S)) \subset \mathbb{R}^{n-k}$ is open from Proposition 2.2.8, and $\mathbf{v} \in V'$ if and only if $(\mathbf{d}, \mathbf{v}) \in \boldsymbol{\phi}(U \cap S) \subset B$. Define the $\mathcal{G}$ function

$\mathbf{y} : V' \to \mathbb{R}^k$ such that $\mathbf{y}(\mathbf{v}) = \mathbf{s}(\mathbf{d}, \mathbf{v})$. Then

$$\mathbf{g}_1{}^{-1}(U \cap S) = \mathbf{h}^{-1}(\boldsymbol{\phi}(U \cap S)) = \{(\mathbf{u}, \mathbf{v}) \in \mathbb{R}^k \times V' : \mathbf{u} = \mathbf{y}(\mathbf{v})\}. \tag{6.33}$$

□

Proof. **(c)** Noting that $U' = \boldsymbol{\pi}^n_k(\boldsymbol{\phi}(U)) \subset \mathbb{R}^k$ is open, we have that $\mathbf{f}(U) \subset V$ and

$$\boldsymbol{\psi} \circ \mathbf{f}(U) = \boldsymbol{\psi} \circ \mathbf{f} \circ \boldsymbol{\phi}^{-1}(\boldsymbol{\phi}(U)) = \boldsymbol{\iota}^k_m \circ \boldsymbol{\pi}^n_k(\boldsymbol{\phi}(U)) = \big\{(\mathbf{u}, \mathbf{0}) \in \mathbb{R}^k \times \mathbb{R}^{m-k} :\ \mathbf{u} \in U'\big\}. \tag{6.34}$$

Then

$$\mathbf{g}_2 \circ \mathbf{f}(U) = \mathbf{q}^{-1} \circ \boldsymbol{\psi} \circ \mathbf{f}(U) = \big\{(\mathbf{u}, \mathbf{z}(\mathbf{u})) \in \mathbb{R}^k \times \mathbb{R}^{m-k} :\ \mathbf{u} \in U'\big\}. \tag{6.35}$$

Moreover, since $\boldsymbol{\psi}(\mathbf{f}(U)) = U' \times \{\mathbf{0}_{m-k}\}$ is a k-slice of $W = U' \times \mathbb{R}^{m-k}$ and $\boldsymbol{\psi}(\mathbf{f}(U)) \subset \boldsymbol{\psi}(V) = W$,

$$\boldsymbol{\psi}(\mathbf{f}(U)) = \{\mathbf{x} \in \boldsymbol{\psi}(V) :\ x_{k+1} = \cdots = x_m = 0\}\,. \tag{6.36}$$

□

Remark 6.4.3. For a C^r map of constant rank, $\mathbf{g}_1, \mathbf{g}_2$ can always be chosen as permutation linear homeomorphisms, therefore we can say that $U \cap \mathbf{f}^{-1}(\mathbf{c})$ and $\mathbf{f}(U)$ are directly graphs (within permutation) of C^r functions. In the Lipschitz and PC^r cases, in general we can only say that these two sets can be *transformed* into graphs by the homeomorphisms $\mathbf{g}_1{}^{-1}$ and $\mathbf{g}_2$, respectively.

Remark 6.4.4. For $\mathcal{G}$ submersions [immersions], one can prove separately a stronger version of Theorem 6.4.2 (a) where only a single $\mathcal{G}$ chart, $\mathbf{g}_1$ [$\mathbf{g}_2$], is needed. The results follow from Theorem 6.4.2(a) and Proposition 6.3.6.

Chapter 7

Lipschitz and Piecewise-differentiable manifolds

This chapter introduces piecewise-differentiable (PC^r) manifolds according to a unified general framework that also applies to nonsmooth Lipschitz manifolds and smooth manifolds. We present definitions of nonsmooth manifolds and embedded submanifolds for abstract sets as well as for subsets of $\mathbb{R}^n$, and explore the relationships between them. The PC^r and Lipschitz Rank Theorems from Chapter 6, in terms of the Clarke Jacobian and B-subdifferential generalized derivative sets, are used to characterize level sets of functions between nonsmooth manifolds as embedded submanifolds. We illustrate how the Level Set Theorems developed in this chapter can be applied to functions on Euclidean space, including the piecewise-differentiable process model for distillation columns presented in Chapter 3.

7.1. Introduction

Despite the prevalence of nonsmooth "manifold-like" objects in many applications, including the level sets of the nonsmooth MESH model from Chapter 3, there is a lack of precise topological notions to describe these sets in terms of their specific nonsmooth properties. Abstract Lipschitz manifolds have been previously defined in the literature [74, 58, 67] according to the standard differential topology framework presented in Definition 7.2.3. However, Lipschitz Rank Theorems (Theorem 6.4.2 in Chapter 6) have not yet been employed to define Lipschitz embeddings and embedded submanifolds (Definitions 7.4.1, 7.4.2) nor to develop a Level Set Theorem (Theorem 7.6.1) for Lipschitz functions. Using the PC^r Rank Theorem likewise developed in Chapter 6, in this chapter we also define PC^r embedded submanifolds and obtain Level Set Theorems. We follow a unified exposition approach where the symbol $\mathcal{G}$, representing a category on open sets of $\mathbb{R}^n$, can be replaced by C^r, PC^r and (locally) Lipschitz. As demonstrated in Theorem 7.4.3, our definitions of Lipschitz and PC^r embedded submanifolds are equivalent to geometric descriptions in terms of k-slices of open sets and of local nonsmooth homeomorphic transformations into graphs.

To the best of our knowledge, a well-defined concept for piecewise-differentiable manifolds has not been proposed in the literature. In the field of differential topology, the concept of "piecewise-differentiable" (PD) functions arises in the context of triangulations of smooth manifolds. In [89], Whitehead showed that every smoothly embedded submanifold $M \subset \mathbb{R}^n$ admits a so-called PD homeomorphism $h : K \to M$, where K is a simplicial complex, and the restriction of h to each simplex in K is a smooth map with full column rank (i.e., an immersion). This definition naturally extends to that of PD homeomorphisms between open subsets of $\mathbb{R}^n$, since the latter admit triangulations as well as a smooth structure. However, in general PD homeomorphisms cannot be composed or inverted and thus do not lead to the concept of a PD manifold.

In contrast, piecewise-differentiable (PC^r) functions as established in [78] by Scholtes (see Section 2.1.2) induce a well-defined category on open subsets of $\mathbb{R}^n$. Intuitively, the "pieces" where a PC^r function is C^r are not restricted to have any particular geometric structure, which ultimately allows these functions to be closed under composition. Taking that into account, we define PC^r manifolds in Section 7.2 using the same standard framework from differential topology. Since PC^r functions are locally Lipschitz continuous, PC^r manifolds are a subset of Lipschitz manifolds whose special properties should be taken advantage of in future research. One of the advantages of working with the former is that the B-subdifferential of a PC^r function is a finite set, which can be expressed in terms of the derivatives of the C^r pieces.

In addition to the most general “intrinsic” or abstract way to define manifolds, more concrete definitions for subsets of $\mathbb{R}^n$ are also traditionally employed. In this chapter we generalize the so-called “extrinsic” smooth manifold definition (e.g., see [60]), which is based on the concept of “extended” homeomorphisms, to PC^r and Lipschitz manifolds in $\mathbb{R}^n$. The definition of Lipschitz manifolds in $\mathbb{R}^n$ by Rockafellar [73], which describes sets that can be transformed into the graph of a Lipschitz function by a C^1 homeomorphism, is a special instance of an extrinsic Lipschitz manifold.

Theorems 7.5.2 and 7.5.3 show that every $\mathcal{G} \in \{C^r, PC^r, \text{Lipschitz}\}$ embedded submanifold of $\mathbb{R}^n$ is an extrinsic $\mathcal{G}$ manifold, while the converse holds for $\mathcal{G} = C^r$ but may or may not hold for $\mathcal{G} \in \{PC^r, \text{Lipschitz}\}$. In other words, given the equivalence in Theorem 7.4.3, the following remains an open research question: can every $\mathcal{G} \in \{PC^r, \text{Lipschitz}\}$ extrinsic manifold be locally transformed into the graph of a $\mathcal{G}$ function by a $\mathcal{G}$ homeomorphism? In case the answer turns out to be negative, the extrinsic $\mathcal{G}$ manifold definition would constitute the most general decription of topologically “well-behaved” nonsmooth sets in $\mathbb{R}^n$. In Example 2 of [62] the authors present a 1-dimensional extrinsic Lipschitz manifold in $\mathbb{R}^2$ which cannot be transformed into a Lipschitz graph by any linear orthogonal homeomorphism $\mathbf{g} : \mathbb{R}^2 \to \mathbb{R}^2$. However, there might be a Lipschitz homeomorphism that can achieve this transformation, in which case the set would be an embedded Lipschitz submanifold of $\mathbb{R}^2$ by Theorem 7.4.3.

7.2. Piecewise-differentiable (PC^r) manifolds

We refer the reader to Chapter 2 and to Section 6.2 of Chapter 6 for notation and definitions that will be used throughout the present chapter.

First we recall that a topological manifold is a Hausdorff and second countable topological space M such that every $p \in M$ has an associated chart (U, ϕ) around p, where $U \subset M$ is a neighborhood of p, $\phi : U \to \phi(U)$ is a homeomorphism, $\phi(U) \subset \mathbb{R}^n$ is open, and n is called the dimension of M at p. An atlas for M is a collection of charts $\mathcal{A} = \{(U_i, \phi_i)\}_{i \in I}$ such that $M = \bigcup_{i \in I} U_i$. We say M is n-dimensional if it has dimension n at all $p \in M$. Henceforth we will consider manifolds with a single overall dimension out of convenience, which is not restrictive since each connected subset of a topological manifold must have a unique dimension. A topological manifold with boundary is defined in the same way, except that in this case $\phi(U) \subset \mathbb{H}^n$ is an open subset of the n-dimensional upper half space, $\mathbb{H}^n = \{(x_1, \ldots, x_n) \in \mathbb{R}^n : x_n \geq 0\}$.

Piecewise-differentiable (PC^r) functions, as previously characterized in Section 2.1.2, are closed under composition and thus define the PC^r category on open subsets of $\mathbb{R}^n$. In order to provide a unified exposition, the symbol $\mathcal{G}$ representing a generic category may be replaced with C^r, PC^r, or Lipschitz unless otherwise stated. Note that $C^r \subset PC^r \subset$ Lipschitz.

An important concept is that of "extended" $\mathcal{G}$ homeomorphisms between arbitrary subsets of $\mathbb{R}^n$, which generalizes Definition 2.2.1 of $\mathcal{G}$ homeomorphisms. One can easily verify that extended $\mathcal{G}$ functions and homeomorphisms, as presented in Definition 7.2.1, are closed under composition. Moreover, an extended $\mathcal{G}$ homeomorphism between open subsets of $\mathbb{R}^n$ is a $\mathcal{G}$ homeomorphism in the standard sense.

Definition 7.2.1 (Extended $\mathcal{G}$ homeomorphisms). Let $\mathcal{G}$ represent a category of functions defined on open subsets of $\mathbb{R}^n$. A function $\mathbf{f} : U \to V$ between arbitrary sets $U \subset \mathbb{R}^n, V \subset \mathbb{R}^m$ is said to be an extended $\mathcal{G}$ function if for every $\mathbf{x} \in U$ there exists an open set $W \subset \mathbb{R}^n$ containing $\mathbf{x}$ and a $\mathcal{G}$ function $\mathbf{F} : W \to \mathbb{R}^m$ such that $\mathbf{f}|_{U\cap W} = \mathbf{F}|_{U\cap W}$. $\mathbf{f}$ is said to be an extended $\mathcal{G}$ homeomorphism if it is a homeomorphism and both $\mathbf{f} : U \to V$ and $\mathbf{f}^{-1} : V \to U$ are extended $\mathcal{G}$ functions.

Remark 7.2.2. Every Lipschitz homeomorphism between arbitrary sets is also an extended Lipschitz homeomorphism, given that Lipschitz functions always admit an extension to an open superset [36].

PC^r homeomorphisms between open subsets of $\mathbb{R}^n$ form a pseudogroup on $\mathbb{R}^n$ (see Definition 3.1.1 in [81]). For this reason we can use the following standard $\mathcal{G}$ manifold definition, which is traditionally applied to $\mathcal{G} = C^r$, Lipschitz, analytic, etc (see [54, 58, 81]), to propose the concept of $\mathcal{G} = PC^r$ manifolds.

Definition 7.2.3 (**$\mathcal{G}$ manifold**)**.** Let $\mathcal{G}$ be a category that generates a pseudogroup on $\mathbb{R}^n$. A $\mathcal{G}$ atlas for a topological manifold (with boundary) is an atlas $\mathcal{A}$ containing only $\mathcal{G}$-compatible charts; i.e., given $(U_i, \phi_i), (U_j, \phi_j) \in \mathcal{A}$ with $U_i \cap U_j \neq \varnothing$, the transition map $\phi_j \circ \phi_i^{-1} : \phi_i(U_i \cap U_j) \to \phi_j(U_i \cap U_j)$ is a $\mathcal{G}$ homeomorphism (potentially in the extended sense). We say $\mathcal{A}$ is a maximal $\mathcal{G}$ atlas if it is not contained in any other $\mathcal{G}$ atlas. A $\mathcal{G}$ manifold $(M, \mathcal{A})$ (with boundary) is a topological manifold (with boundary) M together with a maximal $\mathcal{G}$ atlas $\mathcal{A}$, called a $\mathcal{G}$ structure for M.

Given the common definition framework, $\mathcal{G}$ manifolds share several properties and definitions with smooth manifolds. For instance, we can define a $\mathcal{G}$ structure $\mathcal{A}|_N$ for any open subset $N \subset M$ by restricting the domains of the charts in $\mathcal{A}$ to N. The standard $\mathcal{G}$ structure for $\mathbb{R}^n$ is the (unique) maximal $\mathcal{G}$ atlas $\mathcal{A}^{\mathcal{G}}_{\mathbb{R}^n}$ containing the identity chart $(\mathbb{R}^n, id_{\mathbb{R}^n})$. Therefore, $(U, \boldsymbol{\phi}) \in \mathcal{A}^{\mathcal{G}}_{\mathbb{R}^n}$ if and only if $\boldsymbol{\phi} : U \to \boldsymbol{\phi}(U)$ is a $\mathcal{G}$ homeomorphism between open subsets of $\mathbb{R}^n$. According to Definition 6.4.1, we will also call $(U, \boldsymbol{\phi}) \in \mathcal{A}^{\mathcal{G}}_{\mathbb{R}^n}$ a $\mathcal{G}$ chart for $\mathbb{R}^n$. Finally, we present the following short proposition for completeness.

Proposition 7.2.4. Let $(M, \mathcal{A}_M)$ be an n-dimensional $\mathcal{G}$ manifold, $(U, \phi) \in \mathcal{A}_M$, and $(V, \mathbf{g}) \in \mathcal{A}^{\mathcal{G}}_{\mathbb{R}^n}$ with $\phi(U) \subset V$. Then $(U, \mathbf{g} \circ \phi) \in \mathcal{A}_M$.

Proof. Let $(W, \psi) \in \mathcal{A}_M$ be any chart with $U \cap W \neq \varnothing$. Since $\phi \circ \psi^{-1} : \psi(U \cap W) \to \phi(U \cap W)$ is a $\mathcal{G}$ homeomorphism, so is the composition $\mathbf{g} \circ \phi \circ \psi^{-1} : \psi(U \cap W) \to \mathbf{g} \circ \phi(U \cap W)$. □

7.3. Rank Theorems for functions between manifolds

In Section 6.3 we defined $\mathcal{G}$ submersions, immersions, and constant rank maps $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ in terms of the existence of $\mathcal{G}$ homeomorphisms $\mathbf{g}_1 : \mathbb{R}^n \to \mathbb{R}^n$ and/or $\mathbf{g}_2 : \mathbb{R}^m \to \mathbb{R}^m$, such that the generalized derivatives of the so-called chart representative $\mathbf{f} \circ \mathbf{g}_1$, $\mathbf{g}_2 \circ \mathbf{f}$ or $\mathbf{g}_2 \circ \mathbf{f} \circ \mathbf{g}_1$, respectively, satisfy the required rank conditions. Definition 7.3.1 establishes $\mathcal{G}$ functions between $\mathcal{G}$ manifolds, and extends the concepts of submersions, immersions and constant rank maps based on the corresponding Euclidean chart representatives. In particular, by applying Proposition 6.3.6 to the chart representatives we have that a $\mathcal{G}$ submersion [immersion] $f : M \to N$ at $p \in M$ is also a $\mathcal{G}$ map of constant rank m [n] around p, where n, m are the dimensions of M, N respectively.

Definition 7.3.1. Let $(M, \mathcal{A}_M)$, $(N, \mathcal{A}_N)$ be $\mathcal{G}$ manifolds. A map $f : M \to N$ is said to be a $\mathcal{G}$ function/ submersion/ immersion at $p \in M$ [map of constant rank k around $p \in M$] if there exist charts $(U, \phi) \in \mathcal{A}_M$ around p and $(V, \psi) \in \mathcal{A}_N$ around $f(p)$, with $f(U) \subset V$, such that the chart representative $\psi \circ f \circ \phi^{-1} : \phi(U) \to \psi(V)$ is a $\mathcal{G}$ function/ submersion/ immersion at $\phi(p)$ [map of constant rank k around $\phi(p)$] in the Euclidean sense.

Remark 7.3.2. In the $\mathcal{G} = C^r$ case, $f : M \to N$ is traditionally defined as a submersion/ immersion/ map of constant rank based on the rank of the differential $df_p : T_pM \to T_{f(p)}N$, which is a linear map between tangent spaces. This is equivalent to the above definition because any choice of charts $(U, \phi) \in \mathcal{A}_M$, $(V, \psi) \in \mathcal{A}_N$ provides bases for $T_pM, T_{f(p)}N$ such that the matrix of df_p corresponds to the Jacobian of $\psi \circ f \circ \phi^{-1}$ at $\phi(p)$ (see [54] p. 63). On the other hand, in the Lipschitz and PC^r cases we must necessarily create these definitions in terms of chart representatives. It is possible that the conditions might be satisfied with some but not all choices of charts.

The following Rank Theorem for functions between manifolds, which is a standard result in smooth manifold theory [e.g., [54]], is novel in the $\mathcal{G} = PC^r$ and Lipschitz cases. Slices of open subsets of $\mathbb{R}^n$, and the projection and inclusion $\boldsymbol{\iota}^k_m, \boldsymbol{\pi}^n_k$ functions were defined in Chapter 2.

Our nonsmooth Rank Theorems, both in their Euclidean (Theorem 6.4.2) and abstract manifold (Theorem 7.3.3) versions, provide conditions to conclude when a level set can be homeomorphically transformed into a graph around a given point. Theorem 7.4.3 will then show this is equivalent to the level set being an embedded submanifold of the domain manifold.

Theorem 7.3.3 (**Manifold $\mathcal{G}$ Rank Theorem**)**.** Let $(M, \mathcal{A}_M)$ and $(N, \mathcal{A}_N)$ be n-dimensional and m-dimensional $\mathcal{G}$ manifolds, $f : M \to N$ be a function, $p \in M$ and $f(p) = c$. Suppose that f is a $\mathcal{G}$ map of constant rank k around p, where $k \leq m, n$.

Then there exist charts $(U, \phi) \in \mathcal{A}_M$ around p and $(V, \psi) \in \mathcal{A}_N$ around $f(p)$, with $f(U) \subset V$, such that:

$$\begin{aligned} &\textit{(a)} \quad \psi \circ f \circ \phi^{-1} = \boldsymbol{\iota}^k_m \circ \boldsymbol{\pi}^n_k : \phi(U) \to \mathbb{R}^m, \\ &\qquad (x_1, \ldots, x_k, x_{k+1}, \ldots, x_n) \mapsto (x_1, \ldots, x_k, 0, \ldots, 0). \end{aligned} \tag{7.1}$$

(b) $\phi(U \cap f^{-1}(c))$ is an $(n-k)$-slice of $\phi(U) \subset \mathbb{R}^n$.

(c) $\psi(f(U))$ is a k-slice of $\psi(V) \subset \mathbb{R}^m$.

Proof. **(a)** By Definition 7.3.1 there exist charts $(U_1, \phi_1) \in \mathcal{A}_M$ around p and $(V_1, \psi_1) \in \mathcal{A}_N$ around $f(p)$ such that $\mathbf{F} = \psi_1 \circ f \circ (\phi_1)^{-1} : \phi_1(U_1) \to \mathbb{R}^m$ is a $\mathcal{G}$ map of constant rank around $\phi_1(p)$ in the Euclidean sense. Applying the Euclidean Rank Theorem 6.4.2(a) to $\mathbf{F}$ at $\phi_1(p)$, we can get charts $(U_2, \boldsymbol{\phi}_2) \in \mathcal{A}^{\mathcal{G}}_{\mathbb{R}^n}$ around $\phi_1(p)$ and $(V_2, \boldsymbol{\psi}_2) \in \mathcal{A}^{\mathcal{G}}_{\mathbb{R}^m}$ around $\psi_1(f(p))$, with $U_2 \subset \phi_1(U_1)$ and $V_2 \subset \psi_1(V_1)$, such that $\boldsymbol{\psi}_2 \circ \mathbf{F} \circ (\boldsymbol{\phi}_2)^{-1} = \boldsymbol{\iota}^k_m \circ \boldsymbol{\pi}^n_k$ on $\phi_2(U_2)$. Then let $U = (\phi_1)^{-1}(U_2)$, $\phi = \boldsymbol{\phi}_2 \circ \phi_1$, $V = (\psi_1)^{-1}(V_2)$, $\psi = \boldsymbol{\psi}_2 \circ \psi_1$, and shrink

U if necessary (replacing $U_2 = \phi_1(U)$ accordingly) to ensure $f(U) \subset V$, which is possible since f is continuous. We know $(U, \phi) \in \mathcal{A}_M$ is a chart around p and $(V, \psi) \in \mathcal{A}_N$ is a chart around $f(p)$ from Proposition 7.2.4, and

$$\psi \circ f \circ \phi^{-1} = \boldsymbol{\psi}_2 \circ \mathbf{f} \circ (\boldsymbol{\phi}_2)^{-1} = \boldsymbol{\iota}_m^k \circ \boldsymbol{\pi}_k^n \quad \text{on} \;\; \phi(U) = \boldsymbol{\phi}_2(U_2). \tag{7.2}$$

□

Proof. **(b)** Letting $\mathbf{c}' = \mathbf{F}(\phi_1(p)) = \psi_1(c)$, and given that ψ_1 is a homeomorphism and $U_2 = \phi_1(U)$,

$$\begin{aligned} U_2 \cap \mathbf{F}^{-1}(\mathbf{c}') &= \big\{\mathbf{x} \in U_2 : \mathbf{F}(\mathbf{x}) = \psi_1 \circ f \circ (\phi_1)^{-1}(\mathbf{x}) = \psi_1(c)\big\} = \big\{\mathbf{x} \in U_2 : f \circ (\phi_1)^{-1}(\mathbf{x}) = c\big\} \\ &= \{\phi_1(y) : y \in U, \;\; f(y) = c\} = \phi_1(U \cap f^{-1}(c)). \end{aligned} \tag{7.3}$$

Theorem 6.4.2(b) applied to $\mathbf{F}$ gives that $\boldsymbol{\phi}_2(U_2 \cap \mathbf{F}^{-1}(\mathbf{c}')) = \boldsymbol{\phi}_2 \circ \phi_1(U \cap f^{-1}(c)) = \phi(U \cap f^{-1}(c))$ is an $(n-k)$-slice of $\boldsymbol{\phi}_2(U_2) = \phi(U) \subset \mathbb{R}^n$. □

Proof. **(c)** Theorem 6.4.2(c) applied to $\mathbf{F}$ gives that

$$\boldsymbol{\psi}_2(\mathbf{F}(U_2)) = \boldsymbol{\psi}_2 \circ \psi_1 \circ f \circ (\phi_1)^{-1}(U_2) = \boldsymbol{\psi}_2 \circ \psi_1 \circ f(U) = \psi(f(U)) \tag{7.4}$$

is a k-slice of $\boldsymbol{\psi}_2(V_2) = \boldsymbol{\psi}_2 \circ \psi_1(V) = \psi(f(U)) \subset \mathbb{R}^m$. □

7.4. Embedded submanifolds

Our immersion, submersion and constant rank definitions were designed so that the same Rank Theorem results and the same definitions of embeddings and submanifolds from differential topology could be extended to Lipschitz and PC^r manifolds, as presented in Definitions 7.4.1 and 7.4.2 below. Theorem 7.4.3 establishes that sets which can be locally $\mathcal{G}$-homeomorphically transformed into the graph of a $\mathcal{G}$ function of k variables are equivalent to k-dimensional embedded $\mathcal{G}$ submanifolds. Moreover, it shows these sets can also be equivalently characterized in terms of local k-slices. We note that the key aspect in obtaining the equivalence between the k-slice and graph representations is the usage of a $\mathcal{G}$ homeomorphism to transform the manifold locally into a graph. In the case of Lipschitz manifolds in $\mathbb{R}^n$, this is in contrast to Definition 2.6 of [62] and to Rockafellar's Definition 7.5.4, which require the homeomorphism to be linear and orthogonal or C^1, respectively, instead of merely Lipschitz.

In the $\mathcal{G} =$ Lipschitz case the local slice characterization in Statement 1) of Theorem 7.4.3 corresponds to the Lipschitz submanifold Definition 2 from [67] and to Definition 2.3 in [62]. The standard smooth versions of Definitions 7.4.1, 7.4.2 and Theorem 7.4.3 can be found in standard textbooks such as [54].

Definition 7.4.1 (**$\mathcal{G}$ Embedding**)**.** Let $(M, \mathcal{A}_M)$, $(N, \mathcal{A}_N)$ be $\mathcal{G}$ manifolds. A map $f : M \to N$ is said to be a $\mathcal{G}$ embedding if it is a $\mathcal{G}$ immersion and a topological embedding, i.e., $f : M \to f(M)$ is a homeomorphism using the subspace topology for $f(M) \subset N$.

Definition 7.4.2 (**Embedded $\mathcal{G}$ submanifold**)**.** Let $(M, \mathcal{A})$ be a $\mathcal{G}$ manifold. We say $S \subset M$ is an embedded $\mathcal{G}$ submanifold of $(M, \mathcal{A})$ if it is a topological manifold with the subspace topology, and if it admits a $\mathcal{G}$ structure $\mathcal{A}_S$ such that the inclusion map $\iota : S \to M$ is a $\mathcal{G}$ embedding.

Theorem 7.4.3. Let $(M, \mathcal{A}_M)$ be an n-dimensional $\mathcal{G}$ manifold, $k \leq n$ and $S \subset M$. Then the following statements are equivalent:

1) For every $p \in S$ there exists a chart $(U, \phi) \in \mathcal{A}_M$ such that $p \in U$ and $\phi(U \cap S)$ is a k-slice of $\phi(U) \subset \mathbb{R}^n$;

2) For every $p \in S$ there exists a chart $(U', \phi') \in \mathcal{A}_M$ with $p \in U'$ and a $\mathcal{G}$ function $\mathbf{y} : V' \to \mathbb{R}^{n-k}$, $V' \subset \mathbb{R}^k$ open, such that $\phi'(U' \cap S)$ is the graph of $\mathbf{y}$;

3) $S \subset M$ is a k-dimensional embedded $\mathcal{G}$ submanifold of $(M, \mathcal{A}_M)$.

Proof. $(1 \Rightarrow 2, 3)$ Since M is a topological manifold, with the subset topology $S \subset M$ is automatically Hausdorff and second countable. Let $p \in S$ and $(U, \phi) \in \mathcal{A}_M$ be a chart according to 1). Let $\phi(U \cap S)$ be according to Equation 7.5 and let $V = U \cap S$, which is a neighborhood of p in S in the subspace topology. Since coordinate permutations are $\mathcal{G}$ homeomorphisms, without loss of generality we can assume

$$\phi(U \cap S) = \big\{\mathbf{x} \in \phi(U) \subset \mathbb{R}^n : (x_{k+1}, \dots, x_n) = \mathbf{c}\big\}, \quad \mathbf{c} \in \mathbb{R}^{n-k}. \tag{7.5}$$

The projection $V' = \boldsymbol{\pi}^n_k(\phi(V)) \subset \mathbb{R}^k$ is open according to Proposition 2.2.8. Then Statement 2) holds with $(U', \phi') = (U, \phi)$ for the constant (thus $\mathcal{G}$) function $\mathbf{y} : V' \to \mathbb{R}^{n-k}$, $\mathbf{y}(\mathbf{x}) = \mathbf{c}$.

Further, $\boldsymbol{\pi}^n_k|_{\phi(V)} : \phi(V) \to V'$ is an extended linear (thus $\mathcal{G}$) homeomorphism with inverse

$$\mathbf{i}^k_n|_{V'} : V' \to \phi(V), \qquad \mathbf{i}^k_n(u_1, \dots, u_k) = (u_1, \dots, u_k, \mathbf{c}). \tag{7.6}$$

The restriction $\phi|_V : V \to \phi(V)$ is a homeomorphism using the subspace topology for $V \subset S$ because $S \subset M$ has the subspace topology. By composition, $\psi = \boldsymbol{\pi}^n_k \circ \phi : V \to V'$ is a homeomorphism from a neighborhood $V \subset S$ of p onto an open subset $V' \subset \mathbb{R}^k$, therefore S is a k-dimensional topological manifold.

Let $\mathcal{A} = \big\{(V_p, \psi_p) : p \in S\big\}$ be the thus constructed atlas and $(V_\alpha, \psi_\alpha), (V_\beta, \psi_\beta) \in \mathcal{A}$ with $V_\alpha \cap V_\beta \neq \varnothing$. Then there exist charts $(U_i, \phi_i) \in \mathcal{A}_M$ such that $\psi_i = \boldsymbol{\pi}^n_k \circ \phi_i$ and $V_i = U_i \cap S$ for $i = \alpha, \beta$, and $U_\alpha \cap U_\beta \neq \varnothing$. From $\mathcal{G}$ compatibility of ϕ_α, ϕ_β and the fact that extended $\mathcal{G}$ homeomorphisms are closed under composition, the transition map

$$\psi_\alpha \circ {\psi_\beta}^{-1} = \boldsymbol{\pi}^n_k \circ \phi_\alpha \circ (\phi_\beta)^{-1} \circ (\boldsymbol{\pi}^n_k)^{-1} \tag{7.7}$$

is an extended $\mathcal{G}$ homeomorphism from $\psi_\beta(V_\alpha \cap V_\beta)$ onto $\psi_\alpha(V_\alpha \cap V_\beta)$. Since these are open sets in $\mathbb{R}^k$, $\psi_\alpha \circ {\psi_\beta}^{-1}$ is a $\mathcal{G}$ homeomorphism. Then $\mathcal{A}$ is a $\mathcal{G}$ atlas and we let $\mathcal{A}_S$ be its corresponding $\mathcal{G}$ maximal atlas.

Let $p \in S$ and take charts $(V, \psi) \in \mathcal{A}_S$, $(U, \phi) \in \mathcal{A}$ such that $p \in V = U \cap S$ and $\psi = \boldsymbol{\pi}^n_k \circ \phi : V \to V'$. The corresponding chart representative of the inclusion map $\iota : S \to M$ at p,

$$\phi \circ \iota \circ \psi^{-1} = \phi|_V \circ \psi^{-1} = (\boldsymbol{\pi}^n_k|_{\phi(V)})^{-1} = \mathbf{i}^k_n|_{V'} : V' \to \phi(V), \tag{7.8}$$

is a linear and injective function between Euclidean subsets, which makes $\iota : S \to M$ a $\mathcal{G}$ immersion for $\mathcal{G} = C^r, PC^r$, and Lipschitz. $\iota : S \to M$ is also clearly a topological embedding because $\iota(S) = S \subset M$ has the subspace topology, which makes the inclusion map a $\mathcal{G}$ embedding. □

Proof. (3 ⇒ 1) Let $p \in S$. Since $\iota : S \to M$ is a $\mathcal{G}$ immersion, it is a map of constant rank k, and Theorem 7.3.3(c) gives charts $(U, \phi) \in \mathcal{A}_S, (V, \psi) \in \mathcal{A}_M$ around $p = \iota(p)$ with $\iota(U) = U \subset V$ such that $\psi(\iota(U)) = \psi(U)$ is a k-slice of $\psi(V) \subset \mathbb{R}^n$. Since $U \subset S$ is open in the subset topology of $S \subset M$, there exists $W \subset M$ open such that $U = W \cap S$. Then $U = (W \cap V) \cap S$ and the slice condition 1) holds with the chart $(W \cap V, \psi|_{W \cap V}) \in \mathcal{A}_M$. □

Proof. (2 ⇒ 1) Starting from Statement 2) for any $p \in S$, we have that $\phi'(U' \cap S) = \{(\mathbf{u}, \mathbf{y}(\mathbf{u})) : \mathbf{u} \in V'\}$. Let $W = V' \times \mathbb{R}^{n-k}$ and define the $\mathcal{G}$ function $\mathbf{g}(\mathbf{u}, \mathbf{v}) = (\mathbf{u}, \mathbf{v} - \mathbf{y}(\mathbf{u}))$ on W. We can easily check that $\mathbf{g} : W \to W$ is a $\mathcal{G}$ homeomorphism with inverse $\mathbf{g}^{-1}(\mathbf{u}, \mathbf{v}) = (\mathbf{u}, \mathbf{v} + \mathbf{y}(\mathbf{u}))$. Next we let $U = (\phi')^{-1}(W \cap \phi'(U'))$ and define $\phi = \mathbf{g} \circ \phi'$ on U. We have that $U \cap S = U' \cap S$ and the set $\phi(U \cap S) = \mathbf{g} \circ \phi'(U' \cap S) = \{(\mathbf{u}, \mathbf{0}) : \mathbf{u} \in V'\}$ is a k-slice of W, therefore it is also a k-slice of the set $\phi(U) = \mathbf{g}(W \cap \phi'(U')) \subset W$. Finally, by Proposition 7.2.4 we have that $(U, \phi) \in \mathcal{A}_M$. □

7.5. Manifolds in $\mathbb{R}^n$

Now we consider a more concrete "extrinsic" definition of nonsmooth manifolds specifically tailored to subsets of $\mathbb{R}^n$, which is analogous to the smooth manifold definition used in [60]. We previously stated Definition 7.5.1 with $\mathcal{G} = PC^r$ in [19]. Definition 7.5.1 for an extrinsic Lipschitz manifold (with boundary) is equivalent to Definition 2.8 in [62] [Definition 3.1 in [58]].

Definition 7.5.1 (Extrinsic $\mathcal{G}$ manifold). An extrinsic $\mathcal{G}$ manifold (with boundary) is a set $M \subset \mathbb{R}^n$ endowed with the subspace topology such that for every point $\mathbf{x} \in M$ there exists a neighborhood $U \subset M$ of $\mathbf{x}$, an open subset $V \subset \mathbb{R}^k$ [$V \subset \mathbb{H}^k$] for some $k \leq n$ which is called the dimension of M at $\mathbf{x}$, and an extended $\mathcal{G}$ homeomorphism $\boldsymbol{\phi} : U \to V$.

The next two theorems establish the relationship between extrinsic $\mathcal{G}$ manifolds in $\mathbb{R}^n$ and embedded $\mathcal{G}$ submanifolds of $\mathbb{R}^n$. In the $\mathcal{G} = C^r$ case there is a 1-1 correspondence between the two objects. For $\mathcal{G} \in \{PC^r, \text{Lipschitz}\}$, however, it is not known if the same equivalence holds. That is, even though every extrinsic $\mathcal{G}$ manifold admits a $\mathcal{G}$ structure, it might not be possible to find a structure that makes it an embedded $\mathcal{G}$ submanifold of $\mathbb{R}^n$. In the $\mathcal{G} =$ Lipschitz case, this was presented as an open problem in [62]. Nevertheless, the "partial equivalence" demonstrated below is sufficient for our purposes in this chapter.

Theorem 7.5.2. Every extrinsic $\mathcal{G}$ manifold $M \subset \mathbb{R}^n$ admits a $\mathcal{G}$ structure $\mathcal{A}_M$. Moreover, if $\mathcal{G} = C^r$, then $\mathcal{A}_M$ can be chosen such that M is an embedded C^r submanifold of $\left(\mathbb{R}^n, \mathcal{A}_{\mathbb{R}^n}^{C^r}\right)$.

Proof. As a subset of Euclidean space, $M \subset \mathbb{R}^n$ is Hausdorff and second-countable. The extended $\mathcal{G}$ homeomorphisms $\{\boldsymbol{\phi}_i : U_i \to V_i\}_{i \in I}$ within Definition 7.5.1 cover M, therefore M is a topological manifold with atlas $\mathcal{A} = \{(U_i, \boldsymbol{\phi}_i)\}_{i \in I}$. Now let $(U_i, \boldsymbol{\phi}_i), (U_j, \boldsymbol{\phi}_j) \in \mathcal{A}$ with $U_i \cap U_j \neq \varnothing$. The transition map

$$\boldsymbol{\phi}_i \circ \boldsymbol{\phi}_j^{-1} : \boldsymbol{\phi}_j(U_i \cap U_j) \to \boldsymbol{\phi}_i(U_i \cap U_j), \tag{7.9}$$

is an extended $\mathcal{G}$ homeomorphism by composition, thus it is also a $\mathcal{G}$ homeomorphism since $\boldsymbol{\phi}_i(U_i \cap U_j), \boldsymbol{\phi}_j(U_i \cap U_j) \subset \mathbb{R}^k$ are open, for some $k \leq n$. Therefore $\mathcal{A}$ is a $\mathcal{G}$ atlas for M, and it is contained in a unique maximal $\mathcal{G}$ atlas $\mathcal{A}_M$.

Now suppose $\mathcal{G} = C^r$, and let $\mathbf{x} \in M$ and $\iota : M \to \mathbb{R}^n$ be the inclusion map. There exists a chart $(U, \boldsymbol{\phi}) \in \mathcal{A}_M$ with $\mathbf{x} \in U$; further, let $(V, \boldsymbol{\psi}) = (\mathbb{R}^n, id_{\mathbb{R}^n}) \in \mathcal{A}^{C^r}_{\mathbb{R}^n}$. Then the inclusion map has the chart representative $\boldsymbol{\psi} \circ \boldsymbol{\iota} \circ \boldsymbol{\phi}^{-1} = \boldsymbol{\phi}^{-1} : \boldsymbol{\phi}(U) \to \mathbb{R}^n$ at $\mathbf{x}$. Because the latter is injective and C^r on the open set $\boldsymbol{\phi}(U)$, its Jacobian is full column rank at $\mathbf{x}$. Therefore $\iota : M \to \mathbb{R}^n$ (as a map between manifolds) is a C^r immersion, and a topological embedding since $\boldsymbol{\iota}(M) = M \subset \mathbb{R}^n$ has the subspace topology. □

Theorem 7.5.3. Let $M \subset \mathbb{R}^n$ have the subspace topology. If M is a k-dimensional embedded $\mathcal{G}$ submanifold of $\left(\mathbb{R}^n, \mathcal{A}^{\mathcal{G}}_{\mathbb{R}^n}\right)$, then $M \subset \mathbb{R}^n$ is a k-dimensional extrinsic $\mathcal{G}$ manifold.

Proof. From Theorem 7.4.3, for every $\mathbf{x} \in M$ there exists a $\mathcal{G}$ chart $(V, \boldsymbol{\psi}) \in \mathcal{A}^{\mathcal{G}}_{\mathbb{R}^n}$ such that $\mathbf{x} \in V$ and $\boldsymbol{\psi}(V \cap M)$ is a k-slice of $\boldsymbol{\psi}(V) \subset \mathbb{R}^n$. The set $U = V \cap M$ is a neighborhood of $\mathbf{x}$ in M according to the subspace topology. Since coordinate permutations are $\mathcal{G}$ homeomorphisms, without loss of generality we can assume $U' = \boldsymbol{\pi}^n_k(\boldsymbol{\psi}(U)) \subset \mathbb{R}^k$ is open from Proposition 2.2.8. Given $\boldsymbol{\pi}^n_k : \boldsymbol{\psi}(U) \to U'$ is an extended linear (thus $\mathcal{G}$) homeomorphism, by composition $\boldsymbol{\phi} = \boldsymbol{\pi}^n_k \circ \boldsymbol{\psi}|_U : U \to U'$ is an extended $\mathcal{G}$ homeomorphism. □

Lastly, we present the definition of a Lipschitz manifold in $\mathbb{R}^n$ that was used by Rockafellar in [73]. As Proposition 7.5.5 shows together with Theorem 7.5.3, a Lipschitz manifold in the Rockafellar sense is a special case of an extrinsic Lipschitz manifold.

Definition 7.5.4. A k-dimensional Lipschitz manifold in the Rockafellar sense is a set $M \subset \mathbb{R}^n$ such that for every point $\mathbf{x}_0 \in M$ there exists a C^1 chart $(U, \boldsymbol{\phi})$ for $\mathbb{R}^n$ around $\mathbf{x}_0$ such that $\boldsymbol{\phi}(U \cap M) \subset \mathbb{R}^n$ is the graph of a Lipschitz function $\mathbf{y} : V \to \mathbb{R}^{n-k}$, $V \subset \mathbb{R}^k$ open.

Proposition 7.5.5. A k-dimensional Lipschitz manifold $M \subset \mathbb{R}^n$ in the Rockafellar sense is a k-dimensional Lipschitz embedded submanifold of $\left(\mathbb{R}^n, \mathcal{A}^{\text{Lip.}}_{\mathbb{R}^n}\right)$.

Proof. In Definition 7.5.4 the C^1 chart $(W, \boldsymbol{\phi})$ belongs to $\mathcal{A}^{\text{Lip.}}_{\mathbb{R}^n}$, therefore Statement 2) in Theorem 7.4.3 holds for $S = M$ and thus Statement 3) also holds. □

7.6. Level Set and Image Theorems

In this section we conclude our main goal of characterizing level sets of nonsmooth functions as PC^r and Lipschitz manifolds. Theorem 7.6.1 establishes level sets of functions between abstract $\mathcal{G}$ manifolds as embedded submanifolds of the domain manifold, while Theorem 7.6.2 allows us to characterize local images as embedded submanifolds of the target set (i.e., codomain) manifold. If we extend the smooth notion of a regular level set $\mathbf{f}^{-1}(\mathbf{c})$ to a set on which $\mathbf{f}$ is a $\mathcal{G}$ submersion, Theorem 7.6.1 gives the same standard result that regular level sets are embedded $\mathcal{G}$ submanifolds.

Theorem 7.6.1 ($\mathcal{G}$ **Level Set Theorem**). Let $(M, \mathcal{A}_M)$ and $(N, \mathcal{A}_N)$ be n-dimensional and m-dimensional $\mathcal{G}$ manifolds and $p \in M$ with $f(p) = c$. If $f : M \to N$ is a $\mathcal{G}$ map of constant rank $k \leq m, n$ around p [$\mathcal{G}$ submersion at p], then p has a neighborhood $U \subset M$ such that $U \cap f^{-1}(c)$ is an $(n-k)$-dimensional [$(n-m)$-dimensional] embedded $\mathcal{G}$ submanifold of $(M, \mathcal{A}_M)$.

Moreover, if f is a $\mathcal{G}$ map of constant rank $k \leq m, n$ around every $p \in f^{-1}(c)$ [$\mathcal{G}$ submersion on $f^{-1}(c)$], then $f^{-1}(c) \subset M$ is an $(n-k)$-dimensional [$(n-m)$-dimensional] embedded $\mathcal{G}$ submanifold of $(M, \mathcal{A}_M)$.

Proof. Since $\mathbf{f}$ is a $\mathcal{G}$ map of constant rank k [m] around p, by Theorem 7.3.3 (b) there exists a chart $(U, \boldsymbol{\phi}) \in \mathcal{A}_M$ around p such that $\boldsymbol{\phi}(U \cap f^{-1}(c))$ is an $(n-k)$-slice [$(n-m)$-slice] of $\boldsymbol{\phi}(U) \subset \mathbb{R}^n$. Then Statement 1) of Theorem 7.4.3 holds with $S = U \cap f^{-1}(c)$. If f has constant rank k [m] around every $p \in f^{-1}(c)$, then Statement 1) of Theorem 7.4.3 holds with $S = f^{-1}(c)$. □

Theorem 7.6.2 ($\mathcal{G}$ **Image Theorem**). Let $(M, \mathcal{A}_M)$ and $(N, \mathcal{A}_N)$ be n-dimensional and m-dimensional $\mathcal{G}$ manifolds. If $f : M \to N$ is a $\mathcal{G}$ map of constant rank $k \leq m, n$ around $p \in M$ [$\mathcal{G}$ immersion at $p \in M$], then p has a neighborhood $U \subset M$ such that $f(U)$ is a k-dimensional [n-dimensional] embedded $\mathcal{G}$ submanifold of $(N, \mathcal{A}_N)$.

Proof. From Theorem 7.3.3 (c) there exist charts $(U, \phi) \in \mathcal{A}_M$ around p and $(V, \psi) \in \mathcal{A}_N$ with $f(U) \subset V$ such that $\psi(f(U)) = \psi(V \cap f(U))$ is a k-slice of $\psi(V) \subset \mathbb{R}^m$. Then Statement 1) of Theorem 7.4.3 holds with $S = f(U) \subset N$. □

Remark 7.6.3. As in the smooth case, the above result does not imply that the image $f(U_0) \subset N$ of an open set $U_0 \subset M$ where f has constant rank k is a k-dimensional embedded $\mathcal{G}$ submanifold of the codomain (e.g., see Example 4.19 in [54]).

Now we specialize Theorem 7.6.1 to functions between Euclidean spaces, which is the case for practical applications. Theorem 6.4.2(b) allowed us to conclude that if $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ is a $\mathcal{G}$ map of constant rank k around $\mathbf{x}_0 \in \mathbb{R}^n$ with respect to homeomorphisms $\mathbf{g}_1 : \mathbb{R}^n \to \mathbb{R}^n$, $\mathbf{g}_2 : \mathbb{R}^m \to \mathbb{R}^m$ and $\mathbf{f}(\mathbf{x}_0) = \mathbf{c}$, then ${\mathbf{g}_1}^{-1}(\mathbf{f}^{-1}(\mathbf{c}))$ is locally the graph of a $\mathcal{G}$ function $\mathbf{y} : \mathbb{R}^{n-k} \to \mathbb{R}^k$ around $\mathbf{x}_0$. In other words, locally $\mathbf{f}^{-1}(\mathbf{c})$ can be $\mathcal{G}$-homeomorphically transformed into a $\mathcal{G}$ graph. As formalized in Theorem 7.6.4 below, now we can further say that $\mathbf{f}^{-1}(\mathbf{c}) \subset \mathbb{R}^n$ is an $(n-k)$-dimensional extrinsic $\mathcal{G}$ manifold around $\mathbf{x}_0$.

Theorem 7.6.4 (**Euclidean $\mathcal{G}$ Level Set Theorem**)**.** Let $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ be a $\mathcal{G}$ function, $\mathbf{x}_0 \in \mathbb{R}^n$ and $\mathbf{f}(\mathbf{x}_0) = \mathbf{c}$. If $\mathbf{f}$ is a $\mathcal{G}$ map of constant rank $k \leq m, n$ around $\mathbf{x}_0$ [$\mathcal{G}$ submersion at $\mathbf{x}_0$], then $\mathbf{x}_0$ has a neighborhood $U \subset \mathbb{R}^n$ such that $U \cap \mathbf{f}^{-1}(\mathbf{c}) \subset \mathbb{R}^n$ is an $(n-k)$-dimensional [$(n-m)$-dimensional] embedded $\mathcal{G}$ submanifold of $\left(\mathbb{R}^n, \mathcal{A}^{\mathcal{G}}_{\mathbb{R}^n}\right)$ and extrinsic $\mathcal{G}$ manifold.

Moreover, if $\mathbf{f}$ is a $\mathcal{G}$ map of constant rank $k \leq m, n$ around every $\mathbf{p} \in \mathbf{f}^{-1}(\mathbf{c})$ [$\mathcal{G}$ submersion on $\mathbf{f}^{-1}(\mathbf{c})$], then $\mathbf{f}^{-1}(\mathbf{c}) \subset \mathbb{R}^n$ is an $(n-k)$-dimensional [$(n-m)$-dimensional] embedded $\mathcal{G}$ submanifold of $\left(\mathbb{R}^n, \mathcal{A}^{\mathcal{G}}_{\mathbb{R}^n}\right)$ and extrinsic $\mathcal{G}$ manifold.

Proof. As a map from $\left(\mathbb{R}^n, \mathcal{A}^{\mathcal{G}}_{\mathbb{R}^n}\right)$ into $\left(\mathbb{R}^m, \mathcal{A}^{\mathcal{G}}_{\mathbb{R}^m}\right)$, $\mathbf{f} = \boldsymbol{\psi} \circ \mathbf{f} \circ \boldsymbol{\phi}^{-1}$acts as its own chart representative with $\boldsymbol{\psi} = id_{\mathbb{R}^m}$ and $\boldsymbol{\phi} = id_{\mathbb{R}^n}$. Then $\mathbf{f}$ is a map of constant rank k [m] in the manifold sense, and we can apply Theorems 7.6.1 and 7.5.3 to get the desired results. □

Similarly, an Euclidean version of the $\mathcal{G}$ Image Theorem 7.6.2 could also be written.

Finally, we will state a specialized Level Set Theorem for Lipschitz functions $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ based on the sufficient rank conditions SCRA and SFRA from Section 6.3.1. In [24], the authors show that if SFRA holds at $\mathbf{x}_0$ such that $\mathbf{f}(\mathbf{x}_0) = \mathbf{c}$, then $\mathbf{f}^{-1}(\mathbf{c})$ is locally an $(n - m)$-dimensional Lipschitz manifold around $\mathbf{x}_0$ in the Rockafellar sense. We now extend this result to the non-full rank SCRA case in Theorem 7.6.5.

Theorem 7.6.5 (Lipschitz Level Set Theorem). Let $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ be a Lipschitz function, $\mathbf{x}_0 \in \mathbb{R}^n$ and $\mathbf{f}(\mathbf{x}_0) = \mathbf{c}$. If the SCRA holds at $\mathbf{x}_0$ with rank $k \leq m, n$ [SFRA holds at $\mathbf{x}_0$], then $\mathbf{x}_0$ has a neighborhood $U \subset \mathbb{R}^n$ such that $U \cap \mathbf{f}^{-1}(\mathbf{c}) \subset \mathbb{R}^n$ is an $(n - k)$-dimensional [$(n - m)$-dimensional] Lipschitz manifold in the Rockafellar sense.

Proof. By Propositions 6.3.13 and 6.3.16, $\mathbf{f}$ is a Lipschitz map of constant rank k [m] around $\mathbf{x}_0$ with respect to orthogonal homeomorphisms $\mathbf{g}_1 : \mathbb{R}^n \to \mathbb{R}^n$, $\mathbf{g}_2 : \mathbb{R}^m \to \mathbb{R}^m$. By Theorem 6.4.2(b), $\mathbf{x}_0$ has a neighborhood $U \subset \mathbb{R}^n$ such that ${\mathbf{g}_1}^{-1}(U \cap \mathbf{f}^{-1}(\mathbf{c}))$ is the graph of a Lipschitz function $\mathbf{y} : V \to \mathbb{R}^{n-k}$, $V \subset \mathbb{R}^k$ open. Since $(\mathbb{R}^n, {\mathbf{g}_1}^{-1}) \in \mathcal{A}^{C^1}_{\mathbb{R}^n}$, by Definition 7.5.4 it follows that $U \cap \mathbf{f}^{-1}(\mathbf{c})$ is a k-dimensional [m-dimensional] Lipschitz manifold in the Rockafellar sense. □

7.7. Examples

In this section we illustrate how the Rank and Level Set Theorem results from both this chapter and Chapter 6 can be applied to functions on Euclidean space. First we present two prototypical examples, involving a PC^r submersion and a PC^r map of constant rank, for which we verify both Lipschitz and PC^r sufficient rank conditions that were presented in Section 6.3. Then we close with a higher-dimensional example taken from a case study in nonsmooth distillation column modeling, where both PC^r submersion and constant rank results are applied.

7.7.1. Example 1: PC^r submersion

Consider the function $\mathbf{f} : \mathbb{R}^3 \to \mathbb{R}^2$,

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} \min(x_1, x_2) \\ \min(-x_2 + x_3, x_3) \end{pmatrix}, \tag{7.10}$$

whose level set $\mathbf{f}^{-1}(\mathbf{0}) \subset \mathbb{R}^3$ is represented in Figure 7.1(a). This function was cited in [24, 43] as an example for which a simple permutation homeomorphism is not enough to transform its level sets into graphs. We have that $\mathbf{f}$ is piecewise-linear (thus also PC^∞), and the four linear functions

$$\mathbf{f}_{(1)}(\mathbf{x}) = \begin{pmatrix} x_1 \\ x_3 \end{pmatrix}, \quad \mathbf{f}_{(2)}(\mathbf{x}) = \begin{pmatrix} x_1 \\ -x_2 + x_3 \end{pmatrix}, \quad \mathbf{f}_{(3)}(\mathbf{x}) = \begin{pmatrix} x_2 \\ -x_2 + x_3 \end{pmatrix}, \quad \mathbf{f}_{(4)}(\mathbf{x}) = \begin{pmatrix} x_2 \\ x_3 \end{pmatrix} \tag{7.11}$$

form an essentially active selection set for $\mathbf{f}$ at $\mathbf{x}_0 = \mathbf{0}$. Therefore (see Section 2.1.2),

$$\partial\mathbf{f}(\mathbf{0}) = \operatorname{conv}\left\{ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right\} \tag{7.12}$$

$$= \left\{ \begin{bmatrix} \alpha_1 + \alpha_2 & \alpha_3 + \alpha_4 & 0 \\ 0 & -(\alpha_2 + \alpha_3) & 1 \end{bmatrix} : \alpha_i \geq 0, \sum_{i=1}^{4} \alpha_i = 1 \right\}. \tag{7.13}$$

Since $\alpha_1 + \alpha_2$ and $\alpha_3 + \alpha_4$ cannot both be equal to zero, every matrix in $\partial\mathbf{f}(\mathbf{0})$ has full row rank 2. However, no 2×2 submatrix remains invertible for all elements of $\partial\mathbf{f}(\mathbf{0})$, thus we cannot rearrange the coordinates x_1, x_2, x_3 to apply Clarke's Implicit Function Theorem. The FRA does hold at $\mathbf{x}_0 = \mathbf{0}$ (see Definition 6.3.8) but that might not necessarily imply that $\mathbf{f}$ is a Lipschitz submersion at $\mathbf{x}_0$ (see Remark 6.3.9). To show that the SFRA holds at the origin we must find a 2-dimensional subspace $E \subset \mathbb{R}^3$ such that

$$\mathcal{N}(\mathbf{A}) \cap E = \{\mathbf{0}\}, \quad \forall \mathbf{A} \in \partial \mathbf{f}(\mathbf{x}_0), \tag{7.14}$$

or equivalently (Remark 6.3.11)

$$\mathcal{R}(\mathbf{A}^{\mathrm{T}}) \cap E^{\perp} = \{\mathbf{0}\}, \quad \forall \mathbf{A} \in \partial \mathbf{f}(\mathbf{x}_0). \tag{7.15}$$

For a vector $\mathbf{v}_3 = (a, b, c)$ to be a basis of $E^{\perp}$, it cannot belong to the row space of any matrix in $\partial \mathbf{f}(\mathbf{0})$. That is, we must have

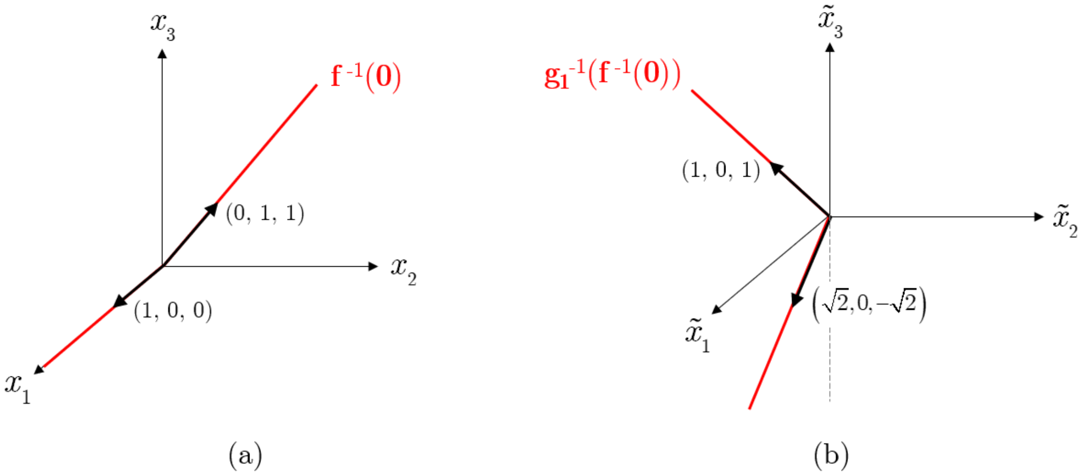
$$\det \begin{bmatrix} \alpha_1 + \alpha_2 & \alpha_3 + \alpha_4 & 0 \\ 0 & -(\alpha_2 + \alpha_3) & 1 \\ a & b & c \end{bmatrix} = a - (\alpha_1 + \alpha_3)\left[a + b + c(\alpha_2 + \alpha_3)\right] \neq 0, \quad \forall \alpha_i \geq 0 \text{ s.t. } \sum_{i=1}^{4} \alpha_i = 1, \tag{7.16}$$

which happens to be satisfied by the vector $\mathbf{v}_3 = \frac{1}{2}\left(-\sqrt{2}, 1, 1\right)$. Therefore, SFRA holds at $\mathbf{x}_0$ with $E = (\mathrm{span}(\mathbf{v}_3))^{\perp}$. Following the approach within the proof of Proposition 6.3.16, we create the orthogonal linear homeomorphism $\mathbf{g}_1(\mathbf{x}) = \mathbf{P}\mathbf{x}$ where the first 2 columns of $\mathbf{P}$ are an orthonormal basis of E and the third column, $\mathbf{v}_3$, is an orthonormal basis of $E^{\perp}$:

$$\mathbf{P} = \frac{1}{2} \begin{bmatrix} \sqrt{2} & 0 & -\sqrt{2} \\ 1 & \sqrt{2} & 1 \\ 1 & -\sqrt{2} & 1 \end{bmatrix} \tag{7.17}$$

The homeomorphism $\mathbf{g}_1$ is a change of basis from the standard unit vectors to the columns of the matrix $\mathbf{P}$. By Proposition 6.3.17 we can say $\mathbf{f}$ is a Lipschitz submersion at $\mathbf{x}_0$ with respect to $\mathbf{g}_1$, and since the latter is a linear homeomorphism, by Proposition 6.3.7 $\mathbf{f}$ is also a PC^{∞} submersion at $\mathbf{x}_0 = \mathbf{0}$ with respect to $\mathbf{g}_1$. By Proposition 6.3.6, $\mathbf{f}$ is a PC^{∞} map of constant rank 2 around $\mathbf{x}_0$ with respect to $\mathbf{g}_1, id_{\mathbb{R}^2}$.

Theorem 6.4.2(b) gives that $\mathbf{g_1}^{-1}(\mathbf{f}^{-1}(\mathbf{0})) = \mathbf{P}^{-1}\mathbf{f}^{-1}(\mathbf{0})$, represented in Figure 7.1(b), is locally the graph of a PC^∞ function $\mathbf{y} : \mathbb{R} \to \mathbb{R}^2$ around $\mathbf{x}_0$. In this case, this function happens to be $(\tilde{x}_1, \tilde{x}_2) = (|\tilde{x}_3|, 0) = \mathbf{y}(\tilde{x}_3)$. Theorem 7.6.5 shows that $\mathbf{f}^{-1}(\mathbf{0}) \subset \mathbb{R}^n$ is locally a 1-dimensional Lipschitz manifold in the Rockafellar sense around $\mathbf{x}_0$, which implies that this statement also holds in the extrinsic and embedded submanifold senses. Furthermore, we can apply Theorem 7.6.4 with $\mathcal{G} = PC^\infty$ to show that $\mathbf{f}^{-1}(\mathbf{0})$ is a PC^∞ 1-dimensional manifold around $\mathbf{x}_0$, both in the extrinsic and embedded submanifold senses.



x_3
$\mathbf{f}^{-1}(\mathbf{0})$
$(0, 1, 1)$
x_2
$(1, 0, 0)$
x_1
$\tilde{x}_3$
$\mathbf{g_1}^{-1}(\mathbf{f}^{-1}(\mathbf{0}))$
$(1, 0, 1)$
$\tilde{x}_2$
$(\sqrt{2}, 0, -\sqrt{2})$
$\tilde{x}_1$

(a) (b)

Figure 7.1: (a) Original level set $\mathbf{f}^{-1}(\mathbf{0})$, (b) Transformed level set $\mathbf{g_1}^{-1}(\mathbf{f}^{-1}(\mathbf{0}))$.

7.7.2. Example 2: PC^r map of constant rank

Now we add a third component to the function $\mathbf{f}$ from the previous example to create $\mathbf{h} : \mathbb{R}^3 \to \mathbb{R}^3$,

$$\mathbf{h}(\mathbf{x}) = \begin{pmatrix} \mathbf{f}(\mathbf{x}) \\ h_3(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \min(x_1, x_2) \\ \min(-x_2 + x_3, x_3) \\ \max(0, \min(x_1, x_2)) \end{pmatrix}. \tag{7.18}$$

We must have $\mathbf{h}^{-1}(\mathbf{0}) = \mathbf{f}^{-1}(\mathbf{0})$ because the equation $h_3(\mathbf{x}) = 0$ enforces either $0 = 0$ or $f_1(\mathbf{x}) = 0$. Now we will show how the PC^r Rank and Level Set Theorems can be applied to $\mathbf{h}$ to predict the behavior of its level set based on its generalized derivatives.

Since $h_1(\mathbf{x}), h_2(\mathbf{x})$ have two pieces and $h_3(\mathbf{x})$ has three pieces, one might be tempted to assume that $\mathbf{h}$ must have up to $2 \cdot 2 \cdot 3 = 12$ pieces. However, the PC^∞ function $\mathbf{h}$ has the following set of five essentially active selection functions at $\mathbf{x}_0 = \mathbf{0}$:

$$\mathbf{h}_{(1)}(\mathbf{x}) = \begin{pmatrix} x_1 \\ x_3 \\ 0 \end{pmatrix}, \quad \mathbf{h}_{(2)}(\mathbf{x}) = \begin{pmatrix} x_1 \\ -x_2 + x_3 \\ 0 \end{pmatrix}, \quad \mathbf{h}_{(3)}(\mathbf{x}) = \begin{pmatrix} x_2 \\ -x_2 + x_3 \\ x_2 \end{pmatrix},$$

$$\mathbf{h}_{(4)}(\mathbf{x}) = \begin{pmatrix} x_2 \\ x_3 \\ 0 \end{pmatrix}, \quad \mathbf{h}_{(5)}(\mathbf{x}) = \begin{pmatrix} x_1 \\ -x_2 + x_3 \\ x_1 \end{pmatrix}. \tag{7.19}$$

The regions where each piece $\mathbf{h}_{(i)}$ is active, which are independent of x_3, are represented in Figure 7.2 in x_2-x_1 space.

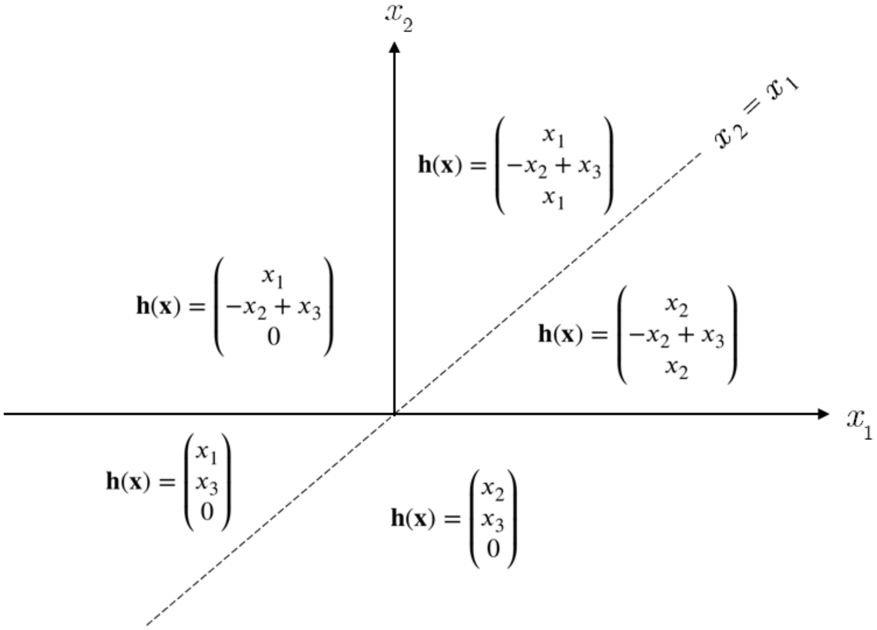

x_2
$x_2 = x_1$
$\mathbf{h}(\mathbf{x}) = \begin{pmatrix} x_1 \\ -x_2 + x_3 \\ x_1 \end{pmatrix}$
$\mathbf{h}(\mathbf{x}) = \begin{pmatrix} x_1 \\ -x_2 + x_3 \\ 0 \end{pmatrix}$
$\mathbf{h}(\mathbf{x}) = \begin{pmatrix} x_2 \\ -x_2 + x_3 \\ x_2 \end{pmatrix}$
x_1
$\mathbf{h}(\mathbf{x}) = \begin{pmatrix} x_1 \\ x_3 \\ 0 \end{pmatrix}$
$\mathbf{h}(\mathbf{x}) = \begin{pmatrix} x_2 \\ x_3 \\ 0 \end{pmatrix}$

Figure 7.2: The five essentially active selection functions for $\mathbf{h}$ at $\mathbf{x}_0 = \mathbf{0}$ in x_2-x_1 space.

We will transform $\mathbf{h}$ into $\mathbf{F} = \mathbf{h} \circ \mathbf{g}_1$ using the same orthogonal linear homeomorphism $\mathbf{g}_1(\mathbf{x}) = \mathbf{P}\mathbf{x}$ from the previous example, where $\mathbf{P}$ was given in Equation 7.17. The five functions $\mathbf{h}_{(i)}$ from Equation 7.19 are a valid set of selection functions for $\mathbf{h}$ at every $\mathbf{x} \in \mathbb{R}^3$, therefore $\partial^B \mathbf{h}(\mathbf{x}) \subset \partial^B \mathbf{h}(\mathbf{0})$ for every $\mathbf{x} \in \mathbb{R}^3$ and

$$\partial^B \mathbf{F}(\mathbf{x}) \subset \partial^B \mathbf{h}(\mathbf{0})\mathbf{P} = \left\{ \frac{1}{2}\begin{bmatrix} \sqrt{2} & 0 & -\sqrt{2} \\ 1 & -\sqrt{2} & 1 \\ 0 & 0 & 0 \end{bmatrix}, \frac{1}{2}\begin{bmatrix} \sqrt{2} & 0 & -\sqrt{2} \\ 0 & -2\sqrt{2} & 0 \\ 0 & 0 & 0 \end{bmatrix}, \frac{1}{2}\begin{bmatrix} 1 & \sqrt{2} & 1 \\ 0 & -2\sqrt{2} & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix}, \right.$$
$$\left. \frac{1}{2}\begin{bmatrix} 1 & \sqrt{2} & 1 \\ 1 & -\sqrt{2} & 1 \\ 0 & 0 & 0 \end{bmatrix}, \frac{1}{2}\begin{bmatrix} \sqrt{2} & 0 & -\sqrt{2} \\ 0 & -2\sqrt{2} & 0 \\ \sqrt{2} & 0 & -\sqrt{2} \end{bmatrix} \right\} \tag{7.20}$$

for every $\mathbf{x} \in \mathbb{R}^3$. Every matrix in $\partial^B \mathbf{F}(\mathbf{x})$ has rank 2 and all leading 2×2 submatrices in $\partial^B \mathbf{F}(\mathbf{x})$ have determinant < 0, for every $\mathbf{x} \in \mathbb{R}^3$. Moreover, the set of leading 2×2 submatrices in $\partial^B \mathbf{F}(\mathbf{x})$ already contains all possible row-by-row permutations of its elements (see 2.1.6). Therefore, $\pi_2^3 \circ \mathbf{F}$ is completely coherently oriented with respect to the first 2 variables at $\tilde{\mathbf{x}}_0 = \mathbf{0}$.

Considering Proposition 6.3.18 and Definition 6.3.2, we conclude $\mathbf{h} : \mathbb{R}^3 \to \mathbb{R}^3$ is a PC^∞ map of constant rank 2 around $\mathbf{x}_0$ w.r.t. $\mathbf{g}_1, id_{\mathbb{R}^3}$, which are orthogonal linear homeomorphisms. Theorem 7.6.4 then gives that $\mathbf{h}^{-1}(\mathbf{0}) \subset \mathbb{R}^3$ is a 1-dimensional PC^∞ manifold around $\mathbf{x}_0 = \mathbf{0}$ in the embedded submanifold and in the extrinsic senses. Note that the SCRA would have been considerably more cumbersome to attempt to verify in this simple example compared to the sufficient PC^r conditions demonstrated above.

Remark 7.7.1. The level set of a function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ might be an $(n-k)$-dimensional $\mathcal{G}$ manifold around a point $\mathbf{x}_0$ without $\mathbf{f}$ being a $\mathcal{G}$ map of constant rank k around $\mathbf{x}_0$. We can see this behavior even for simple C^r functions such as

$$\mathbf{f}(x_1, x_2) = \begin{pmatrix} x_2 - x_1 \\ x_2^2 - x_1^2 \end{pmatrix}. \tag{7.21}$$

The level set $\mathbf{f}^{-1}(\mathbf{0}) \subset \mathbb{R}^2$ is a 1-dimensional smooth manifold corresponding to the line $x_2 = x_1$. Nevertheless, $\mathbf{f}$ is not a map of constant rank 1 around any $\mathbf{x}_0 \in \mathbf{f}^{-1}(\mathbf{0})$ because

$$\mathbf{Jf}(x_1, x_2) = \begin{bmatrix} -1 & 1 \\ -2x_1 & 2x_2 \end{bmatrix} \tag{7.22}$$

has rank 2 everywhere except on $\mathbf{f}^{-1}(\mathbf{0})$ where its rank is 1.

7.7.3. Example 3: PC^r distillation model

Distillation is the most traditional and widely used industrial method for separating chemical mixtures. In [19] we developed a distillation model, with PC^∞ equations represented as

$$\mathbf{f}(\mathbf{x}, \lambda) = \mathbf{0}, \quad \mathbf{f} : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n, \tag{7.23}$$

that can generate solutions $\mathbf{x}$ with dry (or vaporless) stages for certain parameter values λ. The level set $\mathbf{f}^{-1}(\mathbf{0}) \subset \mathbb{R}^{n+1}$ is a connected 1-dimensional PC^∞ manifold, illustrated in Figure 7.3 for Case Study 1 from [19] where $n = 190$. The first plot shows how five liquid flow rates (a subset of the variables in $\mathbf{x} \in \mathbb{R}^n$) reach zero within the distillation column as functions of the reflux ratio λ. The second plot represents the corresponding behavior of $\mathbf{x} \in \mathbb{R}^n$ schematically. The restriction of $\mathbf{f}^{-1}(\mathbf{0})$ to the hyperplane $\lambda = \lambda_{\text{cr}}$, i.e., the level set $\mathbf{g}^{-1}(\mathbf{0}) \subset \mathbb{R}^n$ of $\mathbf{g}(\mathbf{x}) = \mathbf{f}(\mathbf{x}, \lambda_{\text{cr}})$, is a connected 1-dimensional PC^∞ manifold with two boundary points.

In our distillation model example, $\mathbf{f}$ is a C^∞ submersion at points $(\mathbf{x}, \lambda) \in \mathbf{f}^{-1}(\mathbf{0})$ where $\mathbf{f}$ is smooth, since $\mathbf{Jf}(\mathbf{x}, \lambda)$ has full row rank $n-1$. We know $\mathbf{f}$ is a PC^∞ submersion at the nonsmooth points $(\mathbf{x}, \lambda_{\text{cr}}) \in \mathbf{f}^{-1}(\mathbf{0})$ because we can verify that the sufficient conditions of Theorem 6.3.19(a) are satisfied. Similarly, $\mathbf{g}$ is a C^∞ map of constant rank $n-1$ around points $\mathbf{x} \in \mathbf{g}^{-1}(\mathbf{0})$ where $\mathbf{g}$ is smooth. Theorem 6.3.19(b) can be applied to show that $\mathbf{g}$ is a PC^∞ map of constant rank $n-1$ around the nonsmooth points $\mathbf{x} \in \mathbf{g}^{-1}(\mathbf{0})$ which

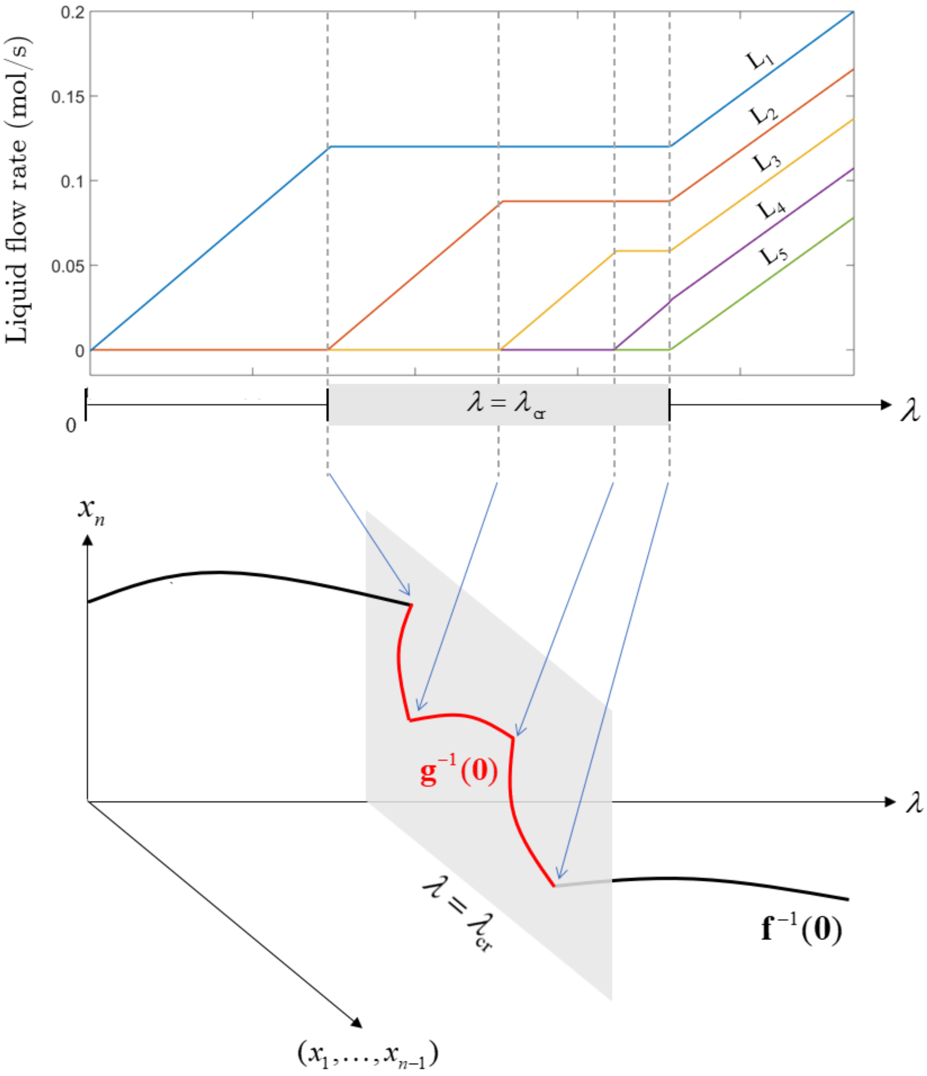


Figure 7.3: 1-dimensional PC^∞ manifold consisting of distillation model solutions from Case Study 1 in [19].

are not boundary points.

Therefore, the Level Set Theorem 7.6.4 can be used to predict the numerically observed behavior of the solutions to the nonsmooth distillation model. That is, $\mathbf{f}^{-1}(\mathbf{0}) \subset \mathbb{R}^{n+1}$ and the interior of $\mathbf{g}^{-1}(\mathbf{0}) \subset \mathbb{R}^n$ are locally 1-dimensional $\mathcal{G}$ submanifolds both in the embedded and extrinsic senses, where $\mathcal{G} = C^\infty$ around the smooth points and $\mathcal{G} = PC^\infty$ around the nonsmooth points. On the other hand, Theorem 6.3.19 would have to be extended in order to be applicable to the boundary points of $\mathbf{g}^{-1}(\mathbf{0})$.

Chapter 8

Conclusions and future lines of research

The work developed in this thesis encompasses both applied and theoretical components. Chapters 3, 4 and 5 presented nonsmooth models and equation-solving methods that can simulate distillation columns more accurately and more robustly than current state-of-the-art methods and software, including Aspen Plus. Chapters 6 and 7 presented a theoretical framework for characterizing the topology of nonsmooth sets, including level sets of nonsmooth functions, in terms of Lipschitz and piecewise-differentiable (PC^r) manifolds.

In Chapter 3 we developed a nonsmooth version of the MESH model to obtain steady states in which one or more distillation stages are dry/vaporless. This model, together with our equation-solving methods, allowed us to discover bifurcations consisting of a continuum of infinitely-many, multiple steady states occurring at a single set of parameter specifications. A natural extension would involve developing a dynamic version of said model, consisting of nonsmooth differential-algebraic equations, and performing dynamic distillation case studies to investigate if and how these degenerate steady states could be approached. Moreover, a dynamic model could allow us to perform a more rigorous stability analysis of these multiple steady states in terms of the model's generalized derivatives.

Another interesting ramification of this thesis would be attempting to validate the results of Chapter 3 experimentally. We showed that steady states with vaporless/dry stages are feasible from the viewpoint of the MESH paradigm, but bringing an experimental column to stabilize at such conditions could perhaps prove challenging or even impossible. In terms of the bifurcations, observing multiple steady states could be unlikely given their occurrence at a very specific set of specifications. However, we might be able to observe at least an apparent discontinuity in steady states around a certain set of operating conditions.

We also note that our steady-state nonsmooth MESH model and continuation strategies could be extended to analyze phase regime transitions in vapor-liquid-liquid systems. Previous computational and experimental results on heterogeneous azeotropic distillation [50, 90] seem to indicate a discontinuity in steady states associated with the appearance of a second liquid phase in several stages. This finding suggests the occurrence of degenerate bifurcations analogous to the ones we presented in Chapter 3.

In Chapter 4 we presented nonsmooth adaptive models consisting of the standard MESH equations with one or two nonsmooth specification equations. The single-soft adaptive model automatically resets one user-chosen "soft" specification if enforcing it would bring one or more flow rates outside of their physical bounds, which are also chosen by the user. In this case the model returns a "nearest best" MESH-feasible solution in which the impacted flow rates are brought to their minimum or maximum allowed values. As a result, the single-soft model does not exhibit MESH solutions with negative flow rates, nor the infinite discontinuities in flow rates with respect to purity specifications that we described in Chapter 4.

On the other hand, one must resort to our double-soft adaptive model when two specifications are infeasible simultaneously. Though inherently singular, we demonstrated that this model can be effectively solved using the pseudo-inverse of the model's B-subdifferential elements within the semismooth Newton method. However, as discussed in Section 2.3.5, there are currently no convergence theorems for this algorithm that could be applicable to singular generalized derivatives, or that consider the specific case of using B-subdifferential elements for a PC^r function. Therefore, developing such theorems would constitute a possible line of future research. Further, the modeling framework behind our single-soft and double-soft adaptive models is not inherently specific to distillation columns, and could thus be extended to other chemical engineering processes and to systems in other application fields.

In Chapter 5 we developed a nonsmooth version of the inside-out algorithm that can converge our adaptive models reliably starting from a "blind" initial guess, as demonstrated with several single-column test cases. Though our model outperformed Aspen Plus' RadFrac in its convergence reliability near infeasible specifications, our MATLAB implementation was not optimized for speed and numerical efficiency and therefore cannot compete with Aspen Plus in this aspect, which is coded in Fortran. Future efforts in this direction would involve implementing our simulation models in a compiled language such as C++, coding the Jacobian matrix of the inner loop analytically for its $N-2$ smooth equations, and using sparse matrix representation for the simultaneous algorithm to speed up the linear solve within the semismooth Newton step. The latter is not viable for the inner loop given that its generalized derivative matrices are dense. Additionally, the reliability of our inside-out algorithm could be further tested and improved upon by considering a wider range of non-ideal systems and column configurations.

In Chapter 5 we also developed a four-tier modeling strategy to simulate distillation columns within flowsheets with recycle, which combines the single and double-soft models with the simultaneous and inside-out algorithms to increase computational efficiency. With test cases involving two pressure-swing distillation flowsheets, we showed that our method can successfully converge to a feasible solution under highly infeasible specifications, which is not possible in Aspen Plus. Moreover, our method also outperformed Aspen Plus in feasible flowsheet test cases due to its ability to proceed through nearly or fully infeasible intermediate flowsheet iterations, and to converge without any initial guess for the tear stream (i.e., zero flow rate). In the future, it would be interesting to test and further develop our method to converge more complex flowsheets, e.g., including multistream heat exchangers modeled according to [85], more recycle streams, and design specifications. To enforce the latter, it might be necessary to use a Newton-type method to converge the flowsheet instead of a fixed-point method, as done in [88] for natural gas liquefaction processes.

The next research step in this path would involve employing our flowsheet simulation method to optimize highly-intensive and integrated processes involving distillation columns and multistream heat exchangers, e.g., an air separation unit for oxycombustion. If successful, this strategy would represent an important step towards fully rigorous optimization of complex processes, since eliminating or greatly reducing convergence errors in each flowsheet pass would allow optimization algorithms to explore a much wider range of process conditions. Though the corresponding process model equations are nonsmooth and our flowsheet pass function potentially discontinuous, one could at least begin to test this approach by using the smooth constrained optimization solver IPOPT and supplying generalized derivatives to the latter, as done in [84]. The output of the single-soft model flatlines beyond the two limiting values of the soft specified variable associated with the lower and upper bounds on flow rate values. Therefore, it remains to be determined if this type of behavior would allow for optimization algorithms to converge effectively.

In Chapter 6 we presented general constant rank conditions to develop PC^r and Lipschitz Rank Theorems, which allow us to characterize level sets of nonsmooth functions in Euclidean space as being homeomorphic to the graph of a nonsmooth function of the same class. We also developed sufficient PC^r rank conditions directly applicable to the residual function of the nonsmooth MESH model, which exhibits at most two essentially active selection functions at any given point. Future avenues of research could include proving sufficient rank conditions for more general subclasses of PC^r functions, which could be practically motivated by studying the level sets of other nonsmooth process models under parametric continuation.

In Chapter 7 we introduced the concept of PC^r manifolds, which can be used to characterize the topology of abstract sets, and utilized our Rank Theorems to define PC^r and Lipschitz embedded submanifolds. We also defined extrinsic nonsmooth manifolds for subsets of Euclidean space, and demonstrated how our constant rank conditions can be applied to conclude that the level set of a nonsmooth function is a correspondingly nonsmooth manifold. An important research question that could be explored is the potential equivalence between PC^r or Lipschitz extrinsic and embedded submanifolds, as discussed in Section 6.1. Further, the elementary theoretical framework for PC^r manifolds from Chapter 7 could be greatly expanded to include several other concepts and results. For example, it might be viable to define and establish the properties of a limiting "tangent" space for PC^r manifolds based on the properties of the B-derivative of PC^r functions.

Appendices

Appendix A

Izmailov's counterexample

Let

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \tag{A.1}$$

$$\mathbf{B}_1 = \begin{pmatrix} -0.2 & 0.8 & 0.2 \\ 0 & -0.5 & 0.4 \end{pmatrix}, \quad \mathbf{B}_2 = \begin{pmatrix} 0.7 & -0.7 & -0.6 \\ -0.8 & 0.1 & 0.6 \end{pmatrix}. \tag{A.2}$$

The set $\mathcal{A} = \operatorname{conv}\{\mathbf{M}, \mathbf{B}_1, \mathbf{B}_2\}$, which was presented and described in [41], is a counterexample to Proposition 2.1 in [17]; in turn, this also invalidates the proof of Theorem 3.1 in [17]. That is, $\mathcal{A}$ is a compact, convex set of $m \times n$ matrices of rank k, where $m = k = 2$ and $n = 3$, for which there exists no $(n-k)$-dimensional subspace $H \subset \mathbb{R}^n$ satisfying

$$\mathcal{R}(\mathbf{A}^{\mathrm{T}}) \cap H = \{\mathbf{0}\} \quad \forall \mathbf{A} \in \mathcal{A}. \tag{A.3}$$

The fact that the convex hull of finitely many matrices is a compact and convex set is a well-known fact.

First we demonstrate that all matrices in $\mathcal{A}$ have rank 2, following a different approach than the one presented in [41]. Note that

$$\mathcal{A} = \{x\mathbf{B}_1 + y\mathbf{B}_2 + z\mathbf{M} \ : \ x, y, z \geq 0, \ x + y + z = 1\}. \tag{A.4}$$

Substituing $z = 1 - x - y$ yields

$$\mathcal{A} = \left\{ \begin{pmatrix} 1 - 1.2x - 0.3y & 0.8x - 0.7y & 0.2x - 0.6y \\ -0.8y & 1 - 1.5x - 0.9y & 0.4x + 0.6y \end{pmatrix} \ : \ x, y \geq 0, x + y \leq 1 \right\}. \tag{A.5}$$

If we fix $y = 0$ then the corresponding matrices in $\mathcal{A}$ have the form

$$\begin{pmatrix} 1 - 1.2x & 0.8x & 0.2x \\ 0 & 1 - 1.5x & 0.4x \end{pmatrix}, \quad x \in [0, 1]. \tag{A.6}$$

There is no $x \in \mathbb{R}$ that makes either of the rows equal to the zero vector. Therefore, the only way such a matrix could have rank < 2 is if the rows are non-zero multiples of each other, and a necessary condition for that is $1 - 1.2x = 0$, that is, $x = 10/12 \in [0, 1]$. The only matrix which satisfies this condition has rank 2:

$$\begin{pmatrix} 0 & 8/12 & 2/12 \\ 0 & -3/12 & 4/12 \end{pmatrix}. \tag{A.7}$$

Therefore, we can restrict our analysis to the case $y \neq 0$. The only way that one of the corresponding matrices in $\mathcal{A}$ may have rank < 2 is if performing Gaussian elimination creates a row of zeroes. Using $-0.8y$ as the pivot to eliminate the first row of the matrices in Equation A.5, we can express this condition in terms of the following nonlinear system:

$$0.8y(0.8x - 0.7y) + (1 - 1.2x - 0.3y)(1 - 1.5x - 0.9y) = 0, \tag{A.8}$$

$$0.8y(0.2x - 0.6y) + (1 - 1.2x - 0.3y)(0.4x + 0.6y) = 0. \tag{A.9}$$

The solutions, which correspond to the intersections of the two conic sections expressed by the equations above, are shown in Figure A.1 and stated approximately below:

$$(x, y) = (0.833, 0),\ (-0.336, 0.957),\ (0.590, 0.507),\ (0.595, -0.206). \tag{A.10}$$

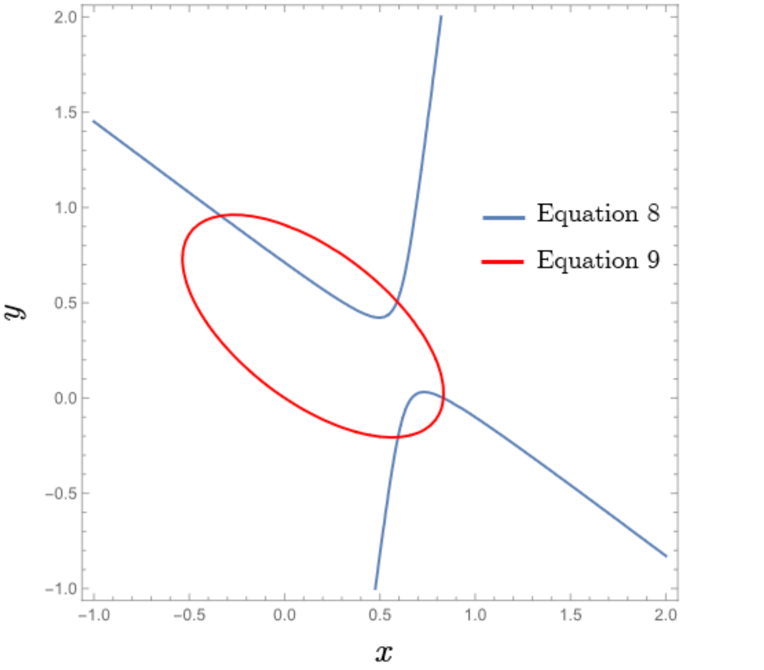


Figure A.1: Solutions of Equations A.8, A.9.

None of these solutions satisfy the conditions $x \geq 0,\ y > 0,\ x + y \leq 1$, therefore we conclude that all matrices in $\mathcal{A}$ have rank 2.

Next, we demonstrate that no 1-dimensional subspace $H \subset \mathbb{R}^3$ satisfies Equation A.3, following the approach from [41]. Let

$$\mathcal{B}_j = \{t\mathbf{B}_j + (1-t)\mathbf{M} \ : \ t \in [0, 1]\}, \quad j = 1, 2, \tag{A.11}$$

and

$$V_j = \left\{\mathbf{v} \in \mathbb{R}^3 \setminus \mathbf{0} \ : \ \mathcal{R}(\mathbf{B}^{\mathrm{T}}) \cap \operatorname{span}(\mathbf{v}) = \{\mathbf{0}\} \quad \forall \mathbf{B} \in \mathcal{B}_j\right\}, \quad j = 1, 2. \tag{A.12}$$

In other words, the spans of all elements of V_j correspond to all the 1-dimensional subspaces $H \subset \mathbb{R}^3$ that would satisfy Equation A.3 for the set $\mathcal{B}_j$ individually. Given that $\mathcal{B}_1, \mathcal{B}_2 \subset \mathcal{A}$, it suffices to show $V_1 \cap V_2 = \varnothing$ to demonstrate our desired result.

For $j = 1, 2$ the set V_j consists of all $\mathbf{v} \in \mathbb{R}^3 \setminus \mathbf{0}$ such that

$$\det \begin{pmatrix} t\mathbf{B}_j + (1-t)\mathbf{M} \\ \mathbf{v}^{\mathrm{T}} \end{pmatrix} \neq 0, \quad \forall t \in [0,1]. \tag{A.13}$$

Expressing Equation A.13 for $j = 1$ yields

$$(0.62v_1 + 0.48v_2 + 1.8v_3)\,t^2 + (-0.2v_1 - 0.4v_2 - 2.7v_3)\,t + v_3 \neq 0, \quad \forall t \in [0,1]. \tag{A.14}$$

and for $j = 2$ we have

$$(-0.96v_1 + 0.66v_2 - 0.29v_3)\,t^2 + (0.6v_1 - 0.6v_2 - 1.2v_3)\,t + v_3 \neq 0, \quad \forall t \in [0,1]. \tag{A.15}$$

If we choose $v_1 = 0$ then for any $v_2, v_3 \in \mathbb{R}$ we have that $t = 10/12 \in [0,1]$ is a root of Equation A.14, since in this case

$$\det \begin{pmatrix} t\mathbf{B}_1 + (1-t)\mathbf{M} \\ (0, v_2, v_3) \end{pmatrix} = v_2(0.48t^2 - 0.4t) + v_3(1.8t^2 - 2.7t) = v_2(0) + v_3(0) = 0. \tag{A.16}$$

Therefore, any vector of the form $(0, v_2, v_3)$ does not belong to V_1 and thus neither to $V_1 \cap V_2$. Then we can limit our analysis to the case $v_1 \neq 0$ and it suffices to show that $V_1' \cap V_2' = \varnothing$, where

$$V_j' = \left\{ (v_2, v_3) \in \mathbb{R}^2 \ : \ (1, v_2, v_3) \in V_j \right\}, \quad j = 1, 2. \tag{A.17}$$

The sets

$$V_1' = \big\{(v_2, v_3) \in \mathbb{R}^2 : (0.62 + 0.48v_2 + 1.8v_3)\, t^2 + (-0.2 - 0.4v_2 - 2.7v_3)\, t + v_3 \neq 0, \\ \forall t \in [0,1]\big\}, \tag{A.18}$$

$$V_2' = \big\{(v_2, v_3) \in \mathbb{R}^2 : (-0.96 + 0.66v_2 - 0.29v_3)\, t^2 + (0.6 - 0.6v_2 - 1.2v_3)\, t + v_3 \neq 0, \\ \forall t \in [0,1]\big\} \tag{A.19}$$

were plotted using Mathematica by expressing all possible scenarios involving the two quadratic roots of each equation (i.e., both roots > 1, both roots < 0, one root > 1 and the other < 0, or no real roots). The result is shown in Figure A.2, where we can see that $V_1' \cap V_2' = \varnothing$.

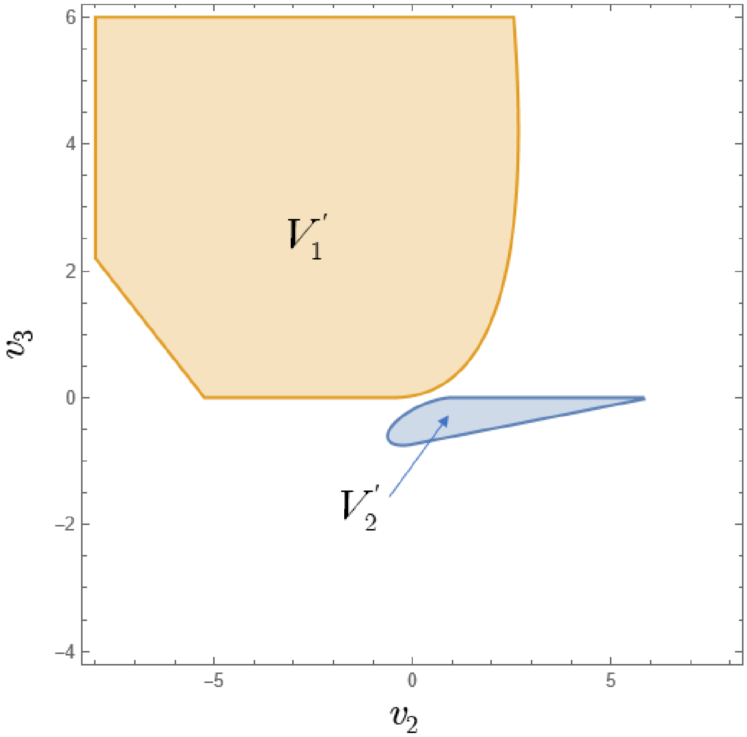


Figure A.2: The sets V_1', V_2'.

Bibliography

[1] Materials for separation technologies: Energy and emission reduction opportunities. Technical report, Oak Ridge, TN, 2005.

[2] Aspen Plus v10. Technical report, Bedford, MA, 2017.

[3] D. G. Anderson. Iterative procedures for nonlinear integral equations. *Journal of the ACM*, 12(4):547–560, 1965.

[4] AspenTech. *Aspen Plus steady-state simulation User Guide, Version 10, Volume 2.*

[5] A. Auslender. Theorem of constant rank for Lipschitzian maps. In A. V. Fiacco, editor, *Mathematical Programming with Data Perturbations II*. CRC Press, 2nd edition.

[6] L. Auslender and R. E. MacKenzie. *Introduction to Differentiable Manifolds.* McGraw-Hill, Hightstown, New Jersey, 1963.

[7] D. Bartl and M. Fabian. Every compact convex subset of matrices is the Clarke Jacobian of some Lipschitzian mapping. *Proceedings of the American Mathematical Society*, 149(11):4771–4779, 2021.

[8] P. I. Barton, K. A. Khan, P. Stechlinski, and H. A. Watson. Computationally relevant generalized derivatives: theory, evaluation and applications. *Optimization Methods and Software*, 33(4-6):1030–1072, 2018.

[9] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear programming: theory and algorithms.* John wiley & sons, 3rd edition, 2006.

[10] N. Bekiaris, G. A. Meski, C. M. Radu, and M. Morari. Multiple steady states in homogeneous azeotropic distillation. *Ind. Eng. Chem. Res.*, 32:2023–2038, 1993.

[11] N. Bekiaris, T. E. Guttinger, and M. Morari. Multiple steady states in distillation: Effect of vl(l)e inaccuracies. *AIChE J.*, 46:955–979, 2000.

[12] A. Ben-Israel and T. N. Greville. *Generalized inverses: theory and applications*, volume 15. Springer Science & Business Media, 2003.

[13] J. Boston and H. Britt. A radically different formulation and solution of the single-stage flash problem. *Computers & Chemical Engineering*, 2(2-3):109–122, 1978.

[14] J. Boston and S. Sullivan Jr. A new class of solution methods for multicomponent, multistage separation processes. *The Canadian Journal of Chemical Engineering*, 52 (1):52–63, 1974.

[15] J. F. Boston. Inside-out algorithms for multicomponent separation process calculations. In *Computer Applications to Chemical Engineering*, pages 135–151. ACS Publications, 1980.

[16] L. G. Bullard and L. T. Biegler. Iterated linear programming strategies for non-smooth simulation: a penalty based method for vapor-liquid equilibrium applications. *Comput. Chem. Eng.*, 17:95–109, 1993.

[17] G. J. Butler, J. G. Timourian, and C. Viger. The rank theorem for locally Lipschitz continuous functions. *Canadian Mathematical Bulletin*, 31(2):217–226, 1988.

[18] A. Carranza-Abaid and R. Gonzalez-Garcia. A Petlyuk distillation column dynamic analysis: Hysteresis and bifurcations. *Chem. Eng. Process. Process Intensif.*, 149: 107843, 2020.

[19] S. M. Cavalcanti and P. I. Barton. Multiple steady states and nonsmooth bifurcations in dry and vaporless distillation columns. *Industrial & Engineering Chemistry Research*, 59(40):18000–18018, 2020.

[20] X. Chen and T. Yamamoto. Newton-like methods for solving underdetermined nonlinear equations with nondifferentiable terms. *Journal of Computational and Applied Mathematics*, 55(3):311–324, 1994.

[21] X. Chen, Z. Nashed, and L. Qi. Convergence of Newton's method for singular smooth and nonsmooth equations using adaptive outer inverses. *SIAM Journal on Optimization*, 7(2):445–462, 1997.

[22] F. H. Clarke. *Optimization and nonsmooth analysis*. SIAM, 1990.

[23] D. Dorsch, H. T. Jongen, and V. Shikhman. On structure and computation of generalized Nash equilibria. *SIAM Journal on Optimization*, 23(1):452–474, 2013.

[24] D. Dorsch, H. T. Jongen, J.-J. Rückmann, and V. Shikhman. On the local representation of piecewise smooth equations as a Lipschitz manifold. *Journal of Mathematical Analysis and Applications*, 411(2):916–930, 2014.

[25] A. W. Dowling and L. T. Biegler. A framework for efficient large scale equation-oriented flowsheet optimization. *Comput. Chem. Eng.*, 72:3–20, 2015.

[26] A. W. Dowling, C. Balwani, Q. Gao, and L. T. Biegler. Equation-oriented optimization of cryogenic systems for coal oxycombustion power generation. *Energy Procedia*, 63:421–430, 2014.

[27] F. Facchinei and J.-S. Pang. *Finite-Dimensional Variational Inequalities and Complementarity Problems*, volume 2. Springer-Verlag New York, Inc., New York, NY, 2003.

[28] F. Facchinei, A. Fischer, and M. Herrich. An LP-Newton method: nonsmooth equations, KKT systems, and nonisolated solutions. *Mathematical Programming*, 146: 1–36, 2014.

[29] H.-r. Fang and Y. Saad. Two classes of multisecant methods for nonlinear acceleration. *Numerical linear algebra with applications*, 16(3):197–221, 2009.

[30] A. Fischer, M. Herrich, A. F. Izmailov, and M. V. Solodov. A globally convergent LP-Newton method. *SIAM Journal on Optimization*, 26(4):2012–2033, 2016.

[31] V. Gopal and L. T. Biegler. Smoothing methods for complementarity problems in process engineering. *AIChE J.*, 45:1535–1547, 1999.

[32] A. Griewank and A. Walther. *Evaluating derivatives: principles and techniques of algorithmic differentiation.* SIAM, 2008.

[33] I. E. Grossmann and F. Trespalacios. Systematic modeling of discrete-continuous optimization models through generalized disjunctive programming. *AIChE J.*, 59: 3276–3295, 2013.

[34] C. H. Gutzler. An iterative method of wegstein for solving simultaneous nonlinear equations, 1958.

[35] M. Herrich. *Local conver'gence of Newton-type methods for nonsmooth constrained equations and applications.* PhD thesis, Technische Universität Dresden, 2014.

[36] J.-B. Hiriart-Urruty. Extension of Lipschitz functions. *Journal of Mathematical Analysis and Applications*, 77(2):539–554, 1980.

[37] C. D. Holland and K. S. McMahon. Comparison of vaporization efficiencies with Murphree-type efficiencies in distillation - I. *Chem. Eng. Sci.*, 25:431–436, 1970.

[38] J. L. Humphrey and G. E. Keller II. *Separation Process Technology.* McGraw-Hill, New York, NY, 1997.

[39] I. C. F. Ipsen and C. D. Meyer. The angle between complementary subspaces. *The American Mathematical Monthly*, 102(10):904–911, 1995.

[40] A. Iqbal, S. A. Ahmad, et al. Strategies for separating pressure sensitive binary azeotropes. *Journal of King Saud University-Engineering Sciences*, 34(2):88–97, 2022.

[41] A. F. Izmailov. On a problem of existence of a nondegeneracy subspace for a convex compact family of epimorphisms. In V. A. Bereznev, editor, *Theoretical and Applied Problems of Nonlinear Analysis*, pages 34–49. Computer Center RAS, 2010. In Russian.

[42] E. W. Jacobsen and S. Skogestad. Multiple steady states in ideal two-product distillation. *AIChE J.*, 37:499–511, 1991.

[43] H. T. Jongen, J.-J. Rückmann, and V. Shikhman. On stability of the feasible set of a mathematical problem with complementarity problems. *SIAM Journal on Optimization*, 20(3):1171–1184, 2010. doi: 10.1137/08072694X.

[44] H. B. Keller. *Global Homotopies and Newton Methods*, pages 73–94. Academic Press, New York, NY, 1978.

[45] K. A. Khan and P. I. Barton. Generalized derivatives for solutions of parametric ordinary differential equations with non-differentiable right-hand sides. *Journal of Optimization Theory and Applications*, 163:355–386, 2014.

[46] K. A. Khan and P. I. Barton. A vector forward mode of automatic differentiation for generalized derivative evaluation. *Optim. Methods Softw.*, 30:1185–1212, 2015.

[47] A. Kienle, M. Groebel, and E. D. Gilles. Multiple steady states in binary distillation - theoretical and experimental results. *Chem. Eng. Sci.*, 50:2691–2703, 1995.

[48] M. Kojima and S. Shindo. Extension of Newton and quasi-Newton methods to systems of PC^1 equations. *Journal of the Operations Research Society of Japan*, 29 (4):352–375, 1986.

[49] J. W. Kovach III and W. D. Seider. Heterogeneous azeotropic distillation: Homotopy-continuation methods. *Comput. Chem. Eng.*, 11:593–605, 1987.

[50] J. W. Kovach III and W. D. Seider. Heterogeneous azeotropic distillation: Experimental and simulation results. *AIChE J.*, 33:1300–1314, 1987.

[51] R. Krishnamurthy and R. Taylor. A nonequilibrium stage model of multicomponent separation processes. Part I: Model description and method of solution. *AIChE J.*, 31:449–456, 1985.

[52] B. Kummer. Lipschitzian inverse functions, directional derivatives and application in C^{1,1} optimization. *Journal of Optimization Theory and Applications*, 70:559–580, 1991.

[53] Y.-D. Lang and L. T. Biegler. Distributed stream method for tray optimization. *AIChE J.*, 48:582–595, 2002.

[54] J. M. Lee. *Introduction to Smooth Manifolds.* Springer-Verlag, New York, NY, first edition, 2003.

[55] M. Lee, C. Dorn, G. A. Meski, and M. Morari. Limit cycles in homogeneous azeotropic distillation. *Ind. Eng. Chem. Res.*, 38:2021–2027, 1999.

[56] Y. Levin and A. Ben-Israel. A Newton method for systems of m equations in n variables. *Nonlinear Analysis-Theory Methods and Applications*, 47(3):1961–1972, 2001.

[57] Y. Levin and A. Ben-Israel. A Newton method for systems of m equations in n

variables. *Nonlinear Analysis-Theory Methods and Applications*, 47(3):1961–1972, 2001.

[58] J. Luukkainen and J. Väisälä. Elements of Lipschitz topology. *Ann. Acad. Sci. Fenn. Ser.*, 3(1):85–122, 1977. doi: 10.5186/aasfm.1977.0315.

[59] T. Magnussen, M. L. Michelsen, and A. Fredenslund. Azeotropic distillation using UNIFAC. *Inst. Chem. Eng. Symp. Ser.*, 56:4.2/1–4.2/19, 1979.

[60] J. Milnor and D. W. Weaver. *Topology from the differentiable viewpoint*. The University Press of Virginia, Charlottesville, 1965.

[61] D. Muller and W. Marquardt. Experimental verification of multiple steady states in heterogeneous azeotropic distillation. *Ind. Eng. Chem. Res.*, 36:5410–5418, 1997.

[62] J. Naumann and C. G. Simader. *Measure and integration on Lipschitz-manifolds*. Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II, Institut für Mathematik, 2011.

[63] R. D. Neidinger. Introduction to automatic differentiation and matlab object-oriented programming. *SIAM review*, 52(3):545–563, 2010.

[64] Y. Nesterov. Lexicographic differentiation of nonsmooth functions. *Mathematical programming*, 104:669–700, 2005.

[65] J. M. Ortega and W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*. SIAM, 2000.

[66] J.-S. Pang and D. Ralph. Piecewise smoothness, local invertibility, and parametric analysis of normal maps. *Mathematics of operations research*, 21(2):401–426, 1996.

[67] J.-P. Penot. What is a Lipschitzian manifold? *Set-Valued and Variational Analysis*, 30(3):1031–1040, 2022.

[68] R. Penrose. On best approximate solutions of linear matrix equations. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 52, pages 17–19. Cambridge University Press, 1956.

[69] L. Qi and J. Sun. A nonsmooth version of Newton's method. *Math. Program.*, 58: 353–367, 1993.

[70] A. U. Raghunathan and L. T. Biegler. Mathematical programs with equilibrium constraints (MPECs) in process engineering. *Comput. Chem. Eng.*, 27:1381–1392, 2003.

[71] A. U. Raghunathan, M. S. Diaz, and L. T. Biegler. An MPEC formulation for dynamic optimization of distillation operations. *Comput. Chem. Eng.*, 28:2037–2052, 2004.

[72] D. Ralph and S. Scholtes. Sensitivity analysis of composite piecewise smooth equations. *Mathematical Programming*, 76(3):593–612, 1997.

[73] R. T. Rockafellar. Maximal monotone relations and the second derivatives of nonsmooth functions. In *Annales de l'Institut Henri Poincaré C, Analyse non linéaire*, volume 2, pages 167–184. Elsevier, 1985.

[74] J. Rosenberg. Applications of analysis on Lipschitz manifolds. *Proc. Miniconferences on Harmonic Analysis and Operator Algebras (Canberra, t987), Proc. Centre for Math. Analysis*, 16:269–283, 1988.

[75] R. Russell. A flexible and reliable method solves single-tower and crude-distillation-column problems. *Chem. Eng.*, 90:53–59, 1983.

[76] A. M. Sahlodin, H. A. J. Watson, and P. I. Barton. Nonsmooth model for dynamic simulation of phase changes. *AIChE J.*, 62:3334–3351, 2016.

[77] G. Scheffknecht, L. Al-Makhadmeh, U. Schnell, and J. Maier. Oxy-fuel coal combustion—a review of the current state-of-the-art. *International Journal of Greenhouse Gas Control*, 5:S16–S35, 2011.

[78] S. Scholtes. *Introduction to Piecewise Differentiable Equations.* Springer, New York, NY, 2012.

[79] J. D. Seader, E. J. Henley, and D. K. Roper. *Separation Process Principles.* John Wiley & Sons, Inc., Hoboken, NJ, 3 edition, 2011.

[80] R. Taylor, R. Krishna, and H. Koojiman. Real-world modeling of distillation. *Chem. Eng. Prog.*, 99:28–39, 2003.

[81] W. P. Thurston. Three-dimensional geometry and topology, volume 1. In *Three-Dimensional Geometry and Topology, Volume 1.* Princeton university press, 2014.

[82] S. Venkataraman, W. K. Chan, J. Boston, et al. Reactive distillation using ASPEN PLUS. *Chemical Engineering Progress*, 86(8):45–54, 1990.

[83] J. Warga. Fat homeomorphisms and unbounded derivate containers. *Journal of Mathematical Analysis and Applications*, 81(2):545–560, 1981.

[84] H. A. Watson, M. Vikse, T. Gundersen, and P. I. Barton. Optimization of single mixed-refrigerant natural gas liquefaction processes described by nondifferentiable models. *Energy*, 150:860–876, 2018.

[85] H. A. J. Watson and P. I. Barton. Modeling phase changes in multistream heat exchangers. *Int. J. Heat Mass. Tran.*, 105:207–219, 2017.

[86] H. A. J. Watson, K. A. Khan, and P. I. Barton. Multistream heat exchanger modeling and design. *AIChE J.*, 61:3390–3403, 2015.

[87] H. A. J. Watson, M. Vikse, T. Gundersen, and P. I. Barton. Reliable flash calculations: Part 1. Nonsmooth inside-out algorithms. *Ind. Eng. Chem. Res.*, 56:960–973, 2017.

[88] H. A. J. Watson, M. Vikse, T. Gundersen, and P. I. Barton. Reliable flash calculations: Part 2. Process flowsheeting with nonsmooth models and generalized derivatives. *Ind. Eng. Chem. Res.*, 56:14848–14864, 2017.

[89] J. H. C. Whitehead. On c^1-complexes. *Annals of Mathematics*, pages 809–824, 1940.

[90] S. Widagdo, W. D. Seider, and D. H. Sebastian. Bifurcation analysis in heterogeneous azeotropic distillation. *AIChE J.*, 35:1457–1464, 1989.

[91] J. Zhang, B. O'Donoghue, and S. Boyd. Globally convergent type-I Anderson acceleration for nonsmooth fixed-point iterations. *SIAM Journal on Optimization*, 30(4): 3170–3197, 2020.

[92] Q. Zhang, M. Liu, C. Li, and A. Zeng. Heat-integrated pressure-swing distillation process for separation of the maximum-boiling azeotrope diethylamine and methanol. *Journal of the Taiwan Institute of Chemical Engineers*, 93:644–659, 2018.

[93] D. Zhu, J. P. Eason, and L. T. Biegler. Energy-efficient CO_2 liquefaction for oxy-combustion power plant with ASU-CPU integration enhanced by cascaded sub-ambient energy utilization and waste heat recovery. *Int. J. Greenh. Gas. Con.*, 61: 124–137, 2017.