CARNEGIE MELLON UNIVERSITY

Solution of Nonsmooth and Inequality Constrained Problems in Process Simulation and Optimization

A Dissertation

Submitted to the Graduate School

in Partial Fulfillment of the Requirements

for the degree of

Doctor of Philosophy

in

Chemical Engineering

by

Vipin Gopal

Pittsburgh, Pennsylvania

April 23, 1997

Abstract

Process simulation and optimization have emerged as valuable tools for plant design, analysis and operation. This thesis targets solution of a relevant class of nonsmooth and inequality constrained problems arising in chemical process simulation and optimization.

The first part of this thesis develops an improved framework for the solution of constrained simulation problems. Conversion of the equation solving problem to an optimization problem and solution of the resulting nonlinear program using Successive Linear Programming (SLP), provides a natural framework for dealing with variable bounds, conditionals, ill-conditioning and singularities, amongst others. Our work proposes a new descent strategy which combines the two major classes of strategies for ensuring descent: line search and trust region methods. Numerical results on a variety of test problems show that the combined line search - trust region method performs at least as well as the better of pure trust region or line search methods.

Further, the thesis addresses dynamic simulation problems, particularly issues related to nonsmoothness, and their solution techniques based on SLP methods. Dynamic simulation and optimization problems in chemical process engineering are most often modeled by Differential Algebraic Equations (DAEs). Determination of consistent initial conditions for a system of DAEs is a vital aspect of its solution. An SLP formulation with the DAE derivative array equations as the constraints in the optimization problem has been derived and tested. This approach handles roundoff errors and inconsistent specifications allowing reliable convergence strategies and is an attractive alternative to expensive SVD based least square methods. A theoretical contribution to the problem of reinitialization after discontinuities of systems modeled by DAEs, is also made in this thesis. A mathematical criterion is derived to identify the subset of variables which are continuous across a discontinuity caused by a step change in the input variables, for problems of arbitrary index. This has relevance in numerous applications and shows that the currently used assumption of the continuity of the differential variables is not always valid. A new algorithm for the detection and location of events which cause a discrete change in the modeling equations, has also been presented. The test problems solved include complex flow networks with checkvalves exhibiting hysteresis behavior. Phase equilibrium problems were considered as an important and relevant practical application of implicit nonsmooth systems. Traditional equation-based systems require clumsy swapping of equation sets, each of which is valid at particular phase combinations, when there is a change in the number of phases present. Our approach introduces a single unified optimization formulation which automatically solves the relevant set of equations and thus avoids the swapping of equation sets. The approach has been applied to nonideal multiphase systems involving liquid-liquid-vapor equilibrium and calculations in the critical region. Application of smoothing methods to directly handle the complementarity conditions arising from the optimality conditions of the above problems, particularly phase equilibrium, is also investigated. Simple representations based on smoothing techniques, of complex nested discontinuities occurring in process engineering have also been developed.

Incorporation of efficient primal-dual interior point methods, a variation of the Mehrotra predictor-corrector method in particular, within SLP and successive quadratic programming (SQP) for the solution of large structured process optimization problems was also studied. This is demonstrated to overcome the combinatorial complexity associated with active set methods, on problems drawn from optimal control, leading to significant computational savings. We also present a novel interpretation of Mehrotra predictor-corrector methods using the the principles of reduced Hessian SQP (rSQP).

This thesis is dedicated to
my beloved parents,
Premalatha and Gopal,
for their love, support
and encouragement.

Their belief in the value of
learning and higher education
has always inspired me.

Acknowledgements

As the old saying goes, one way to become good at something is to learn from the very best. I am incredibly privileged at that; my sincere thanks and appreciation to the very best, my advisor and mentor, Larry Biegler. His contagious enthusiasm for creativity and quality research, and his personality bubbling with a million ideas and endless energy have been my motivation and constant source of encouragement from day one of research at CMU. Under his tutelage I have learned to think with clarity, approach work with integrity, and above all, set high standards for myself. I will always be grateful for the significant impact he has had on my intellectual development. Larry and his wife, Lynne, are also acknowledged for being wonderful hosts during my stay at Pittsburgh.

I would like to express my appreciation to the other members of my committee, Ignacio Grossmann, Spyros Pandis and Gerald Thompson, for their inspiration, insight and words of encouragement during this research. Ignacio is also thanked for the intellectually stimulating discussions and for introducing me to the wonders of optimization in his classes. Thanks to Spyros for playing a key role in my professional development, providing thoughtful advice about my research and other aspects of professional endeavor. A special thanks goes to Gerald Thompson, for teaching me the first course in Linear Programming.

Thanks to Arthur Westerberg and Erik Ydstie for their seminal contributions to this field, which has always been an inspiration. I would also like to acknowledge the valuable input from Prof. Steve Campbell, Department of Mathematics at North Carolina State University, which helped shape up the work on DAEs. Funding from the Engineering Design Research Center is also greatly acknowledged.

Without a doubt, CMU Chemical Engineering Department has provided an excellent atmosphere to work in. I am grateful to the faculty, students and other department personnel that I have interacted with over the years. In particular, I would like to acknowledge:
Adam Turk, for all the help I have taken and for keeping me up with current affairs;
Purt Tanartkit, José Pinto and Lino Santos, for the countless hours of discussions on topics ranging from nonlinear programming to the state of affairs in Bosnia;
Arturo *King* Cervantes and Hector *Compadre* Yeomans, for hosting *las grandes* parties;
Ramesh *Dada* Iyer, for daily blessings from the God-like and Yi-Dong Lang, for being so jovial;
Tarun Bhatia, Scott Probst and George Staus, for being wonderful officemates;
Arzu and Metin Türkay, for sharing the *baklawas*;
Terumi Nagase, Jack Doney and Christine Hrenya, for always willing to offer help with anything;

Ramesh Raman, Dimitrios Varvarezos, Ignacio Quesada, Iauw-Bhieng Tjoa and Claudia Schmid, for playing the role of *elders* to perfection;
Dave Ternet, Ben Allan and Vicente Rico-Ramirez, for occassionally hopping over from EDRC;
Aaron Goldstein and Jenny Lo, for providing the experimental perspective;
Laura Shaheen and Nancy Tetlow, for helping me with the nitty-gritty;
João Albuquerque, just for being himself; Juan Zamora, Vipul Jain, Chad Farschman and Craig Schmidt, for keeping up the excitement on the third and fourth floors.

Thanks are also due to my *desi* friends in Pittsburgh. I wish I could thank each and every one by name, who have played bridge and laughed at Hindi movies with me, and whose delicious *curries* and *pulavs* have made Appendices A and E possible. In particular, thanks to Prasad, Sudha, Sridhar R., George, Prem, Ashit, Sundar, Sita, Shravan, and my roommates, Sayam, Shenoy and Joga.

Finally, and most importantly, I would like to thank my family. My little sister, Vipula, deserves special thanks for her unending love and encouragement through our weekly conversations over the phone. I would not have made it through my Ph.D. without her support. I dedicate this thesis to my parents, Gopal and Premalatha, for instilling in me the value of scholarship and hard work. Thanks to *achhan* and *mummy*, for all the sacrifices they have made with their own careers, to dedicate significantly more time and attention to the education of their children. Both have been role-models for a life time of learning. I know my accomplishments mean more to them than they do to me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Process systems engineering has emerged as a major discipline in chemical engineering in the last two decades. Today, chemical industries perform process modeling, design, synthesis, analysis, simulation and optimization as integral part of their day to day activities. Research activity in process systems engineering is influenced heavily and often driven by the following two factors (Evans [45])

- Demands of the industry and society
- Advances in computer hardware technology

In view of the global economy, increased safety concerns and strict environmental regulations, demands from the industry for better concepts, ideas and tools from the process systems engineering community is higher than ever. This has resulted in increased challenges and a multitude of problems the process systems engineer could look at.

Advances in computer hardware technology has resulted in the increase of computational efficiency by orders of magnitude in recent years. As a consequence, this has helped process systems engineers to expand the horizons of what could be achieved in a realistic computational time. This trend in the availability of faster and cheaper processors is expected to continue and for the process engineer this means realization of what was deemed *impossible* a few years ago.

In view of the above, in this study, we look at an interesting class of problems arising in process simulation and optimization, quite relevant in industrial practice, and their solution methodologies.

1.1.1 Process Simulation

Recent years have seen the emergence of process simulation as a indispensable tool for plant design, analysis and operation. The first process simulator was developed about forty years ago and since then simulation technology has advanced by leaps and bounds that, today, no major chemical or petrochemical plant would be built without first simulating the process on the computer [45]. Process simulation has many benefits, among them being ecomomic experimentation, evaluation of alternate policies, experimental replication, test of parameter sensitivity and study of system safety and stability [59].

Steady state simulation is part and parcel of chemical engineering process design and analysis. Biegler [15] and Evans [46] provide a review of the state of the art in steady state process simulation. *Equation oriented* steady state simulators, as opposed to their *sequential modular* counterparts, rely on an equation solving engine which must be able to handle poor starting points, inequality constraints, singularities and ill-conditioning, among others. Process simulations also invariably encounter variables which need to be restricted within certain bounds. Typical examples are the nonnegativity restrictions on physical quantities like mass, volume or temperature. Process models could show abnormal behavior if these variables are allowed to take values beyond these bounds because of physical or mathematical infeasibility. A systematic treatment of these bounds by the solver can avoid convergence failures which might arise due to ad hoc strategies currently employed. Equation oriented simulators also need to deal with nonsmooth relations, characterized by discontinuous derivatives or kinks, which are often enforced as procedures in the sequential modular mode. Some of these issues have been addressed by Bullard [27] in the context of steady state simulation.

Of increasing interest in the simulation community is the development of efficient *dynamic simulation* tools. Many authors have surveyed the present and future applications of dynamic simulation tools in chemical process industries (Perkins [84], Marquardt [69], Naess et al. [76], Pantelides and Barton [81]). Marquardt [69] points out that most of the current dynamic simulation packages are developed to solve models of predominantly continuous nature. However, few industrial processes could be considered to operate in a truly continuous manner. Simulation of dynamic chemical processes often involves discrete actions and logical constraints and these can lead to discontinuities in the modeling equations. Process models can thus be nonsmooth at various points in state space, with these points specified through explicit or implicit relations. Typical examples of these include phase and flow transitions, heat and mass transfer correlations in various regimes and startup and shutdown operations in continuous plants. The discontinuity might cause either a switch in the model equations or even a change in the structure and dimensionality of the plant model. Hence an effective and consistent representation mechanism of the model discontinuities and their solution technique is very important in an equation based modeling environment.

In this thesis, we develop a new framework based on Successive Linear Programming (SLP) for the efficient solution of nonsmooth dynamic simulation problems.

1.1.2 Process Optimization

Mathematical programming approaches in process optimization in chemical engineering has evolved under two major heads.

- Nonlinear programming [16]
- Mixed Integer Approaches [55], [56], [86]

In this thesis, we will mainly confine ourselves to discussion of solution of continuous optimization problems described by nonlinear programming (NLP). However, efficient solution techniques for NLP could also aid in the solution of NLP subproblems arising in Mixed Integer Nonlinear Programming Problems (MINLP).

Nonlinear programming problems in process engineering are solved by two classes of approaches.

- Reduced gradient methods [75]
- Iterative techniques - SLP [103] and Successive Quadratic Programming (SQP) [57], [88], [14]

Of these approaches, SQP has become the method of choice for the solution of nonlinear programming problems. With the widespread use of SQP for process optimization in design, operations and control, a variety of different implementations and strategies have been developed for this algorithm. One important benefit for SQP is its ability to be modified in order to take advantage of a particular problem structure. This is especially important for large, sparse systems that include problems in parameter estimation [2], multiperiod optimization [98] and dynamic optimization [94]. However, one hurdle in exploiting the problem structure is an efficient strategy for treating inequality constraints in the QP subproblem. Here, while most of the commonly used QP codes use active set strategies, it has been observed that they can become combinatorially expensive for large problems. In linear programming, the simplex method (Dantzig [37]) searches at the vertices of the polyhedra described by the constraints and is also combinatorial.

Motivated by the search for an efficient technique for solving the highly constrained LP subproblems arising from the solution of nonsmooth dynamic simulation problems, we also looked at solution methodologies for large inequality constrained nonlinear programming problems to address the bottlenecks posed by the active set methods. In this thesis, we also describe a premise for addressing this problem.

1.1.3 Modeling and Differential Algebraic Equations

A large class of dynamic chemical processes can be modeled as a set of differential-algebraic equations (DAEs). The DAE consists of differential equations which describe the dynamic behavior of the system, like mass and energy balances, and algebraic equations which enforce physically meaningful constraints, like thermodynamic relations. The complexity associated with the solution of a DAE is related to its *index* which will be described in more detail in Chapter 3. Thus, any study of dynamic simulation and optimization of chemical processes calls for a systematic understanding of the nature and properties of DAE models describing them.

This thesis will also attempt to further the understanding of the problems in process dynamics through an analysis of the DAE describing it.

1.2 Thesis Description and Outline

In the first part of this thesis, we develop a method based on SLP for the treatment of variable bounds, inequalities and nonsmoothness occurring in process simulation problems. The SLP method converts the equation solving problem to an optimization problem, with no degrees of freedom at the solution. Chapter 2 describes an algorithmic improvement in SLP methods for solving constrained simulation problems. Numerical analysts have traditionally used one of the two major strategies, line search or trust region, for ensuring descent in globally convergent methods for nonlinear equation solving or optimization [40]. In this work, we present a new descent strategy combining both line search and trust region methods in the context of SLP. In the small example problems tested, the combined strategy is shown to be performing as well as either the line search or the trust region method, regardless of whether there is a difference in the individual performance of these methods on a particular problem.

Chapters 3 and 4 look at various aspects of discontinuities occuring in dynamic simulation problems and their solution technique based on SLP methods. The first step in performing dynamic simulation of chemical processes described by DAEs is to provide consistent initial conditions to the model. This task is far from trivial as the number of initial conditions one could independently specify is not known *a priori*. It could be any number less than or equal to the number of differential variables in the DAE and is mainly dictated by the properties and characteristics of the DAE system. Successive differentiation of the DAE could generate additional algebraic equations which limits the initial conditions which could be specified independently. It is well known that even slightly inconsistent initial conditions could lead to a failure of the simulation exercise. Hence, it is of vital importance that one specify consistent initial conditions to the solver to generate numerically accurate solutions. An equally relevant problem is the one concerning reinitialization. Discrete changes in the modeling equations are marked by the occurrence of an *event*, which is usually triggered by a certain condition becoming true (Barton [10]). Reinitialization after a discontinuity is an even tougher problem as it involves the additional exercise of determining the subset of variables which are continuous across a discontinuity. In Chapter 3, we develop an approach based on SLP for the initialization and reinitialization after discontinuities of DAEs.

The methods employed to solve the dynamic problem should also be able to determine the time of occurrence of events, which may not be known in advance. Thus the detection of state events calls for condition monitoring while the problem is being solved in a particular time interval and determining the exact time at which the conditions are satisfied. A related issue associated with these problems is the detection of the correct state event. Consider the scenario in which more than one condition is satisfied within short times. The solution algorithm should be able to determine which of those conditions is satisfied first and identify the event associated with it. This is important; if the wrong condition is detected as having been satisfied first, we might end up with an entirely different set of equations in the next phase. In short, we should be able to detect and locate the earliest event in a time interval while dealing with nonsmooth dynamic simulation problems. Chapter 4 describes a new algorithm for the treatment of discontinuities occuring in dynamic simulation problems. Here we see that LP-based methods can aid in event detection and resolution. The SLP method is also shown to provide a very natural way of incorporating variable bounds and physically realistic conditionals through simple inequalities. Examples are provided for the illustration of these ideas. We also consider an important class of problems that occurs most often in chemical process simulation in this chapter. It deals with the calculation of phase equilibrium. Phase equilibrium problems are nonsmooth in the sense that different sets of equations are valid depending on the number of phases present. An efficient method for dealing with this 'nonsmoothness' could lead to robustness and considerable savings in computation time. We present a single consistent optimization based formulation which will automatically solve the relevant set of equations without having to determine the number of phases present at equilibrium in advance. Example problems are drawn from liquid-vapor and liquid-liquid-vapor systems.

Having formulated and solved a broad range of problems in nonsmooth dynamic simulation ranging from initialization and reinitialization of DAEs to the detection and location of discontinuities with SLP based methods, our next focus was how to improve the efficiency in solving these SLP subproblems. The general nature of these problems is that they are highly constrained, with the constraints arising from physically realistic conditionals and variable bounds. In fact, this is exactly the nature of a large fraction of continuous optimization problems arising in chemical process engineering. This motivated us to look at this problem from a larger perspective and address the bottleneck issue of the complexity associated with active set methods in nonlinear optimization as described in section 1.1.2.

Chapter 5 describes incorporation of the concepts of interior point methods within a nonlinear programming framework. Interior point methods, first introduced by Karmarkar [62] for linear programming problems in 1984, has polynomial complexity and is preferred for solving large scale LP problems. We consider a class of primal-dual interior point methods, a Mehrotra predictor-corrector method [72], for our study. We also provide a novel interpretation of predictor-corrector methods using reduced Hessian SQP (rSQP). Incorporation of interior point methods within an SQP framework, which we henceforth call Interior point SQP (ISQP), is demonstrated to overcome the combinatorial complexity associated with active set methods. Example problems are drawn from optimal control.

Efficient methods for the treatment of complementarity conditions arising from the optimality conditions of a nonlinear programming problem is investigated in Chapter 6. The motivation for this study comes from the solution of the implicit discontinuities described in Chapter 4. Direct enforcement of the complementarity conditions could be problematic because of the inherent nondifferentiability and singularity associated with them. Chapter 6 looks at a class of smoothing methods for the treatment of smoothing methods. Applicability of smoothing methods for modeling other nondifferentiable functions such as the absolute value, nested min-max operators is also studied. As part of practical applications, simulation of distillation columns where dry or vaporless trays could be present are considered. Standard formulations with predetermined number of phases could fail in such a scenario. An equation based formulation with complementarity conditions derived from the optimization formulation described in chapter 4 is used for the solution of the distillation examples.

Finally, Chapter 7 summarises the contributions of this research and potential directions for future work.

Chapter 2

An Improved Successive Linear Programming Algorithm

Among numerical analysts, there is a strong suspicion
that if any iterative method for any problem
in any field is exceptionally effective,
then it is Newton's method in some context.
-Dennis and Schnabel [39]

Synopsis

Bullard and Biegler [24] developed a Successive Linear Programming (SLP) algorithm for solving a system of nonlinear equations constrained by inequalities and variable bounds. In this chapter, we modify the SLP algorithm by introducing a descent strategy combining two major classes of descent: line search and trust region methods. Theoretical and global convergence properties of the new method are also developed in this chapter. Example problems considered indicate that the combined line search - trust region framework performs as well as the better individual method in almost all problems.

2.1 Introduction

Consider the solution of a nonlinear system of equations subject to inequality constraints and variable bounds.

$$\begin{aligned} & h_j(x) = 0 \\ & g_k(x) \leq 0 \\ & x^L \leq x \leq x^U \end{aligned} \tag{CSP}$$

This problem is termed as the *constrained simulation problem* (CSP). The citation from [39] at the beginning of this chapter would suggest that an effective method to solve (CSP) would have its roots in a Newton's method. However, direct application of the basic Newton's method to solve (CSP) could be problematic for a variety of reasons including its inability to handle inequality constraints or a singular Jacobian. This requires reformulation of Newton's method which retains its quadratic convergence rate while avoiding its numerical problems.

Bullard and Biegler [24] introduced a technique based on successive linear programming for solving (CSP). The basic idea of the approach is to convert the equation solving problem to an optimization problem. A sequence of linear programs based on the reformulation of a Newton's method is solved, which yields search directions leading to the solution of the nonlinear problem.

The method is described in detail in section 2.2.

2.2 SLP Method for Constrained Simulation Problems

To solve (CSP), a merit function is defined as

$$\mu(x_i) \quad = \quad \sum_j |h_j(x_i)| + \sum_k g_k(x_i)_+ \tag{2.1}$$

where $g_k(x_i)_+ = max\,[0, g_k(x_i)]$. Since $\mu \geq 0$ has a minimum of zero if and only if (CSP) is solved, μ, a non differentiable function, is minimized subject to $x^L \leq x \leq x^U$. By adding auxiliary variables, this problem can be written as :

$$\begin{aligned} Min \quad & \sum_j (p_j + n_j) + \sum_k s_k \\ s.t. \quad & h_j(x) = p_j - n_j \\ & g_k(x) \leq s_k \\ & x^L \leq x \leq x^U \\ & p_j, n_j, s_k \geq 0 \end{aligned} \tag{2.2}$$

Barrodale and Roberts [8] proved that for $\sum_j p_j + n_j$ at a minimum, a complementarity condition $p_j n_j = 0$ holds. Thus $(p_j + n_j)$ represents $|h_j(x)|$ and s_k represents $g_k(x)_+$. Linearizing the constraints about x_i leads to the following constrained simulation linear program, CSLP:

$$\begin{aligned} Min \quad & \sum_j (p_j + n_j) + \sum_k s_k \\ s.t. \quad & h_j(x_i) + \nabla h_j(x_i)^T d_i = p_j - n_j \\ & g_k(x_i) + \nabla g_k(x_i)^T d_i \leq s_k \\ & x^L \leq x_i + d_i \leq x^U \\ & p_j, n_j, s_k \geq 0 \end{aligned} \tag{CSLP}$$

The solution of (CSLP) generates a search direction d^i. To stabilize the performance of the algorithm, a line search is carried out to find some fraction α so that $x_{i+1} = x_i + \alpha d_i$. It should be noted that this method reduces to Newton's method in the absence of active inequality constraints and thus has the ability to converge quadratically to the solution. This LP approach is also shown to generate a descent direction for the 1-norm of the constraint violations. The method has global convergence properties for a nonzero solution of the linear program. The algorithm, when tested on problems ranging from 2 to 977 equations, compared favorably with the existing methods.

In the next section, to improve the performance of this method, we incorporate a new descent strategy into this algorithm. The new strategy combines two major classes of strategies used to ensure descent: line search and trust region methods. Dennis and Schnabel [40] and Fletcher [48] provide a detailed study of these methods.

2.3 A New Descent Strategy

2.3.1 Line Search Methods

The basic concept behind using line search methods is to choose a descent direction d_i from the current point x_i and select an *acceptable* point x_{i+1} in this direction at which $\mu(x)$ decreases. The implementation for the algorithm by Bullard and Biegler [24] uses the traditional monotonic, Armijo-type line search [4]. The search direction d_i is obtained from the solution of (CSLP) and a condition (2.3) that requires a sufficient decrease in the merit function is enforced.

$$\mu(x_i + \alpha d_i) - \mu(x_i) \leq 0.1\alpha\mu^o \tag{2.3}$$

μ^o is the directional derivative defined below.

Definition 2.1 *[36] The generalized directional derivative of μ at x in the direction $d \in \Re^n$ is*

$$\mu^o(x; d) \triangleq \limsup_{\substack{y \to x \\ t \downarrow 0}} \frac{\mu(y + td) - \mu(y)}{t} \tag{2.4}$$

□

We can also show the following result for the directional derivative.

Lemma 2.2 *For a scalar $\lambda > 0$,*

$$\mu^o(x; \lambda d) = \lambda\mu^o(x; d) \tag{2.5}$$

Proof is straightforward. □

From the search direction generated by CSLP, an upper bound on the directional derivative of the merit function, $D_d\mu$, can be derived (2.6), using p_j, n_j and s_k obtained from the solution of the LP.

$$D_d\mu = \sum_j (p_j + n_j) + \sum_k s_k - \sum_j \left|h_j(x^i)\right| - \sum_k g_k(x^i)_+ \tag{2.6}$$

At the full step ($\alpha = 1$), (2.3) is evaluated. If satisfied, the variables are updated and a new step is calculated. Otherwise a new fractional step (2.7) is calculated from a quadratic interpolation of the merit function and continued until a suitable α is found.

$$\alpha = max\left\{0.1\alpha, \frac{-0.5\alpha^2\mu^o}{\mu\,(x^i + \alpha d^i) - \mu(x^i) - \alpha\mu^o}\right\} \tag{2.7}$$

2.3.2 Trust Region Methods

Search directions generated by CSLP are always descent directions ($\mu^o < 0$) for $d \neq 0$,, but they still may be poor if the problem is ill-conditioned. In such cases, the search direction is relatively large and is nearly orthogonal to the steepest descent direction. Trust region methods avoid this problem by restricting the stepsize d_i to an area Δ_i around the current point x_i. The trust region is frequently adjusted after each step depending on how valid the approximations of the model are. Most of the current implementations increase or decrease the trust region by comparing the *actual* reduction in the merit function and the *predicted* reduction on linearization. The closer their ratio is to unity, the better the agreement. Duff et al. [43] describe this strategy in the context of using linear programs to solve sparse nonlinear equations.

Since the merit function is nondifferentiable, it is possible for the *Maratos effect* to occur in both line search and trust region methods (Fletcher [48]). Here x^i may be arbitrarily close to the solution x^*, but a full step may fail to reduce the merit function. A nonmonotonic line search such as the watchdog technique (Chamberlain *et al.* [31]) or a second order correction (Fletcher [48]) could be used to overcome this effect.

2.3.3 Combination

The main difference between line search and trust region methods lies in the sequence in which they use the linear model and enforce the acceptable step length. The line search methods use a linear model to obtain a search direction d and then choose a fractional step α. Thus the line search occurs on a 1-dimensional subspace. On the other hand, trust region methods first choose a maximum acceptable step length Δ and then use the linear model to obtain a search direction d, the length of which cannot exceed Δ. The search here is not restricted to a lower dimensional subspace. The trust region step may not always be in the Newton direction, particularly when Δ is small when compared to the Newton step. Instead, as Δ goes to zero, the LP direction becomes the closest vertex to the steepest descent direction. Since the trust region becomes inactive near the solution and there are no degrees of freedom, both the Newton and LP-based method exhibit quadratic convergence near a nonsingular solution.

Relatively little work is necessary for the line search method as only one LP is solved at each iteration. But, when trust region methods are used, finding the ideal trust region bounds is likely to require solution of more than one LP per iteration. However, trust region methods require fewer major iterations as they use a Newton step if it lies within the trust region.

In a combination of these two methods, the trust region eliminates the possibility of a large ill-conditioned LP step which results in slow convergence. On the other hand, a line search helps to reduce the number of times the LP should be solved, leading to less computational expense. First the trust region constraints are added to the linear model to generate a search direction d_i. A line search is done in this direction to get the new point x_{i+1}. The predicted and actual reduction in merit function are then calculated on the basis of the new point and the size of the trust region is then updated. The detailed algorithm is given below.

0. $i = 0$

   Set trust region bounds

   $\bar{\Delta} = 0.5 \max\left(x_0 - x^L, x^U - x_0\right) \qquad \gamma_0 = 1$

   Initialize the problem (at the solution of the previous time step in the dynamic problem)

1. Evaluate $h, \nabla h, g, \nabla g$ (constraints and their gradients).

2. Solve $CSLP(x_i, d_i)$ with trust region constraints.

3. Compute *predicted* change in exact penalty by linearization
   $\mu_l(x_i, d_i) - \mu(x_i) = \Delta\mu_{l,i}$, where
   $\mu_l(x_i, d_i) = \sum_j \left|h_j(x_i) + \nabla h_j(x_i)^T d_i\right| + \sum_k \left(g_k(x_i) + \nabla g_k(x_i)^T d_i\right)_+$
   If $|\Delta\mu_{l,i}| \leq \epsilon_1$, return. Else goto 4.

4. Set $\alpha = 1$. $n_l = 1$.

5. $x^N = x_i + \alpha d_i$. Calculate μ_N (exact penalty at the new point x^N)

6. If the actual change in exact penalty $\Delta\mu_i = \mu_N - \mu(x_i) \leq 0.1\alpha\mu^o$, go to 7.
   Else if $n_l \leq 5$,
   set $\alpha = \max\left\{0.1\alpha, \frac{-0.5\alpha^2\mu^o}{\mu_N - \mu(x_i) - \alpha\mu^o}\right\}$. $n_l = n_l + 1$. Goto 5
   Else if $n_l > 5$,
   Resolve trust region subproblem with a reduced trust region. Goto 8.

7. New point $x_{i+1} = x_i + \alpha d_i$.

8. Set the trust region radius for the next iteration.
   Calculate the ratio of *actual* to *predicted* change
   $$r_i(\alpha d) = \frac{\mu(x_i + \alpha d_i) - \mu(x_i)}{\mu_l(x_i, \alpha d_i) - \mu(x_i)} \tag{2.8}$$
   $$\gamma_{i+1} = \begin{cases} \gamma_i/m & \text{if } r_i < \rho_1 \\ m\gamma_i & \text{if } r_i > \rho_2 \quad \} \\ \gamma_i & \text{otherwise} \end{cases}$$
   The next trust region $\Delta_{i+1} = \bar{\Delta}\max(\gamma_{i+1}, \gamma')$

9. $i = i + 1$. Go to 2.

In the numerical examples described in section 2.5, parameters ρ_1, ρ_2 and m are fixed at 0.25, 0.75 and 2 respectively, similar to the values used by Zhang et al. [103]. Also, the upper bound for the directional derivative μ^o given by (2.6) is used for numerical computations. γ' is set to 10^{-3}. Note that if (2.3) is not satisfied and the line search fails after a prespecified number of trials (5 in the above algorithm), $r_i < 0.10$ and γ is automatically decreased. Even though it is shown in theorem 2.9 that a point satisfying the line search criterion can always be found, we impose a maximum of 5 line search trials to avoid small steps and for more efficient practical implementation. The convergence criteria employed here is $|\Delta\mu_{l,i}| \leq \epsilon_1$ (step 3), and this occurs at a stationary point of $\mu(x)$. This criterion is somewhat conservative and, if, in addition, $\mu_i \leq \epsilon_2$, then the problem is solved. Otherwise, x_i is only a stationary point which we term as a *pseudosolution.* Heuristic strategies for recovering from a pseudosolution (e.g. rescaling the problem) are described by Bullard and Biegler [24].

In the following section we consider some theoretical properties on this framework.

2.4 Theoretical Properties

2.4.1 Definitions and Assumptions

In this section, we present global convergence properties of the combined line search - trust region method. For this purpose, we first define some terms used in the convergence analysis. The definitions are from Clarke [36] and Dennis *et al.* [38].

Definition 2.3 *A function $\mu : \Re^n \to \Re$ is said to be (locally) Lipschitz near x if for some constant $K > 0$ and $\epsilon > 0$, μ satisfies the Lipschitz condition*

$$| \mu(x_1) - \mu(x_2) | \leq K \parallel x_1 - x_2 \parallel$$

for all x_1, x_2 in the ϵ-neighborhood $N(x) = \{y \mid \parallel y - x \parallel < \epsilon\}$ of x.

□

Definition 2.4 *A function μ that is (locally) Lipschitz near x is said to be regular at x if the one-sided directional derivative*

$$\mu'(x; d) \triangleq lim_{t \downarrow 0} \frac{\mu(x + td) - \mu(x)}{t}$$

exists for all directions $d \in \Re^n$ and

$$\mu'(x; d) = \mu^o(x; d)$$

where $\mu^o(x; d)$ is the generalized directional derivative at x defined by (2.4). □

Let $L_0 = \{x \mid \mu(x) \leq \mu(x_0)\}$ be the level set of f at x_0, the starting point of the algorithm. We make the following assumptions for μ and its linear approximation μ_l.

Assumptions 2.5 *The following assumptions will be made for the analysis in section 2.4.2.*

- *μ is regular on L_0 and for every $x \in L_0$, the linear approximation $\mu_l(x, d)$ is regular with respect to $d \in \Re^n$.*
- *For every $d \in \Re^n$, the linear approximation $\mu_l(x, d)$ is continuous in x.* □

2.4.2 Global Convergence

Having defined the terms and stated the assumptions, in this section, we develop a global convergence analysis for the combined line search - trust region approach. The following analysis builds upon the unified global convergence analysis of pure trust region methods for nonsmooth optimization developed by Dennis *et al.* [38]. Below, we show that incorporation of line search within that framework retains the global convergence properties.

Lemma 2.6 *[38] If μ is Lipschitz continuous with constant K near x, then for every $d \in \Re^n$, $\mu^o(x; d)$ exists and*

$$| \mu^o(x; d) | \leq K \parallel d \parallel$$

□

Lemma 2.7 *[38] If μ is Lipschitz continuous with constant K near x, then $0 \in \partial\mu(x)$ iff $\mu^o(x; d) \geq 0$ for all d in $\Re^n$.* □

The following theorem shows existence of a descent direction.

Theorem 2.8 *At any x, not a stationary point of μ, there exists a direction d_i, where $\parallel d_i \parallel \leq \Delta_i$, ($\Delta_i \neq 0$), which satisfies $\mu^o(x; d_i) < 0$.*

Proof

From Lemma 2.6 and 2.7, if $0 \notin \partial\mu(x)$, then

$$\mu^o(x; d) < 0 \; for \; some \; \hat{d}_i. \tag{2.9}$$

Consider the two cases.

1. If a $\hat{d}_i$ which satisfies 2.9 is $\leq \Delta_i$, $d_i = \hat{d}_i$, proof is complete.

2. If a $\hat{d}_i$ which satisfies (2.9) is $> \Delta_i$, then $\exists$ some $\lambda > 0$ s.t. $\| \lambda \hat{d}_i \| \leq \Delta_i$. Set $d_i = \lambda \hat{d}_i$. $\mu^o(x; d_i) = \mu^o(x; \lambda \hat{d}_i) \overset{Lemma\ 2.2}{=} \lambda \mu^o(x; \hat{d}_i) < 0$. □

The next theorem shows the existence of a point which satisfies the line search criteria on the search direction obtained from the trust region subproblem.

Theorem 2.9 *Given a descent direction d_i from Theorem 2.8, $\exists$ α s.t.*
$x_{i+1} = x_i + \alpha d_i, \qquad \alpha \in (\alpha_l, 1) \qquad \alpha_l > 0.$
which satisfies the Armijo criteria

$$\mu(x_i + \alpha d_i) \leq \mu(x_i) + c_A \alpha \mu^o(x_i, d_i) \qquad 0 < c_A < 1 \tag{2.10}$$

Proof
We will prove by contradiction. Let us assume that the Armijo criteria (2.10) failed for all $\alpha > \tilde{\alpha}$. Then for all $\alpha > \tilde{\alpha}$, $\mu(x_i + \alpha d_i) > \mu(x_i) + c_A \alpha \mu^o(x_i, d_i)$. We also have the expansion

$$\mu(x_i + \alpha d_i) \leq \mu(x_i) + \alpha \mu^o(x_i, d_i) + \alpha^2 K_1 \| d_i \|^2 \tag{2.11}$$

for some $K_1 > 0$.
From the above two inequalities, we have, $\mu^o(x_i, d_i) + \alpha K_1 \| d_i \|^2 > c_A \mu^o(x_i, d_i)$. *i.e.*

$$\alpha > \frac{(c_A - 1)\mu^o(x_i, d_i)}{K_1 \| d_i \|^2} \tag{2.12}$$

We also have $\| d_i \| \leq \Delta_i$. (2.12) becomes

$$\alpha > \frac{(c_A - 1)\mu^o(x_i, d_i)}{K_1 \Delta_i^2} = \alpha_l \tag{2.13}$$

Fractional step lengths α which do not satisfy Armijo criteria is bounded below. ie $\exists$ α_i which satisfies (2.10) when a line search is carried out on the search direction obtained from theorem 2.8. □

The next theorem from Dennis *et al.* [38] shows that in a pure trust region method, we can always find a point in the neighborhood of the current point, where the criteria for the acceptance of a new point is satisfied. The proof is lengthy and the reader is referred to the original paper.

Theorem 2.10 *Let $\hat{x} \in L_0$ and $\rho_1 \in (0,1)$. Under Assumptions 2.5, if $\hat{x}$ is not a stationary point of μ, then $\exists\ \hat{\Delta} > 0$ and $\epsilon > 0$, such that, for every x and Δ satisfying $\| x - \hat{x} \| < \epsilon$ and $0 < \Delta \leq \hat{\Delta}$, we have, $r(d) > \rho_1$ for any d obtained from the trust region subproblem.* $\square$

In the next theorem we will show that an Armijo line search will take full steps if the search direction obtained is acceptable in a pure trust region method.

Theorem 2.11 *An acceptable point in a pure trust region framework satisfies Armijo criteria (2.10) with $c_A \leq \rho_1$ for a full step.*

Proof

A point is accepted in a pure trust region method if

$$\begin{aligned} & r_i = \frac{\Delta\mu_i}{\Delta\mu_{l,i}} > \rho_1 \\ \Rightarrow\quad & \mu_i - \mu_{i+1} > \rho_1\alpha(-\mu_i^o) \\ \Rightarrow\quad & \mu_{i+1} < \mu_i + \rho_1\alpha\mu_i^o \end{aligned}$$

For $\alpha = 1$, $c_A \leq \rho_1$ in (2.10), the Armijo criteria is satisfied. $\square$

The implications of theorem 2.10 is that the trust region update will always maintain a finite trust region and will not eventually shrink to a point. Theorem 2.11 shows that the line search takes full steps if the trust region is as small as the neighborhood described in theorem 2.10.

Finally, the global convergence proof.

Theorem 2.12 *The combined line search - trust region strategy has global convergence properties, i.e., if x^* is an accumulation point generated by this algorithm, then x^* is a stationary point of μ.*

Proof

Every accepted point x_{i+1} satisfies the Armijo criteria (2.10). Consider the summation

$$\begin{aligned} \infty &> [\mu(x_0) - \mu(x^*)] \\ &= \sum_{i=0}^{\infty} [\mu(x_i) - \mu(x_{i+1})] \\ &\geq -c_A \sum_{i=0}^{\infty} \alpha_i \mu_i^0(x_i; d_i) \end{aligned} \tag{2.14}$$

μ is bounded below and the summation on the right hand side of (2.14) is bounded. This implies

$$\alpha_i \mu_i^0 \to 0 \quad as \quad i \to \infty$$

To show that $\mu_i^0 \to 0$, we need to show that the case $\mu_i^0 \to \mu^{0*} < 0$ and $\alpha_i \to \alpha^* \to 0$ does not exist.

From (2.13) we can show that

$$\alpha^* > \frac{(c_A - 1)\mu^{o*}}{K_1 \Delta^{*2}} \tag{2.15}$$

From (2.15) it is clear that α^* is bounded below by a function of μ^{0*} and hence, the case $\mu^{0*} < 0$ and $\alpha^* \to 0$ cannot hold. $\square$

2.5 Numerical Examples

The SLP method with the new descent strategy was tested on 14 small problems reported in the literature. All the problems were solved using a Fortran program OPTLP which implements the combined line search/trust region descent strategy. It uses QPSOL (Gill et al. [52]) for the solution of the LPs and implements a relaxation formulation to handle inconsistent linearizations. The tolerance specified was that the 1-norm of the constraint violations be less than 10^{-7}. The numerical results for the solution of these problems without scaling are listed in Table 2.1. The individual problem descriptions could be found in Bullard and Biegler [24] (The numbers in parentheses in column 1 refer to the problem numbers used in that paper).

Table 2.1: Comparison of number of LPs solved with pure line search, pure trust region and a combination of line search and trust region methods.

Problem		Number of Variables	Number of LPs solved		
			Line Search	Line Search + Trust Region	Trust Region
1	(21)	1	18	5	5
2	(10)	2	12	12	21
3	(17)	4	11	11	18
4	(11)	4	15	14	27
5	(1^a)	2	10	10	14
	(1^b)	2	10	9	9
	(1^c)	2	10	10	10
6	(2)	2	6	6	6
7	(4)	2	14	13	13
8	(7)	2	7	7	7
9	(8)	2	5	5	5
10	(12)	2	4	4	4
11	(15)	2	5	5	5
12	(16)	2	6	6	6
13	(9)	3	6	6	6
14	(19^a)	7	8	8	pseudo-solution
	(19^b)	7	15	15	> 100
	(19^c)	7	9	9	11

The question which arises in the context of descent in this framework is when to apply which strategy. In general, line search methods perform better than trust region methods on well-conditioned problems, whereas trust region methods show better performance with ill-conditioned ones. Both methods perform similarly on most of these problems.

For problem 1 a pure trust region method (5 LPs solved) is a better strategy than a pure line search method (18 LPs solved). However, the combined method solved the problem in 5 iterations, showing a performance similar to the better trust region method. On the other hand, line search shows better performance than trust region in problems 2,3,4 and 14. From the table it is evident that the combined method shows a performance similar to the better strategy here, the line search. Hence, for the test problems considered, when there is a significant difference in the two descent methods, the combined strategy shows a performance similar to the better method.

For problems 5-13 there is not much difference between the line search and trust region methods as the number of LPs solved is almost the same. The combined method is as good as either of them. Thus, in problems where there is no significant difference between a line search and a trust region approach, a combination of the two methods does not lead to a decrease in computational efficiency.

These observations are significant because the combined method performs at least as well as either the trust region or the line search methods on all the problems, regardless of whether there is a difference between the individual performance of these methods on a particular problem. The combined strategy is expected to perform better for larger problems as well.

2.6 Conclusions

A previously developed SLP method was improved with the incorporation of a new descent strategy. The new descent criterion combines line search and trust region methods. The question of whether a line search or a trust region method should be used, is problem dependent. It has been understood that trust region methods perform better on ill-conditioned problems whereas line search methods perform well on well-conditioned ones. On the other hand, trust region methods are relatively more expensive than line search methods. In our study described in this chapter, we described a framework combining these two methods in the context of SLP. Global convergence analysis has also been carried out for the combined method. The results from example problems demonstrate that a combination shows a performance as good as the better of the individual methods in almost all problems. *ie.* the combined method performs as well as trust region methods for ill-conditioned problems and is comparable to line search methods for well-conditioned problems. Though it would be premature to claim that the combined method is an answer to the question of which of line search or trust region methods is better in a general context, our analysis shows that it is definitely a direction to be researched in detail in the future. Our initial results show that it has vast potential.

In chapter 3, we use SLP based methods described in this chapter for the initialization and reinitialization of process models described by differential algebraic equations. Further, in chapter 4, we describe how SLP can aid in the treatment of discontinuities occuring in dynamic process simulation.

Chapter 3

Initialization and Reinitialization after Discontinuities of Differential Algebraic Equations

As the proverb says,
'a good beginning is half the business', and
'to have begun well' is praised by all.
-Plato, Laws [87]

Synopsis

Determination of consistent initial conditions is an important aspect of the solution of differential algebraic equations (DAEs). Specification of inconsistent initial conditions, even if they are slightly inconsistent, often leads to a failure in the initialization problem. In this chapter, we present a Successive Linear Programming (SLP) approach for the solution of the DAE derivative array equations for the initialization problem. The SLP formulation handles roundoff errors and inconsistent user specifications among others and allows for reliable convergence strategies that incorporate variable bounds and trust region concepts. A new consistent set of initial conditions is obtained by minimizing the deviation of the variable values from the specified ones. For problems with discontinuities caused by a step change in the input functions, a new criterion is presented for identifying the subset of variables which are continuous across the discontinuity. The SLP formulation is then applied to determine a consistent set of initial conditions for further solution of the problem in the domain after the discontinuity. Numerous example problems are solved to illustrate these concepts.

3.1 Introduction

Differential Algebraic Equations (DAEs) occur frequently in a variety of scientific and engineering problems. A large class of dynamic chemical process models fall into the category of DAEs. In this chapter, we carry out a basic study of DAEs in general, which has applications in a variety of fields in science and engineering. For our study, we consider the form:

$$F(x', x, y, t) \quad = \quad 0 \tag{3.1}$$

where $F \in \Re^n, x, x' \in \Re^{n_d}, y \in \Re^{n-n_d}$ and x' is the time derivative of x. DAEs are characterized by their *index* which is defined (Brenan *et al.* [20]) as the minimum number of times F, or the equations derived from it, should be differentiated in order to determine v' as a continuous function of v, t where

$$v = \{x, y\}, v \in \Re^n. \tag{3.2}$$

Determination of consistent initial conditions for DAEs has been of considerable interest among researchers dealing with systems of the form (3.1). Specifically, in process engineering, determination of consistent initial conditions is the first step in performing dynamic simulation using process models. Differentiation of (3.1) could generate new algebraic equations. These *hidden* algebraic equations in (3.1) impose constraints on the initial conditions of variables which can be specified independently. Thus the consistent initialization problem requires identification of these constraints, implicitly or explicitly, and their treatment along with the original set of equations.

Pantelides [79] utilizes the property that the hidden or *latent* equations are generated from those subsets of equations in which the number of variables (x', y) present in the subset is less than the number of equations in the subset. He uses a graph-theoretical algorithm to locate the *minimally structurally singular* subsets and differentiates them. However, the algorithm is structural and subsets of equations which are numerically singular might escape detection.

Unger *et al.* [97] also present an algorithm based on structural analysis. They determine the structural rank and the structural index of the problem by using a structural version of the symbolic algorithm for index reduction by Gear [51]. Gear's algorithm can be viewed as an extension of the algorithm by Bachmann *et al.* [5] for linear systems. Here the structural rank and index of the DAE (3.1) is solely determined by the patterns of $\partial F/\partial v'$ and $\partial F/\partial v$. However the structurally determined index and degrees of freedom will only provide a lower bound and upper bound respectively for their corresponding differential quantities. They also proposed a combination of structural methods with numerical/symbolic methods to overcome these difficulties as a direction for future research.

Campbell and Moore [28] proposed a method for solving nonlinear, high index problems for which BDF and Implicit RK methods might be inappropriate. They define the derivative array equations

$$F^{\nu}(x, z, z', \dots, z^{(\nu)}, t) = 0 \tag{3.3}$$

as the set of equations derived by differentiating the original DAE (3.1) ν number of times where ν is the index of (3.1) and

$$z = \{x', y\} \tag{3.4}$$

They solve $F^{\nu} = 0$ by successive solution of least squares problems. The method was found to be computationally expensive especially for computing the singular value decomposition of $J_{F^{\nu}}$ (the Jacobian of the derivative array equations) used for the least squares solution.

Leimkuhler *et al.* [65] characterize the consistency requirement by posing a set of equations which comprise the derivative array equations (3.3) (defined at time t_0) and the set of user specified information on initial conditions (3.5).

$$B(x_0', x_0, y_0) = 0 \tag{3.5}$$

The higher derivatives are approximated by forward finite differences. The resulting approximate system, however, might not have a solution and hence is solved in a least squares sense. This is complicated as well because of the rank deficiency of the Jacobian. The method is illustrated for index one and semi explicit index 2 problems in the Hessenberg form.

Chung and Westerberg [35] and Chung [34] proposed a numerical algorithm for consistent initialization by identifying singular subsets of equations from the Jacobian and symbolically differentiating them. Majer *et al.* [67] consider the problem of reinitialization of DAEs after discontinuities which we also discuss in detail in section 3.3.

Leimkuhler *et al.* [65] defines the consistent initialization problem as: *given specified information about the initial state of the problem that is sufficient to specify a unique solution to a DAE, determine the complete vector* $(v(t_0), v'(t_0))$. However, for a particular problem, the user may not know which variables to specify to sufficiently determine an exact solution for the initial condition vector. In other words, the user does not know the degrees of freedom a priori. On the other hand, the user may know a certain set of specifications and we would like our initial conditions to be as close to those given by the user (whether it is over- or exactly specified). Many approaches analyze the set of equations and ask the user for the initial conditions for a particular set of variables, which the user might not have. Hence, it would make sense to work the other way. Given the user specified set of initial conditions we would like to find a consistent initial vector (x', x, y) closest to the specified set. In many practical applications, the exact numerical value of the initial conditions may not be known for satisfying the relevant set of equations within the specified error criteria. The minimization formulation which we present in section 3.2 has its significance in this context. The method is also relevant when there are roundoff errors.

In the next section we develop a successive linear programming (SLP) formulation for the consistent initialization of DAE systems. The motivation for this approach is that the derivative array equations require solution of an underdetermined system, but roundoff errors and incomplete user specifications may render this system inconsistent. An SLP formulation handles this underdetermined nature efficiently and allows for reliable convergence strategies that incorporate variable bounds and trust region concepts. Section 3.3 deals with the related problem of reinitialization after input discontinuities are encountered for the DAE system. Here an analysis on the continuity of the state profiles is presented and compatible SLP formulations are derived for this problem. The SLP formulations in both sections are illustrated with the successful treatment of numerous high index examples. Section 3.4 summarises the chapter. The bulk of the material presented in this chapter could also be found in Gopal and Biegler [54].

3.2 Formulation for Consistent Initialization

3.2.1 The Problem and its Properties

To derive the SLP formulation we first consider properties of the general DAE system (3.1) and the associated derivative array equations (3.3). Let A, B and C be the Jacobian of F in (3.1) with respect to x, y and x' respectively. Thus

$$J_F = \begin{array}{c} \begin{array}{ccc} x & y & x' \end{array} \\ \left[\begin{array}{ccc} A & B & C \end{array}\right] \end{array} \tag{3.6}$$

On differentiating, F' will have terms of x, y, x', y' and x''. Now the Jacobian of F' with respect to these variables is related to the arrays A, B and C in J_F and the Jacobian with respect to x and y will be A' and B' respectively. With respect to x', the terms involving x' in F' come from terms involving x or x' in F. Thus, the derivative of F' with respect to x' will be $A + C'$. By similar arguments it is not difficult to see that the derivative of F' with respect to y' and x'' will be B and C respectively. If we continue the same pattern, the Jacobian of the derivative array equations will look like:

$$J_{F^\nu} = \begin{array}{cc} \begin{array}{cccccccccc} x & y & x' & y' & x'' & y'' & x''' & y''' & x^{(4)} & .. \end{array} & \\ \left[\begin{array}{cccccccccc} A & B & C & & & & & & & \\ A' & B' & A+C' & B & C & & & & & \\ A'' & B'' & 2A'+C'' & 2B' & A+2C' & B & C & & & \\ A''' & B''' & 3A''+C''' & 3B'' & 3A'+3C'' & 3B' & A+3C' & B & C & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{array}\right] & \begin{array}{c} F \\ F' \\ F'' \\ F''' \\ \vdots \end{array} \end{array} \tag{3.7}$$

Generalizing, the Jacobian of $F^{(i)}$ with respect to $x^{(m)}$ (or $y^{(m)}$) is the sum of the element in the Jacobian of $F^{(i-1)}$ with respect to $x^{(m-1)}$ (or $y^{(m-1)}$) and the time derivative of $F^{(i-1)}$ with respect to $x^{(m)}$ (or $y^{(m)}$). Note also that for a constant coefficient linear DAE, (3.7) simplifies to

$$J_{F^\nu,linear} = \begin{array}{c} \begin{array}{cccccccccc} x & y & x' & y' & x'' & y'' & x''' & y''' & x^{(4)} & .. \end{array} \\ \left[\begin{array}{cccccccccc} A & B & C & & & & & & & \\ & & A & B & C & & & & & \\ & & & & A & B & C & & & \\ & & & & & & A & B & C & \\ & & & & & & & & \vdots & \vdots \end{array}\right] \end{array} \begin{array}{c} \\ F \\ F' \\ F'' \\ F''' \\ \vdots \end{array} \tag{3.8}$$

The derivative array equations consist of an underdetermined system of $(\nu+1)n + n_d$ variables and $(\nu+1)n$ equations.

We still need an efficient method for the solution of this nonlinear underdetermined system. Additional problems include inconsistent linearizations, roundoff errors and poor starting points and user specifications. To deal with this problem we propose the following optimization formulation which we solve with SLP. Here we need to assume that the index ν is known and, if needed, the degrees of freedom r for the initial conditions can be determined.

3.2.2 SLP Formulation

In chapter 2, we considered the solution of a nonlinear system of equations subject to inequality constraints and variable bounds (CSP).

$$\begin{aligned} & h(w) = 0 \\ & g(w) \le 0 \\ & w^L \le w \le w^U \end{aligned} \tag{CSP}$$

Here we define $h(w)$ as the derivative array equations, (3.3), and the variable vector w as the entire set of variables in these equations, $(x, z, z', \ldots, z^{(\nu)})$. Chapter 2 also describes an SLP approach for solving (CSP) through the solution of a sequence of (CSLP)s. A steplength algorithm combining both line search and trust region methods has been shown to enhance the computational efficiency and increase the robustness of the algorithm. This approach has the advantage that it can easily be extended to large, sparse systems of equations.

We can also extend problem (CSP) and the SLP approach by minimizing some distance from the specified conditions for the initialization problem. Let the user-specified initial conditions be $X_i = X_i^{sp}$ for $i = 1, \dots n_{ic}$ where $X \subseteq \{x, x', y\}$. An optimization formulation for consistent initialization is

$$\begin{aligned}
& Min \sum_{i=1}^{n_{ic}} \omega_i \delta_i \\
s.t. \quad & F^{\nu}(x, z, z', \dots, z^{(\nu)}, t) = 0 \\
& -\delta_i \le X_i - X_i^{sp} \le \delta_i \qquad i = 1, \dots, n_{ic} \\
& \delta_i \ge 0 \qquad i = 1, \dots, n_{ic} \\
& X_i^L \le X_i \le X_i^U
\end{aligned} \tag{3.9}$$

where ω_i are weights that scale the deviations from the specifications. The NLP (3.9) is solved using Successive Linear Programming (SLP). The method generates a search direction from a linear program obtained from the linearization of (3.9). If the given specification is exact, the solution of the problem will correspond to a consistent set of initial conditions and a zero objective function. In case of an inconsistent overspecification, the optimum solution of (3.9) will be a positive value of the objective function, with a consistent set of initial conditions for the problem, closest to the specified set.

Note that the formulation (3.9) involves minimization of the weighted 1-norm of the violation of initial condition specifications. Hence the number of initial conditions set equal to the specified values at the solution need not be equal to the degrees of freedom for the specification of initial conditions (r) for the problem. Nonetheless it should also be noted that the objective value at the solution of the LP will be less than or equal to what it would have been if exactly r of the variables were set equal to their specified values. This is because (3.9) is not constrained to fix r initial conditions to their specified values but to find a consistent set of initial conditions closest to their specified values.

However, the user might want to have r variables to be set equal to their specified values, if the solution of (3.9) fixes less than r specifications. In other words, he may wish to pick the best set of r variables so that the specification violation of the rest of the variables is minimized. The degrees of freedom could be determined from the methods described in section 1. Once r is known, the following Mixed Integer NonLinear Programming (MINLP) formulation (3.10) could be used for consistent initialization. Here the λ_is are binary variables and M_is correspond to the big-M constraints.

$$
\begin{aligned}
& Min \sum_{i=1}^{n_{ic}} \omega_i \delta_i && \\
s.t.\quad & F^{\nu}(x, z, z', \dots, z^{(\nu)}, t) = 0 && \\
& -\delta_i \le X_i - X_i^{sp} \le \delta_i \qquad i = 1, \dots, n_{ic} && (3.10a) \\
& 0 \le \delta_i \le M_i(1 - \lambda_i) \qquad i = 1, \dots, n_{ic} && (3.10b) \\
& \sum_{i=1}^{n_{ic}} \lambda_i = r && (3.10c) \\
& \lambda_i = 0, 1 \qquad i = 1, \dots, n_{ic} && \\
& X_i^L \le X_i \le X_i^U && (3.10)
\end{aligned}
$$

A value of $\lambda_i = 1 \Rightarrow \delta_i = 0$ which implies that variable X_i is set equal to its specified value at the solution of the problem. The constraint $\sum_{i=1}^{n_{ic}} \lambda_i = r$ serves the purpose of setting at least r variables to their specified values. However, (3.10) is an MINLP and could be expensive for the solution of the above problem. We would like to simplify the above formulation to a form where it could be solved using SLP.

If we rewrite (3.10a) as

$$\delta_i + X_i - X_i^{sp} \ge 0 \qquad (3.11a_1)$$

$$\delta_i - X_i + X_i^{sp} \ge 0 \qquad (3.11a_2)$$

The following inequalities are introduced in place of (3.10b).

$$\lambda_i \left[(\delta_i + X_i - X_i^{sp}) + (\delta_i - X_i + X_i^{sp}) \right] \le 0 \qquad (3.11b_1)$$

$$0 \le \lambda_i \le 1 \qquad (3.11b_2)$$

Note that λ_i in (3.11b) are continuous variables bounded between zero and one as opposed to the binary λ_i used in (3.10b). (3.11b) could be simplified to:

$$\lambda_i \delta_i \leq 0 \tag{3.11c_1}$$

$$0 \leq \lambda_i \leq 1 \tag{3.11c_2}$$

Consider the case $\lambda_i > 0$. (3.11c_1) will be satisfied *iff* $\delta_i = 0$ because of the nonnegativity constraint on δ_i. *ie.* $X_i = X_i^{sp}$. A nonzero λ_i corresponds to the variable X_i set equal to its specified value. Similarly a nonzero δ_i ($X_i \neq X_i^{sp}$) implies $\lambda_i = 0$.

The summation equation (3.10c) is retained. (3.11c_1) and the nonnegativity conditions $\lambda_i \geq 0$ and $\delta_i \geq 0$ reduces to the complementarity condition $\lambda_i \delta_i = 0$. The simplified formulation using continuous variables is given by (3.11).

$$\begin{aligned}
& Min \sum_{i=1}^{n_{ic}} \omega_i \delta_i \\
s.t. \quad & F^{\nu}(x, z, z', \ldots, z^{(\nu)}, t) = 0 \\
& -\delta_i \leq X_i - X_i^{sp} \leq \delta_i \qquad i = 1, \ldots, n_{ic} \\
& \lambda_i \delta_i = 0 \qquad i = 1, \ldots, n_{ic} \\
& \sum_{i=1}^{n_{ic}} \lambda_i = r \\
& 0 \leq \lambda_i \leq 1 \qquad i = 1, \ldots, n_{ic} \\
& \delta_i \geq 0 \qquad i = 1, \ldots, n_{ic} \\
& X_i^L \leq X_i \leq X_i^U
\end{aligned} \tag{3.11}$$

The reformulation of the original MINLP problem to an NLP (3.11) constrained by complementarity conditions presents us with an opportunity to use a variety of techniques suggested to solve complementarity problems (Billups *et al.* [17]). Chen and Mangasarian [33] proposed a class of smoothing functions to solve the complementarity problem approximately. λ and δ satisfy the complementarity condition if and only if $\lambda = (\lambda - \delta)_+$ where $(\lambda - \delta)_+ = \max(0, \lambda - \delta)$. Here the max function is approximated using the smoothing function

$$\max(0,\eta) \approx \eta + \beta \ \log(1 + e^{-\eta/\beta}) \tag{3.12}$$

As the parameter β approaches zero the smooth approximation approaches the nonsmooth max function. Using the smooth approximation (3.12), (3.11) can be reformulated as

$$\begin{aligned}
& Min \sum_{i=1}^{n_{ic}} \omega_i \delta_i \\
s.t. \quad & F^{\nu}(x, z, z', \ldots, z^{(\nu)}, t) = 0 \\
& -\delta_i \leq X_i - X_i^{sp} \leq \delta_i \qquad i = 1, \ldots, n_{ic} \\
& \delta_i = \beta \ \log(1 + e^{-(\lambda_i - \delta_i)/\beta}) \quad i = 1, \ldots, n_{ic} \\
& \sum_{i=1}^{n_{ic}} \lambda_i = r \\
& 0 \leq \lambda_i \leq 1 \qquad i = 1, \ldots, n_{ic} \\
& \delta_i \geq 0 \qquad i = 1, \ldots, n_{ic} \\
& X_i^L \leq X_i \leq X_i^U
\end{aligned} \tag{3.13}$$

Formulation (3.13) is an NLP and is solved using SLP since a vertex solution is expected. (3.13) is less cumbersome to solve when compared to (3.10). In Appendix A, we show that if the degrees of freedom (r) are specified exactly, except for the case of consistent overspecification, a nonzero λ_i will take its upper bound value of 1, which is a solution of the MINLP formulation (3.10). Detailed treatment of the applicability of smoothing functions for solving complementarity problems is presented in chapter 6.

3.2.3 Examples

Below, we consider three example problems to illustrate the application of formulations (3.9) and (3.13).

Example 1: Chemical Reactor Problem

Consider the problem described by (3.14) (Campbell and Moore [28], Pantelides [79]).

$$\begin{aligned}
C' + C + R &= 4 + t + t^2 \\
T' + 2T + R + T_C &= 1 + e^{-t} \\
T^{-1} + ln(R/C) &= 0 \\
C &= cosh(t-1)
\end{aligned} \tag{3.14}$$

The problem is index 3 and has a unique solution. There are no degrees of freedom for specification of the initial conditions. However since there are two differential equations, the user may be tempted to specify initial conditions corresponding to the differential variables. The formulation (3.9) should be able to take care of any such inconsistent specifications and give us a consistent set of initial conditions closest to the specified ones (in this case the unique initial vector). In Table 3.1 we consider two inconsistent specifications for the initial conditions. In both cases the SLP method solved for the correct set of initial conditions with a nonzero objective function at the solution. Note that the second case corresponds to the commonly used steady state initial condition specifications. □

Example 2: Pendulum Problem

This is the popular pendulum problem considered by various authors (Pantelides [79], Gear [51] Unger *et al.* [97]). The formulation is given by (3.15).

$$\begin{aligned}
x' &= v \\
y' &= w \\
v' &= -Tx \\
w' &= -Ty - g \\
x^2 + y^2 &= 1
\end{aligned} \tag{3.15}$$

The problem is of index 3 and has 2 degrees of freedom for specification of initial conditions. The two underlying algebraic equations (3.16) are derived from the successive differentiation of (3.15).

$$\begin{aligned}
xv + yw &= 0 \\
-T(x^2 + y^2) + v^2 + w^2 - gy &= 0
\end{aligned} \tag{3.16}$$

Table 3.1: Initial Conditions for the Chemical Reactor Problem

Specified ICs	Actual ICs (SLP Solution)	Number of SLP Iterations
$C = 1$ $T_C = 0$	$C = 1.5431$ $R = 3.6321$ $T = -1.1682$ $T_C = -0.5726$	12
$C' = 0$ $T' = 0$	$C' = -1.1752$ $T' = 1.2768$	6

Table 3.2: Initial Conditions for the pendulum problem

Specified ICs	Solution from (3.9)	Number of LPs solved
$x = 0.5$ $v = 1.0$	$x = 0.5$ $v = 1.0$	11
$x = 0.5$ $T = 0.5$	$x = 0.5$ $T = 0.5$	12
$x = 0.5$ $v = 1.0$ $T = 0.5$	$x = 0.5$ $v = 1.012$ $T = 0.5$	12

Table 3.2 shows how the proposed method solves for a consistent set of initial conditions for three different specifications.

The number of specifications in the first two sets is equal to the degrees of freedom (2) and the sets (x, v) and (x, T) form a feasible pair (The Jacobian of the original set of equations with respect to all the variables other than the specified ones is nonsingular). As seen, applying SLP to (3.9) solves for the consistent initial vector with the solution for the specified variables being equal to their specified values. However, the third set is an overspecification. In this case SLP finds the initial condition vector which is closest, in the 1-norm sense, to the specified ones.

Now consider the application of (3.13) to this problem. As seen from the specifications in Table 3.3, even though there are two degrees of freedom for this problem, solution with (3.9) sets only one variable to its specified value. On the other hand, the problem when solved with (3.13) and $r = 2$, sets x and w to their specified values. The corresponding objective function is 0.3660 which is the deviation in v. This objective is greater than the value of 0.3128 from (3.9). □

Table 3.3: Solution for the pendulum problem from inconsistent specifications using (3.9) and (3.13)

Specified ICs	Solution from (3.9)	Solution from (3.13)
$x = -0.5$	$x = -0.5463$	$x = -0.5000$
	$y = 0.8376$	$y = 0.8660$
$v = 0.5$	$v = 0.7665$	$v = 0.8660$
$w = 0.5$	$w = 0.5000$	$w = 0.5000$
	$T = 0.0000$	$T = 0.1340$
	$\Sigma\delta_i = 0.3128$	$\Sigma\delta_i = 0.3660$

Example 3: Trajectory Prescribed Path Control (TPPC) Problem

Here we consider a nonlinear example used in the simulation of space vehicles when the shape of the trajectory is prescribed. TPPC problems are treated extensively in Brenan *et al.* [20]. (3.17) is an index 3 problem where the first six equations are the equations of motion in relative coordinates and the last equation is the algebraic path constraint.

$$
\begin{aligned}
H' &= V_R \sin\gamma \\
\xi' &= \frac{V_R \cos\gamma \sin A}{r\cos\lambda} \\
\lambda' &= \frac{V_R}{r}\cos\gamma\cos A \\
V_R' &= -\frac{D}{m} - g\sin\gamma - \Omega_E^2 r\cos\lambda\left(\sin\lambda\cos A\cos\gamma - \cos\lambda\sin\gamma\right) \\
\gamma' &= \frac{L\cos\beta}{mV_R} + \frac{\cos\gamma}{V_R}\left(\frac{V_R^2}{r} - g\right) + 2\Omega_E\cos\lambda\sin A + \\
&\quad \frac{\Omega_E^2 r\cos\lambda}{V_R}\left(\sin\lambda\cos A\sin\gamma + \cos\lambda\cos\gamma\right) \\
A' &= \frac{L\sin\beta}{mV_R\cos\gamma} + \frac{V_R}{r}\cos\gamma\sin A\tan\lambda - 2\Omega_E\left(\cos\lambda\cos A\tan\gamma - \sin\lambda\right) + \\
&\quad \frac{\Omega_E^2 r\cos\lambda\sin\lambda\sin A}{V_R\cos\gamma} \\
\frac{D}{m} &- \left[C_0 + C_1\left(V_R - V_0\right) + C_2\left(V_R - V_0\right)^2 + C_3\left(V_R - V_0\right)^3\right] = 0
\end{aligned}
\tag{3.17}
$$

Computations by Campbell and Moore [28] using the initial values of the state variables obtained from Brenan *et al.* [20] found that their values were not exact. Using these values as initial guesses, they generated new initial conditions using a least squares method. There are two points to be noted in this problem:

(1) Arbitrary specification of all 7 state variables will lead to problems even if they are slightly inconsistent (as observed by Campbell and Moore [28]) since the degrees of freedom is 4.

(2) A different undesirable control history of $\beta < 0$ is possible for this problem. β must be positive in the given set of initial conditions.

The initialization problem when formulated using (3.9) and initialized using the values from Brenan *et al.* [20], returned consistent initial conditions within an accuracy of 10^{-12} in just 3 iterations. These are identical to the ones presented in Table 5 of Campbell and Moore [28]. The SLP method has the added advantage of an easy imposition of a lower bound of zero on β to avoid the undesirable control history. □

In summary, when the failure of the solver is due to inconsistent specification of initial conditions by the user, it is advantageous that the user be presented with a consistent set of initial conditions which is closest to what had been specified before and is numerically consistent. The two formulations, (3.9) and (3.13), serves this purpose and give the user flexibility to use the available information to the best advantage. Campbell and Moore [28] point out that in practical applications, one would expect that some initial values are known and there are reasonable estimates for others. The method has relevance in this context too, where the weights ω_i in (3.9) and (3.13) could be chosen to help the user determine a consistent initial vector reflecting his confidence level in the initial value estimates.

3.3 Reinitialization after discontinuities

In this section, we look at the problem of reinitialization of DAEs after discontinuities, some underlying principles for this problem and the application of the minimization formulation.

3.3.1 Problem Description

The task of finding a consistent set of initial conditions for the reinitialization of a DAE is often more complicated than the *pure* initialization problem. The reason is that the reinitialization problem involves an additional exercise of determining the variables which are continuous across a discontinuity. The numerical values of the continuous variables at the point of discontinuity contribute to the specifications for the reinitialization problem.

Majer *et al.* [67] address the problem of reinitialization of DAEs after discontinuities. They specifically look at index one linear-implicit DAEs of the form (3.18), where there are discontinuous steps in the forcing function u.

$$T(v, u, t)v' = h(v, u, t) \tag{3.18}$$

u is the vector corresponding to the input variables and an algebraic transformation of the form (3.19) exists

$$\begin{aligned} T_{11}(x,y,u)x' + T_{12}(x,y,u)y' &= h_1(x,y,u) \\ 0 &= h_2(x,y,u) \end{aligned} \quad (3.19)$$

They show that the continuity assumptions of state variables can be safely assumed by default only when $T_{12} = 0$. For the case $T_{12} \neq 0$, conditions are laid out for the solvability of consistency equations.

Brüll and Pallaske [23] considered the same index one problem and derived the necessary and sufficient condition for the above system to have a *genuine initial value* for the state variable vector v after a jump in the input variable u.

However, no work has been done for determining the validity of the continuity assumptions of state variables for higher index nonlinear DAEs. In this section, we consider a class of such problems in which the discontinuity is caused by a step change in the input function u.

An assumption about this class of problems is that the index of the problem remains the same at all times. Also, if we were to look at the discrete models at any time as independent problems, the structural degrees of freedom for the specification of initial conditions remain the same. Often it is assumed that the differential variables are continuous if the composition of the differential variables appearing in the describing equations remain unchanged across the discontinuity. This is not always true as was shown by Majer *et al.* [67] and Brüll and Pallaske [23] for fully implicit index one systems. When considering problems of arbitrary index, what is missing are rigorous criteria to determine which variables are continuous and which are not, across a discontinuity caused by a step change in one or more input functions.

3.3.2 Continuity Assumptions for Problems of Arbitrary Index

Consider a DAE of the form

$$F(x', x, y, u, t) = 0 \quad (3.20)$$

where $u \in \Re^{n_u}$. The continuity assumptions hold for that subset of variables v_c, whose underlying ODEs do not depend on any of the derivatives of the function u causing the discontinuity.

Let the system be of index ν. To demonstrate the effect of discontinuities, we assume in this section that differentiation of (3.20) ν times yields locally an explicit ODE of the form (3.21), which is termed the underlying ODE for the DAE (3.20). Here v is defined as in (3.2).

$$v' = f(v, u, u', \ldots, u^{(\nu)}, t) \tag{3.21}$$

For $\delta \geq 0$, we denote $v^- = lim_{\delta \to 0}\ v(t^* - \delta)$ and $v^+ = lim_{\delta \to 0}\ v(t^* + \delta)$, where t^* is the time of discontinuity. Let v_i be a variable which has an underlying ODE of the form (3.22) where $u^{(k)}$ is the highest derivative of u explicitly present in the underlying ODE.

$$v_i' = f_i(v, u, \ldots, u^{(k)}, t) \tag{3.22}$$

Let u_δ be a smoothed step for u defined for $t^* \leq t \leq t^* + \delta$ and v_δ be the corresponding smoothed profiles of the variables. As in Majer *et al.* [67],

$$v_i^+ = lim_{\delta \to 0}\{v_i^- + \int_t^{t+\delta} f_i(v_\delta, u_\delta, ..., u_\delta^{(k)}, t) d\tau\} \tag{3.23}$$

The derivatives of u are not bounded for step changes in u. The integral will vanish only when the derivatives of u do not form part of the integral, explicitly or implicitly through the variables appearing in the underlying ODE. Thus, only those variables whose underlying ODEs could be shown to be independent of any derivative of the input function causing the discontinuity, are continuous across it.

Let the set of continuous variables be v_c. From the above analysis, it is clear that there is no guarantee that all differential variables will fall in the set v_c. Hence, the assumption of continuity of differential variables across the discontinuity, often used in dynamic simulators, is not always true.

An underlying ODE can be determined, locally, using a symbolic algorithm, for example by Gear [51]. However, the form of the ODE found depends on the algebraic manipulation one carries out while deriving it. The above analysis provides a framework for determining the variables which are continuous across a discontinuity, but from a computational standpoint the issue of uniqueness of the underlying ODE of a given DAE (3.1) is to be addressed.

Example 4
Consider the DAE described by (3.24).

$$\begin{aligned} y' + x &= u_1 \\ y &= u_2 \end{aligned} \tag{3.24}$$

Consider the following two forms of the underlying ODE, (3.25) and (3.26).

$$\begin{aligned} x' &= u_1' - u_2'' \\ y' &= u_2' \end{aligned} \tag{3.25}$$

$$\begin{aligned} x' &= u_1' - u_2'' \\ y' &= u_1 - x \end{aligned} \tag{3.26}$$

In the event of a step change in u_2, (3.25) gives a clear picture of the behavior of y, whereas from (3.26), it is not clear because of the implicit dependence on x. □

However, in practical applications, one would not prefer to explicitly determine an underlying ODE and then validate the continuity assumptions. Given all the constraining algebraic equations for a DAE and any particular underlying ODE, one could derive other forms of underlying ODEs. The derivative array equations, in principle, contain the differential and all the constraining algebraic equations for a given DAE. Hence, it would be easier to work with the derivative array equations and determine the dependencies of the differential variables on the derivatives of u rather than deriving the required form of the underlying ODE explicitly.

3.3.3 Identification of the Continuous Set

To identify the continuous set of state variables, we consider two approaches: one based on analysis of J_{F^ν} and the second based on LP sensitivity. These approaches apply specifically to linear DAEs of the form:

$$Ax + By + Cx' + Du = f(t) \tag{3.27}$$

and we develop them for this problem class. We also show, however, that they can be extended with some restrictions to nonlinear problems as well.

1. Analysis of the Jacobian of the derivative array equations
An analysis of the Jacobian of the derivative array equations of (3.27) can be used to determine the set of variables for which the continuity assumption is valid. In the Jacobian of the derivative array equations, we note that the higher derivatives of u are treated as independent variables.

Now consider the example where a Gauss-Jordan elimination yields the following sub-matrix (3.28) within the Jacobian, J_{F^ν}, where ξ_1, ξ_2, ξ_3,... are the incidences in the columns corresponding to the derivatives of u.

$$\overline{J}_{F^\nu} = \begin{array}{c} \begin{array}{ccccccccccc} & & & v' & & & u' & u'' & . & . & u^{(\nu)} \end{array} \\ \left[\begin{array}{cccccc|ccccc} 1 & & & & & & & & & & \xi_1 \\ & 1 & & & & & \xi_2 & & & & \\ & & . & & & & & & & & \\ & & & . & & & & & & & \\ & & & & 1 & & & & & & \\ & & & & & 1 & & \xi_3 & & & \end{array}\right] \end{array} \tag{3.28}$$

The continuous set of state variables, v_c, could be identified from (3.28). The state variables corresponding to the rows which do not contain incidences of u', ..., $u^{(\nu)}$ are continuous across the discontinuity caused by the forcing function u. However we need to safeguard against implicit dependencies. This could be done by choosing an appropriate pivoting strategy while doing symbolic elimination on the Jacobian of the derivative array equations. A verification of this can be carried out from the structure of the Jacobian after performing the elimination. If a particular variable v_k is found to be continuous, we should not be able to trace a path from an incidence in the column under v_k' to an incidence in the derivatives of u. That is, the set of reachable nodes from v_k' should not contain any of the derivatives of u, if v_k is continuous. If such a path is found, further pivoting can give us the desired structure. Pivoting is straightforward in the case of linear systems. Below we illustrate how this approach solves the example of (3.24).

We could determine the implicit dependencies from a form like (3.26) by path tracing on the corresponding Jacobian of the derivative array equations (3.29).

$$\begin{aligned} y' + x &= u_1 \\ y'' + x' &= u_1' \\ y &= u_2 \\ y' &= u_2' \\ y'' &= u_2'' \end{aligned} \tag{3.29}$$

Consider the Jacobian (3.30) of (3.29).

$$\begin{array}{c} \begin{array}{cccccccccc} y'' & y' & x' & y & x & u_1 & u_1' & u_2 & u_2' & u_2'' \end{array} \\ \left[\begin{array}{c|cc|cc|cc|ccc} & 1 & & & 1 & -1 & & & & \\ 1 & & 1 & & & & -1 & & & \\ & & & 1 & & & & -1 & & \\ & 1 & & & & & & & -1 & \\ 1 & & & & & & & & & -1 \end{array}\right] \end{array} \tag{3.30}$$

Manipulations on the Jacobian (3.30) will give the following incidence matrix for the form (3.26).

$$\begin{array}{c}
\begin{array}{c|cc|cc|cc|ccc} y'' & y' & x' & y & x & u_1 & u_1' & u_2 & u_2' & u_2'' \end{array} \\
\left[\begin{array}{c|cc|cc|cc|ccc}
 & 1 & & & 1 & -1 & & & & \\
 & & 1 & & & & -1 & & & 1 \\
 & & & 1 & & & & -1 & & \\
 & & & & -1 & 1 & & & -1 & \\
1 & & & & & & & & & -1
\end{array}\right]
\end{array} \tag{3.31}$$

We can trace a path from an incidence in the column y' to that in u_2' showing that there is dependence on u_2' which is implicit in (3.26). Further pivoting in the column under x will give the desired form (3.32) which corresponds to (3.25).

$$\begin{array}{c}
\begin{array}{c|cc|cc|cc|ccc} y'' & y' & x' & y & x & u_1 & u_1' & u_2 & u_2' & u_2'' \end{array} \\
\left[\begin{array}{c|cc|cc|cc|ccc}
 & 1 & & & & & & & -1 & \\
 & & 1 & & & & -1 & & & 1 \\
 & & & 1 & & & & -1 & & \\
 & & & & -1 & 1 & & & -1 & \\
1 & & & & & & & & & -1
\end{array}\right]
\end{array} \tag{3.32}$$

2. Sensitivity analysis of the derivative array equations

An alternate approach suggested to find the dependencies of the variable derivatives on the derivatives of the input function is to do a sensitivity analysis on the derivative array equations. Caracotsios and Stewart [29] and Maly and Petzold [68] describe sensitivity analysis for DAEs. However, since we are not interested in finding the sensitivity of the entire profile, but just on the dependencies at the point of discontinuity, we consider the following SLP formulation:

$$\begin{aligned}
& Min \sum_{j=1}^{n} (p_j + n_j) \\
s.t. \quad & F_j^{\nu}(x^{(\nu+1)}, ...x, y^{(\nu)}, ..., y, u^{(\nu)}, ...u, t) = p_j - n_j \quad j = 1, \dots, (\nu+1)n \\
& u^{(k)} = u^{(k)-} \qquad k = 0, \dots, \nu
\end{aligned} \tag{3.33}$$

where $u^{(k)-} = lim_{\delta \to 0}\ u^{(k)}(t^* - \delta)$. The numerical value of the variables at the point of discontinuity is the optimal solution to (3.33). Sensitivity analysis is carried out with u and its higher derivatives being treated as independent variables and by linearizing (3.33) at the variables values before the discontinuity. If a particular derivative x_j' is seen to be insensitive to the parameters u',.....,$u^{(\nu+1)}$, it is because it has no local dependence on any of the derivatives of the input variables *ie.* it is continuous across a discontinuity caused by a step change in u. However, one should be aware that the test is local and the analysis provides a framework which is necessary but not sufficient for determining the continuities. The sensitivity analysis is performed using LP sensitivity, which is computationally cheap.

Nonlinear DAEs

The current approach we use for determining the dependencies for nonlinear DAEs is the sensitivity analysis described above. Since (3.33) is solved using SLP, it could be used for nonlinear systems as well. The sensitivity analysis would effectively be performed on the linearized form of the equations at $(t^* - \delta)$. The method has been applied to the example problems solved in section 3.5.

Application of pivoting strategies on the Jacobian of the derivative array equations for the nonlinear case is limited as one can only determine, in graph theory notation, a *maximal* set of reachable nodes. If this maximal set does not contain any nodes corresponding to the implicit dependencies, the sets determined thus far are correct. Hence, analysis of the Jacobian may not give the desired result for nonlinear DAEs.

3.3.4 Applications of the LP Method for Reinitialization

A certain number of specifications for the initial conditions is required for restarting the problem after the discontinuity. Some of these are obtained from the criterion laid out for the validity of continuity assumptions of the variables across the discontinuity. However, the number of variables which are continuous across the discontinuity could be less than, equal to or greater than the degrees of freedom for the specification of initial conditions. Let us analyze the three cases and look at how the problem could be tackled in each of these cases.

1. Number of variables continuous across the discontinuity is greater than the number of degrees of freedom.

Here the degrees of freedom are fewer than the number of specifications for the variables which are found to be continuous across the discontinuity. This is an overspecified problem. It is quite likely that if we fix the values of these variables and solve the equations for consistent initialization, we might not be able to converge it within the required error criterion. One of the reasons for this is the roundoff errors arising from integration in the previous time domain. It is well known that if the initial conditions are not sufficiently accurate, further solution of the problem could be difficult and inaccurate. For this reason we would like to determine the initial conditions such that the equations are solved within a desired error criterion. The following formulation, which is similar to (3.9), is introduced for this purpose.

$$\begin{aligned}
& Min \sum_{i \in c} \omega_i \delta_i \\
s.t. \quad & F^{\nu}(x^{(\nu+1)}, ...x, y^{(\nu)}, ..., y, u^{(\nu)}, ...u, t) = 0 \\
& -\delta_i \leq v_i - v_i^{disc} \leq \delta_i \qquad i \in c \\
& \delta_i \geq 0 \qquad i \in c
\end{aligned} \tag{3.34}$$

v_c is the set of variables which are found to be continuous across the discontinuity, c is its index set and v_i^{disc} are their specified numerical values at the point of discontinuity.

2. The number of continuous variables is equal to the degrees of freedom.

This is the easiest case where we have the exact number of specifications. (3.34) could be applied to determine the values of the remaining variables and the solution will correspond to a zero objective function.

3. The number of continuity conditions is less than the degrees of freedom.

This is an underspecified problem and if we solve (3.34) we will obtain a nonunique solution vector. However, in engineering problems there are often additional discontinuity constraints which are known along with the step change in the input functions. For example, correlations can be added for computing impulse forces in the case of collision of nonpenetrating rigid bodies. If we let $f_i^d(v) = 0$ be the discontinuity relations for the-profiles and we assume that these discontinuity constraints provide at least the minimum required information to cover the remaining degrees of freedom, the following formulation could be used for consistent reinitialization.

$$
\begin{aligned}
& Min \quad \sum_{i \in c} \omega_i^1 \delta_i^c + \sum_{i=1}^{n_f} \omega_i^2 \delta_i^f \\
s.t. \quad & F^\nu(x^{(\nu+1)}, ...x, y^{(\nu)}, ..., y, u^{(\nu)}, ...u, t) = 0 \\
& -\delta_i^c \leq v_i - v_i^{disc} \leq \delta_i^c \qquad i \in c \\
& -\delta_i^f \leq f_i^d \leq \delta_i^f \qquad i = 1, ...n_f \\
& \delta_i^c, \delta_i^f \geq 0
\end{aligned}
\tag{3.35}
$$

The same formulation could be applied to cases 1 and 2 as well. One of the major difficulties associated with reinitialization problems is to determine whether a particular discontinuity presents an under, exact or over specified problem and then the associated clumsy process of the selection of that subset of equations which has to be solved. (3.35) presents a single consistent formulation to handle all of these cases.

One question still unanswered is the presence of the input function u and its higher derivatives in the derivative array equations used in (3.35). While state profiles cannot be calculated that capture the exact discontinuities, profiles can be determined to reinitialize the problem in the next time domain. Here we let

$$
u = \begin{cases} f_a(t) & t < t^* \\ f_b(t) & t > t^* \end{cases} \tag{3.36}
$$

Using the Heaviside function (3.37), u is written as (3.38)

$$Y(t^*) = \begin{cases} 0 & t < t^* \\ 1 & t > t^* \end{cases} \tag{3.37}$$

$$u(t) = f_a(t)(1 - Y(t^*)) + f_b(t)(Y(t^*)) \tag{3.38}$$

ith derivative of (3.38) could be written as

$$u^{(i)}(t) = \sum_{j=0}^{i} \begin{pmatrix} i \\ j \end{pmatrix} Y^{(j)}(t^*)(f_b - f_a)^{(i-j)} + f_a^{(i)} \tag{3.39}$$

From the theory of distributions (Barros-Neto [9]), the first derivative of the Heaviside function corresponds to the Dirac delta function and its higher derivatives in (3.39) are the higher derivatives of this function. These functions are defined only at t^*.

Our problem is to find initial conditions for the derivatives of u for the solution of the problem in the time domain *after* the discontinuity. This is different from tracing the solution from one side of the discontinuity to the other where the derivatives of u are unbounded at the point of discontinuity. Here the idea is to find the initial conditions for the derivatives of u which will enable us to correctly trace the solution trajectory *after* the discontinuity. This is obtained by finding the limit of (3.39) when $t \to t^*$ from the right hand side. The limit is given by

$$\begin{aligned} \lim_{\delta \to 0^+} u^{(i)}(t^* + \delta) &= \lim_{\delta \to 0^+} Y(t^*)(f_b - f_a)^{(i)}(t^* + \delta) + f_a^{(i)}(t^* + \delta) \\ &= f_b^{(i)}(t^*) \end{aligned} \tag{3.40}$$

All the terms containing the higher derivatives of Y vanish as they are zero at any point $(t^* + \delta)$ however small δ may be. Thus for the reinitialization problem, the higher derivatives of u present in the derivative array equations take the form of the higher derivatives of their functional forms defined in the region after the discontinuity.

Let us consider a simple example to illustrate the point. Consider the index three DAE of the form

$$\begin{aligned} x_1' &= y - x_1 \\ x_2' &= x_1 - x_2 \\ x_2 &= u(t) \end{aligned} \tag{3.41}$$

where

$$u(t) = \begin{cases} 0 & t < t^* \\ 1 & t > t^* \end{cases}$$

There are no degrees of freedom for the initial conditions of this problem and the constraining algebraic equations are

$$\begin{aligned} x_2 &= u(t) \\ x_1 &= u'(t) + u(t) \\ y &= u''(t) + 2u'(t) + u(t) \end{aligned} \tag{3.42}$$

The profiles can be plotted as in Figure 3.1.

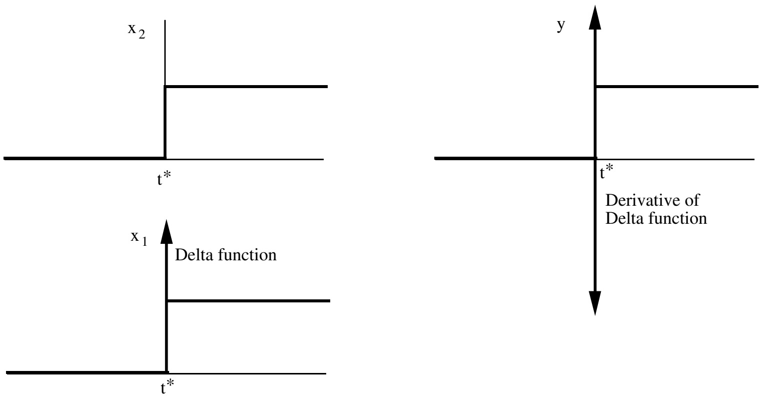

Figure 3.1: Variable profiles

As seen from Figure 3.1, x_1 and y are described by unbounded functions at t^*. Our problem is not to solve the problem at t^*, but to solve for the initial conditions of x_1, x_2 and y for the further solution of the problem after the discontinuity. Here $u = 1$ (and hence

$u' = 0$ and $u'' = 0$) will give us values of $x_1 = x_2 = y = 1$, which are the correct initial conditions for the further solution of the problem in the region after the discontinuity.

Below we consider some example problems to illustrate the ideas described in this section.

3.3.5 Examples

We consider four examples, two where the continuity assumption for the differential variables is valid and the other two where it is not. The first three are nonlinear.

Example 5

This example looks at an implicit index-1 problem (3.43) considered by Brüll and Pallaske [23]. This is a case where the continuity assumptions are not necessarily valid after the discontinuity.

$$\begin{aligned} x_1' + y_1 x_2' &= 0 \\ x_2 &= u_1(t) \\ y_1 &= u_2(t) \end{aligned} \tag{3.43}$$

After making reductions and simplifications, it is clear that the underlying ODE for (3.43) is

$$\begin{aligned} x_1' &= -u_2(t)u_1'(t) \\ x_2' &= u_1'(t) \\ y_1' &= u_2'(t) \end{aligned} \tag{3.44}$$

As seen from (3.44), none of the variables are continuous after a step change in the input functions u_1 and u_2. However, if u_1 is the only function causing the discontinuity, y_1 is found to be continuous across the discontinuity. Similarly, x_1 and x_2 are continuous across a discontinuity caused by a step change in the input function u_2. $\square$

Example 6: Pendulum problem revisited

Consider the pendulum problem from (3.15) in a slightly different context. Let the pendulum be subject to time dependent input forces with the g in (3.15) replaced by an input function u. Below, we examine the continuity of the variables when there is a step change in the forcing function u. The underlying ODE is

$$
\begin{aligned}
x' &= v \\
y' &= w \\
v' &= (yu - v^2 - w^2)x \\
w' &= (yu - v^2 - w^2)y - u \\
T' &= 3wu + yu'
\end{aligned}
\tag{3.45}
$$

From (3.45), it is clear that T' depends on the first derivative of u whereas the derivatives of the other variables depend only on u.

$$
v_c = \{x, y, v, w\} \tag{3.46}
$$

The position and velocity variables are continuous but the string tension exhibits a step change in the event of a step change in the forcing function. A similar result is obtained from the sensitivity analysis (3.33). □

Example 7

Consider the DAE (3.47) for which the continuity assumption is valid. This problem was considered previously by Majer *et al.* [67].

$$
\begin{aligned}
x' &= y_1 - 1 \\
0 &= x - y_2 + y_1^2 + u \\
0 &= 2y_1^{1/3} + y_2^{1/2} - 4
\end{aligned}
\tag{3.47}
$$

Consider the solution of the reinitialization problem of (3.47). The problem has a steady state solution of $y_1 = 1$, $y_2 = 4$ and $x = 3$ for $u = 0$. At time $t = t^*$, a step change is applied to u from 0 to 5.

From the sensitivity analysis, we find that x is continuous across the discontinuity caused by a step change in the input variable u. Setting $v_c = \{x\}$, the initialization problem was

Table 3.4: Reinitialization of Example 7

<table>
<tr><th>Initial Guesses</th><th>Solution</th><th>Number of iterations</th></tr>
<tr><td>$y_1 = 1$
$y_2 = 4$
$x = 3$</td><td rowspan="2">$x = 3.0000$
$y_1 = 0.1975$
$y_2 = 8.0390$
$x' = -0.8025$</td><td>10</td></tr>
<tr><td>$y_1 = 0$
$y_2 = 0$
$x' = 0$</td><td>12</td></tr>
</table>

solved using (3.35). Table 3.4 reports the solution of the problem for two sets of initial guesses.

The first set of initial guesses correspond to the values of the variables immediately before a step change in u was applied. This is probably the most logical initial guess to be chosen when carrying out dynamic simulation of the above system.

The second set is the initial guess used by Majer *et al.* [67] for the solution of the problem. They point out that (3.47) cannot be solved by Newton's method from this set of initial guesses because of the limited region of convergence. They use a more expensive continuation method for the solution of the problem. However, it should be noted that the LP based formulation (3.35) encountered no difficulty in solving the problem.

Apart from being an efficient way for dealing with difficult nonlinear reinitialization problems, the method has been successfully applied to high index problems as well. Below we consider a high index example problem.

Example 8: The CSTR problem

Bachmann *et al.* [5] describe a problem of N Continuous Stirred Tank Reactors (CSTRs) in series without reaction which constitutes a system of index $N+1$. The equations are

$$\begin{aligned} c_i' &= c_{i-1} - c_i \quad i = 1, ..., N \\ c_N &= u(t) \end{aligned} \tag{3.48}$$

Consider the case of 10 CSTRs. The problem is of index 11. Define u as

$$u(t) = \begin{cases} 10+t & t < 10 \\ t & t > 10 \end{cases} \tag{3.49}$$

First the continuity assumptions were checked from the analysis of the Jacobian of the derivative array equations and from sensitivity analysis (3.33). It was found that none of the variables are continuous across a discontinuity caused by a step change in u. The derivative array problem was then solved using the formulation (3.35). No variables were found to satisfy the continuity assumptions and there are no explicitly specified discontinuity constraints. Note that this problem is exactly specified as the degrees of freedom and the number of continuous variables is zero.

The derivative array problem, which has 164 variables, 132 equalities and 22 inequalities was solved in 2 iterations. It was initialized at the solution of the problem before the discontinuity. The initial value for any c_i was found to be

$$c_i^{initial} = 20 - i \qquad i = 0, ..., 10 \tag{3.50}$$

3.4 Conclusions

Overspecification of initial conditions, which occurs mainly due to the lack of knowledge of the degrees of freedom for the specification of initial conditions, could lead to convergence failures in the initialization problem. Most existing methods expect the user to supply the exact number of initial conditions. Instead a minimization formulation is presented for initialization as well as for reintialization in the case of problems with discontinuity, which takes care of the overspecifications. This formulation is solved using Successive Linear Programming (SLP) and can readily be extended to large, sparse systems of equations. Solving this minimization problem ensures that the initial conditions are determined within a required accuracy; failure to do so may result in convergence to poor or inaccurate solutions on further integration.

In section 3.3 we presented a criterion for checking the validity of the assumption of continuity of differential variables across a discontinuity caused by a step change in the input variables. Such criteria have not been reported in the literature for problems of general index. The popular assumption of continuity of all differential variables across a discontinuity (Pantelides and Barton [81]) is shown to be not necessarily valid. The continuous set of variables is determined by performing an LP sensitivity analysis on the derivative array equations.

The applications of the ideas outlined in these sections have been demonstrated on small problems, of low and high index. The concepts apply irrespective of the size and index of the problem and application to larger problems is expected to be straightforward. Also, the proposed SLP based approach is easily interfaceable to DAE solvers like DASSL. This will make the treatment of the initialization and reinitialization problems easier and more user-friendly within these solvers.

Having described an SLP based method for the initialization and reinitialization of process models described by DAEs, in chapter 4, we will outline how SLP methods can aid in the treatment of other aspects of discontuities encountered in nonsmooth dynamic simulation problems.

Chapter 4

Nonsmooth Dynamic Simulation With Applications in Phase Equilibrium

I came like Water, and like Wind I go
-Omar Khayyám, Rubáiyát [63]

Synopsis

Process simulation has emerged as a valuable tool for process design, analysis and operation. In this chapter, we extend the capabilities of successive linear programming (SLP) for dealing with problems encountered in nonsmooth dynamic process simulation. The SLP method has the advantage of naturally dealing with profile bounds as well. This is demonstrated to avoid the computational difficulties which arise from the iterates going into physically unrealistic regions. A new method for the treatment of discontinuities occurring in dynamic simulation problems is also presented in this chapter. The method ensures that any event which has occurred within the time interval in consideration is detected and if more than one event occurs, the detected one is indeed the earliest one. A specific class of implicitly discontinuous process simulation problems, phase equilibrium calculations, is also examined. A new formulation is introduced to solve multiphase problems.

4.1 Introduction

In section 1.1.1 we described the emergence of process simulation as a valuable tool for plant design, analysis and operation and some of the challenges in this area. Steady state simulations are performed routinely in chemical process industries today. Currently, development of efficient dynamic simulators is of great interest. Efficient modeling tools and solution techniques have contributed to the increasing prominence of equation oriented approaches in both steady state and dynamic simulations. Equation oriented simulators rely on a powerful equation solving engine which must be able to handle poor starting points, inequality constraints, singularities and ill-conditioning, among others. Process simulators also need to deal with variables which are restricted within certain bounds, to maintain physical and mathematical feasibility. An example would be molfractions bounded between 0 and 1. Also, dynamic simulation of chemical processes needs treatment of nonsmooth models due to the presence of discrete actions and logical constraints encountered in the simulation exercise. Hence, to address the above issues, a systematic approach is required for the solution of dynamic process simulation problems.

We extend the SLP approach developed in chapter 2 to address these issues occurring in process simulation problems. Section 4.2 focuses on the need for an efficient treatment of variable bounds and how the LP method provides a framework for the purpose. A new algorithm for the treatment of discontinuities occurring in dynamic simulation problems is presented in section 4.3. Section 4.4 describes the treatment of implicit discontinuities by looking at a specific class of problems: phase equilibrium calculations. Section 4.5 summarises the ideas outlined in this chapter.

4.2 Treatment of Profile Bounds

Converting the equation solving problem to an optimization framework allows for inequalities to be added easily. This allows the user to incorporate physically meaningful conditional relations as inequalities that limit the operation of the process under consideration, in a natural fashion. These could be thermodynamic constraints governing the behavior of the system or even simple physical constraints (For example, *sum(component* *holdups)* $\leq$ *volume of the vessel*). Bounds on functions can thus be easily added into this framework.

Chemical process simulation problems often contain variables like temperature, pressure and mole fraction, which are restricted to be within a certain region. Process models are often formulated with the assumption that the variables lie within their specified bounds. It is quite likely that process models show egregious behavior if these variables are allowed to venture out of their specified region. Hence it is important to avoid regions of *physical* and *mathematical* infeasibility in process simulation problems. Most of the conventional nonlinear equation solving techniques do not have a systematic and efficient technique for enforcing bounds on variables. The LP based method presents a very natural way of dealing with variable bounds. The LP method limits the iterates within the bounds on variables and avoids the mathematical problems which arise otherwise.

In the following example we show how enforcement of variable bounds in an LP method helps to solve the problem by restricting the iterates from going into regions of mathematical infeasibility.

4.2.1 Example

Example 1: Dow case study problem

This example is based on a parameter estimation problem originally formulated by the Dow Chemical Company (Blau *et al*. [18]).

$$\frac{dy_1}{dt} = -k_2 y_8 y_2 \tag{4.1}$$

$$\frac{dy_2}{dt} = -k_1 y_6 y_2 + k_3 y_{10} - k_2 y_8 y_2 \tag{4.2}$$

$$\frac{dy_3}{dt} = -k_2 y_8 y_2 + k_1 y_6 y_4 - 0.5 k_3 y_9 \tag{4.3}$$

$$\frac{dy_4}{dt} = -k_1 y_6 y_4 + 0.5 k_3 y_9 \tag{4.4}$$

$$\frac{dy_5}{dt} = -k_1 y_6 y_2 - k_3 y_{10} \tag{4.5}$$

$$\frac{dy_6}{dt} = -k_1 (y_6 y_2 + y_6 y_4) + k_3 (y_{10} + 0.5 y_9) \tag{4.6}$$

$$y_7 = -K_4 + y_6 + y_8 + y_9 + y_{10} \tag{4.7}$$

$$y_8 = \frac{K_2 y_1}{K_2 + y_7} \tag{4.8}$$

$$y_9 = \frac{K_3 y_3}{K_3 + y_7} \tag{4.9}$$

$$y_{10} = \frac{K_1 y_5}{K_1 + y_7} \tag{4.10}$$

$$y_{11} = K_6 ln(y_7 + K_5) \tag{4.11}$$

In the model described by (4.1-4.11), y's represent the state variables, and K's and k's are parameters as reported by R. H. Farris (Biegler *et al.* [13]). Values of the parameters are given by (4.12).

$$\begin{aligned} k_1 &= 1.8192 \\ k_2 &= 2.8595 \\ k_3 &= 2926 \\ K_1 &= 2.575 \times 10^{-16} \\ K_2 &= 4.876 \times 10^{-14} \\ K_3 &= 1.7884 \times 10^{-16} \\ K_4 &= 1.31 \times 10^{-2} \\ K_5 &= 1 \times 10^{-10} \\ K_6 &= 1 \times 10^{-3} \end{aligned} \tag{4.12}$$

The differential equation was descretized using implicit Euler and the resulted system was solved using SLP. The problem was attempted under two conditions: one with the specification of bounds and the other without. In the unbounded case, the solution of the first LP gave a search direction which led to a negative value of y_7. (4.11) could not be evaluated at that point and the method failed. However, in the bounded case, the variables y_1 to y_{10} were bounded below by zero as they represent concentrations in the original problem. The values of y_7 obtained from solving the dynamic problem with the incorporation of bounds are plotted in Figure 4.1. y_7 is very close to zero and with the imposition of bounds, the LP method never allows the iterates to go beyond them, which caused the numerical difficulty in the previous case. Solution of this problem required 1163 LP iterations for 494 time steps.

Another important aspect of the treatment of bounds is in step- size control. Although the DAE solution profile lies within the specified bounds, the discretized equations could have a solution which lies outside the variable bounds. The methods which cannot handle the treatment of bounds might accept this solution as the true solution and proceed with the integration. The LP method will terminate with a pseudo-solution (a solution at the bounds) with the current step-size when there is no solution within the specified bounds, and the problem is then resolved with a smaller step size for more accurate results.

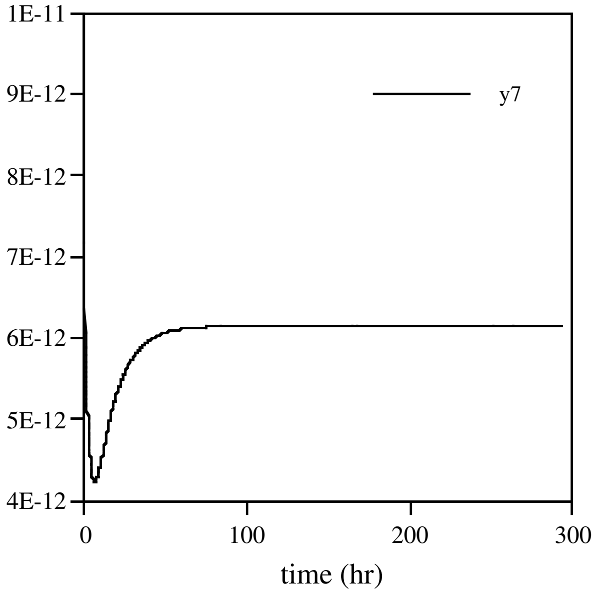

Figure 4.1: Solution of example 1 with the incorporation of bounds.

4.3 Explicit Discontinuities

Very few industrial processes operate in a *truly* continuous manner. Nonsmoothness in modeling equations occur as a result of discrete actions and logical conditions in the dynamic simulation of these processes. In this section we look at various aspects of this problem.

As described in section 1.2, three important issues are to be considered while dealing with nonsmooth dynamic simulation problems:

1. Detection of (state) events which occur within a specified time interval.
2. Exact location of the earliest event within that interval.
3. Reinitialization for restarting the integration following a discontinuity.

In chapter 3, we discussed the solution of the reinitalization problem. In the remainder of this section, we see that SLP-based methods can aid in event detection and resolution.

4.3.1 Existing Methods

According to Marquardt [69], the state of the art in dynamic simulation with discontinuities are the so-called discontinuity locking and switching functions (Joglekar and Reklaitis [60], Pantelides [80], Smith and Morton [90]). When integration is carried out, the model is locked in one continuous case and no discontinuity is allowed. This avoids problems which might occur while integrating across discontinuities, either side of which are described by different models. The switching functions are designed in such a fashion that their zeros correspond to the points of discontinuity. The switching functions will show a sign change when a switch in model takes place during an integration step. Once an event is detected, the occurrence time is located by some interpolation mechanism. The integration proceeds from that time onwards with the newly activated model. Even though this is the general outline, researchers have adopted different ways of defining their switching functions and using them to detect the time of occurrence of events.

Carver [30] described a method for detecting and handling discontinuities in functions which form part of an ordinary differential equation (ODE) set. He differentiated the discontinuity functions instead of treating them as algebraic equations and appended them to the original set of ODEs. The resulting set of ODEs is then integrated using Gear's method [49] with Nordsieck step size control [78]. This procedure maintains a Nordsieck polynomial of order q ($q < 5$) for all the variables including the ones describing the switching functions. State events are detected from the signs of the switching functions and their first derivatives at the boundaries of the integration interval. If any are present, the time at which the discontinuity occurs is determined by solving the polynomial approximation of the switching functions within that interval.

In their description of BOSS, a simulator for batch and semi-continuous processes, Joglekar and Reklaitis [60] did not append the switching function to the original set of DAEs describing the process. The q-th order polynomial approximation for each variable which is obtained from the solution of the DAE is used to construct the polynomial for the switching functions. As described before the signs of the functions are used to detect and the polynomial approximation is then solved to locate any event. This method however suffers from loss of accuracy when the switching functions are nonlinear.

Pantelides [80] does not use discontinuity functions in SpeedUp. The logical conditions are checked at the end of each integration step. If they are triggered at the end of the step, a bisection method is used to locate the state event time. Park and Barton [82] proposed a two-phase strategy for the detection and location of state events. They argue that state event location requires simultaneous determination of a state event time and initial conditions for the DAE in the next time interval. The algorithm detects in the first phase, the switching function whose zero crossing determines the earliest state event. The polynomial approximations are obtained directly from the BDF method (Gear [50]) used to solve the DAEs. The second phase carries out the simultaneous determination of the state event location and reinitialization. The above representative algorithms in literature have their own 'pro's and 'con's. Below, we look at those with respect to the some of the important issues in this problem.

Detection

The first step in the treatment of discontinuities is their 'detection'. A broad spectrum of algorithms base the detection of discontinuities on the signs of the switching functions and their derivatives at the boundaries of the interval (example Joglekar and Reklaitis [60]). Such a procedure can fail to detect multiple zero crossings of the switching function within that interval. The same applies to methods which check for the logical condition at the end of each integration step (Pantelides [80]). For example consider the pressure profile in a pressure vessel system given in figure 4.2. If the condition triggering the opening of a valve was $P > \overline{P}$ the aforementioned detection procedures would not have detected that P had ever gone beyond $\overline{P}$ by verifying the signs of the switching function $P - \overline{P}$ at t_k and t_{k+1}. However, this is not the only problem which is encountered in the detection phase. When multiple zero crossings of the switching function occurs, a wrong state event could be detected as occurring first. Consider a bisection method (Pantelides [80]) applied to the profile in 4.3. A discontinuity is detected upon checking the signs of the switching function at t_k and t_{k+1}. The method then checks the sign of the function at $\bar{t}$. Since the signs of the switching function are the same at t_k and $\bar{t}$, the method will detect point 3 as the offending discontinuity whereas it is 1 in reality.

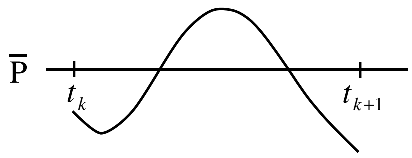

Figure 4.2: A sign checking procedure fails to detect an even number of zero crossings

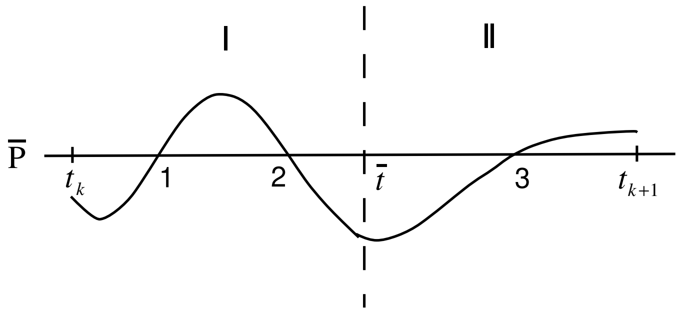

Figure 4.3: A bisection method detects the wrong discontinuity

An alternate method suggested to overcome this difficulty is to construct a polynomial approximation of the switching function and to solve for all the roots of this function within an interval. The earliest of such roots determined is identified as the offending discontinuity. Most of these methods have been implemented in conjunction with the polynomial interpolation formulae generated by the Backward Difference Formula (BDF) methods employed to solve the DAEs themselves. Park and Barton [82] describes a Gear's method with Nordsieck step size control. Roots of the switching function(s) within an interval are detected using an elegant method based on interval analysis. This method is promising, however there are some issues which need to be considered for further improvement of this class of techniques.

(a) Accuracy of the polynomial approximation. The success of the method depends on how accurately the polynomial approximation describes the behavior of the switching function. Consider the following example.

$$x_1' = 10, \qquad x_1(0) = 0 \tag{4.13}$$

The switching functions are defined by

$$z_1 = \cos x_1 - 1.2 \tag{4.14}$$

$$z_2 = x_1 - 5.2 \tag{4.15}$$

It is obvious that z_1 will never be satisfied and z_2 will be triggered off at t=0.52. The 5th order polynomial approximation (analytical) for cos(10t) is shown in figure 4.4. For a constant step size implementation, if the step size was large, the above method would have ended up locating the wrong discontinuity which would never have been triggered in reality.

For a variable step size method, the problem could be tackled in two ways. One way of doing this is by appending the switching functions to the DAEs and solving the augmented system of equations. The other option is treating them independently and analyzing them from the solution of the original set of DAEs. The latter one is a bad option if the switching functions exhibit faster transients when compared to the DAEs describing the behavior of the system. The error control mechanism of the integration procedure might choose larger steps since it deals only with the original set of DAEs. This will result in an inaccurate polynomial approximation for the switching function. The example in (4.13-4.15) and figure 4.4 is one such problem where the DAE is well behaved but the switching function is exhibiting faster transients. In view of the above, appending the switching functions to the DAEs is definitely a better option. However, the integration of the augmented system may require more integration steps as the functional characteristics of the switching functions may limit the step size [82]. However, the following needs to

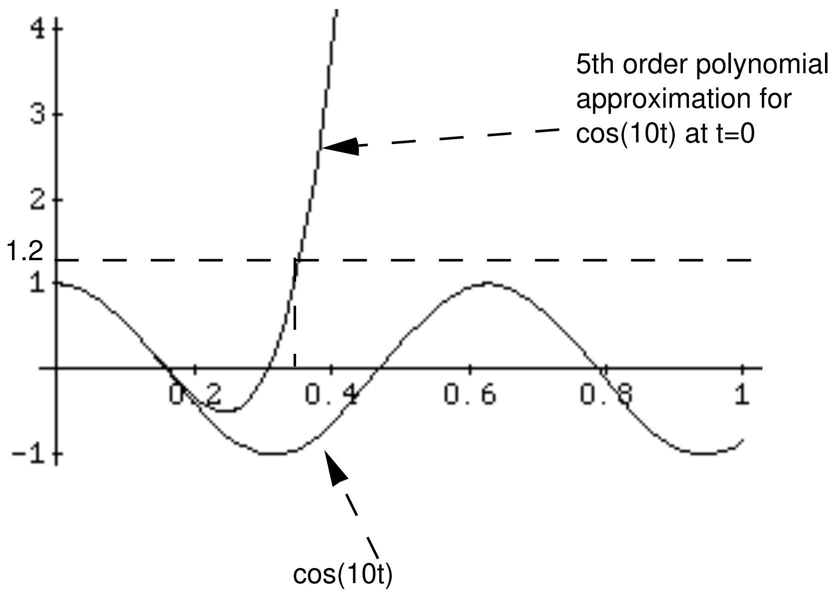

Figure 4.4: A function and its fifth order polynomial approximation

be considered in such an implementation.

(b) Discontinuities encountered one after the other

In some problems it is likely that discontinuities are encountered one after the other in very short time intervals. In such a scenario, the problem is reinitialized over and over again. Brenan *et al.* [20] notes that one of the major disadvantages of multistep methods for applications to problems with discontinuities is that they need to be restarted, usually at low order, after every discontinuity. The lower order polynomial approximations of the switching functions may no longer be accurate enough to determine the exact time of occurence of a discontinuity, for problems encountering frequent discontinuities.

(c) Isolation of the earliest event

[82] utilizes the polynomial approximation of the discontinuity functions to determine the earliest event and isolates that particular event for the determination of the state event time. The question which arises in this context is whether one can isolate an event as occurring first with absolute certainty or not. Gear's method does not guarantee consistency of the differential and algebraic variables between the mesh points. Hence it is possible

that in the event of more than one logical condition being triggered within short intervals, the wrong one may be detected and solved for in the subsequent phase. This could occur due to a variety of reasons. One of them is the difficulties with a BDF implementation for maintaining an accurate polynomial approximation in the event of rapid changes in the interval size as described in the previous section. Also while the integration is starting up or when there is step discontinuity in the variables, the resulting lower degree polynomials do not describe the exact behavior of the system. This advocates the need for an iterative solving procedure rather than relying only upon the polynomial approximation for the detection of an event.

Location and Reinitialization

The two other issues associated with the treatment of discontinuities are the location of the discontinuity and reinitialization for restarting the integration in the next interval. Broadly, one could classify the existing methods under two heads based on their treatment of these issues: (a) algorithms which perform reinitialization independent of the event detection and location and (b) algorithms which couple the reinitialization step with the location phase.

Almost all of the present methods perform the reinitialization step independently after an event has been located. [82] identified a phenomena called discontinuity sticking (figure 4.5) which occur with these methods. In most of the current methods, reinitialization is based on the assumption of continuity of the differential variables across the discontinuity, which we have showed in chapter 2 as not necessarily true. Given the values of the differential variables the new set of DAEs is solved to find the initial values for the other variables. If the state event time has not been located accurately (this will occur if the polynomial is not accurate enough), the converged initialization calculation may show that the event detected has actually not been triggered. The event may again be detected immediately thereafter. To overcome this problem the same authors suggested that the location and the reinitialization steps be performed simultaneously. The consistent state event location and reinitialization step simultaneously solves the new set of DAEs, the polynomial approximation for the differential variables and the offending switching function for the state event time and the continuous variables. This ensures that the switching function maintains the same sign even after reinitialization. The assumption made here is that accurate polynomial approximations to the differential variables are available. We have showed in chapter 2, that, a large number of discontinuities encountered in chemical process simulation could correspond to a discontiuity in the differential as well as the continuous variables. For example, when an analog control loop involving integral action is switched to automatic control, the integral of the controller error is normally initialized to zero causing a step change. Other examples include dumping of the reactant or catalyst into a batch reactor (Barton [10]). In such cases, it would not be possible to use the polynomial approximations for the differential variables to do the initialization calculation using such a procedure. Ideally it would be preferable if one can take care of discontinuity sticking in some other fashion and do the reinitialization calculations without using the polynomial approximations.

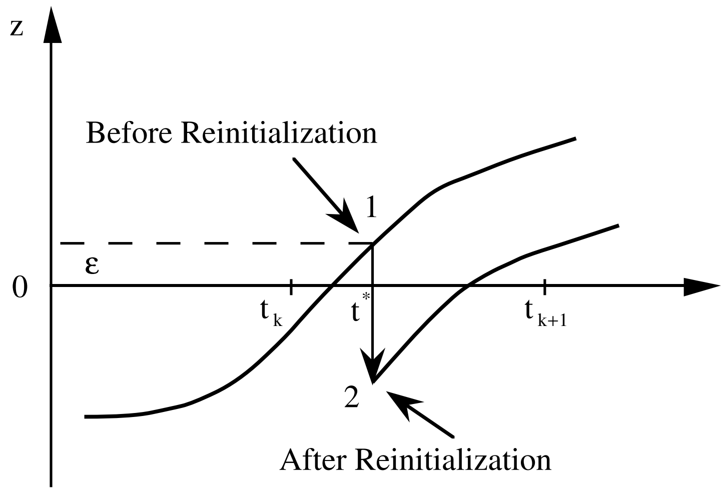

Figure 4.5: Discontinuity sticking

The phenomenon of discontinuity sticking occurs because the integration method does not guarantee consistency of differential and algebraic variables within the mesh points. However, BDF methods do guarantee their consistency at the mesh points. Thus no discontinuity sticking should occur if the point of discontinuity is chosen as a mesh point provided an appropriate tolerance is given for the switching function. Park and Barton (1995) point out that if the switching function is greater than some nonzero tolerance, they will avoid the sticking phenomena in their simultaneous location and reinitialization phase. It should be noted that the same tolerance should do the job in the independent location phase also if the problem is solved at the point of discontinuity. Such a procedure should take care of the problems which are associated with the simultaneous step, described in the previous section. The algorithm described in section 4.3.2 utilizes this idea.

Existence of Solution

An issue which has often been of concern in techniques using discontinuity locking is the question whether the current set of equations, which describes the region before the discontinuity, can also provide a solution in the region beyond the discontinuity. This is important because the discontinuity locking techniques require the current active set of equations to give a solution across the discontinuity for the detection and even location of the discontinuity (Figure 4.6). Nonexistence of solution across the discontinuity could cause presently used discontinuity locking methods to fail (Joglekar and Reklaitis [60], Pantelides and Barton [81]).

The LP based approach, however, has the advantage of having a penalty function in (2.2) to allow constraint relaxation if solutions do not exist. In this case, the algorithm will converge to a pseudosolution, and we can use this information to confirm that we have crossed a discontinuity.

Below we describe a procedure to address the issues outlined above, which is based on SLP and improves on those suggested in the references above.

4.3.2 Algorithm

0. *k=0.* x^0 known (Initial conditions)

1. Problem at the next time step: $t^{k+1} = t^k + h^k$

   Solve the discretized problem using the LP based algorithm described before. Let x^{k+1} be the converged solution at that time.

2. Check for discontinuities (zeros of switching functions) within the time interval. If

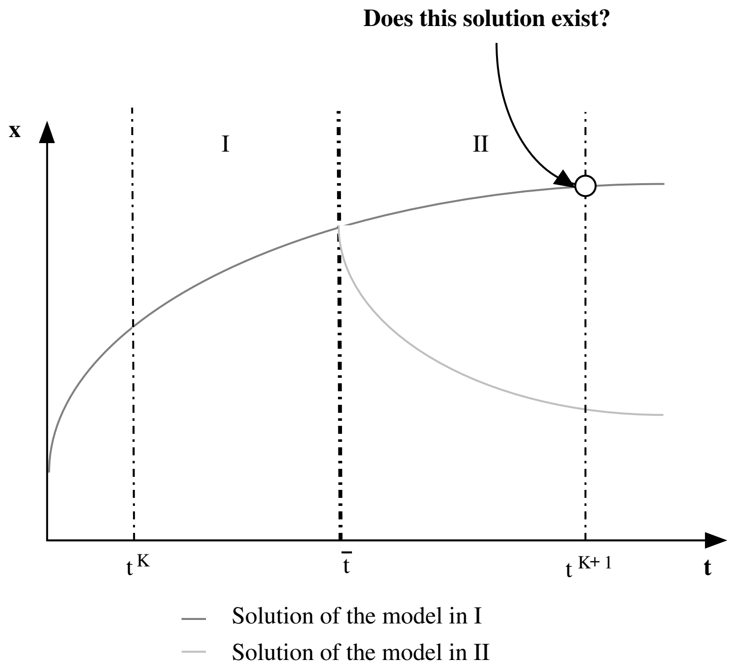

Figure 4.6: Existence of solution in discontinuity locking methods

no discontinuity detected, $k = k + 1$, go to 1. Else, continue.

3. Solve for t^* from $z^* = \pm tolerance$
   (The sign depends on the direction of crossing of the switching function. If the switching function is negative before the discontinuity, + sign is chosen and if it is positive, - sign is chosen)

4. Resolve the problem at t^*.

5. (a) If z^* does not lie within the tolerance specifications at t^* or
   (b) If some other z is found to be crossing zero from the new polynomial constructed from this reduced interval, go to 3 to re-evaluate with the new polynomial. Else

6. Reinitialize the problem at t^* for the new set of equations describing the system.

$k = k + 1$, go to 1.

Step 3 uses the same tolerance as in [82] and eliminates the problem of discontinuity sticking. Solving the problem at the point of discontinuity (step 4) helps to locate it accurately. Note that 5(a) makes sure that the time of occurrence of the discontinuity is determined within a specified tolerance, and 5(b) ensures that the detected discontinuity is indeed the earliest one.

4.3.3 Modeling Explicit Discontinuities

The algorithm described in this section efficiently treats the three important aspects of this problem: detection of the earliest state event, location of that event within specified error bounds, and reinitialization of the problem for the next interval. Nonsmooth dynamic simulation involves transition from one state of the system to another, or in other words, a change from one set of equations describing the process to another. To initialize the problem, smooth inequality constraints can be added to the LP formulation as described in Bullard and Biegler [25] to determine the set of equations which are initially active. Once this is known, we need to keep track of the logical conditions triggering a change from this model to a different one. Here the switching functions come into play. We know that an event has been triggered when one or more of these switching functions crosses zero. The new set of equations associated with that particular switching function takes over, and the simulation proceeds.

Explicit discontinuities occurring in chemical process simulation could be classified as history independent and dependent discontinuities. With history independent discontinuities the current state of the system is independent of the prior states. The transitions between the system states are allowed in any direction and are triggered by the same logical condition in all directions. Examples of these are simple pressure valves (opening and closing triggered by a certain fixed pressure). On the other hand, the current state of the system described by a history dependent discontinuity is dependent on the prior states. They are tougher to solve when compared to their history independent counterparts. As with current dynamic simulators, (see Pantelides and Barton [81]), we introduce an integer flag to represent this memory component. For illustration, let us consider an example.

4.3.4 Examples

Example 2: Three Tanks with Hysteresis Valves

Here we consider a problem with 3 tanks interlinked with 4 hysteresis valves as shown in Figure 4.7.

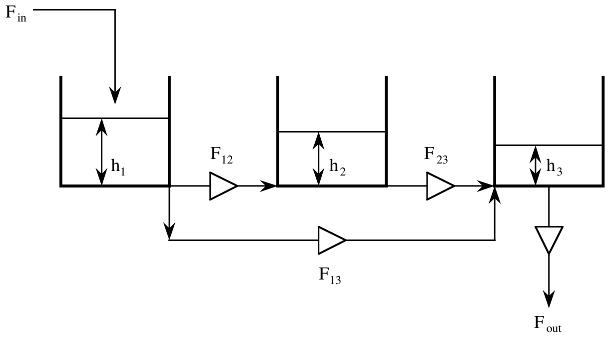

Figure 4.7: Three tanks problem - example 2.

Description of a hysteresis valve

Consider the valve ij fitted between tanks i and j. Let h_i and h_j be the liquid levels in these tanks. The hysteresis valve has two positions: closed (*flow=0*) and open (*flow=function(*$h_i - h_j$)). The logical conditions triggering a change in these two states are

1. Closed to open if

$$h_i - h_j \geq h_{ij}^U \tag{4.16}$$

2. Open to closed if

$$h_i - h_j \leq h_{ij}^L \tag{4.17}$$

When h_{ij}^L and h_{ij}^U are set to different values, the valve behavior can be graphically represented using a hysteresis curve (Figure 4.8). Two sets of equations are to be looked at when dealing with each valve: one which is valid when the valve is open and the other when the valve is closed. Let y_{ij}^0 be a flag for the current position of the valve ij.

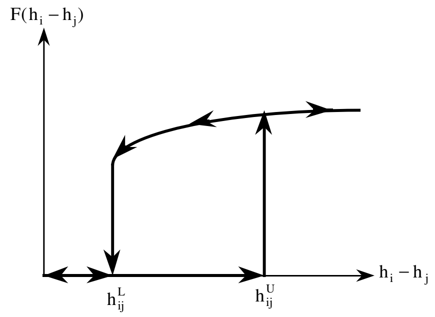

Figure 4.8: Hysteresis curve showing flow in the pipes.

Consider the case when the valve is closed ($y_{ij}^0 = 0$). From (4.16) when $h_i - h_j \geq h_{ij}^U$, the valve opens. So, the corresponding switching function will be

$$z_{ij}^{closed} = h_i - h_j - h_{ij}^U \tag{4.18}$$

The flow through the pipe

$$F_{ij}^{closed} = 0 \tag{4.19}$$

Similarly, when the valve is open ($y_{ij}^0 = 1$), the switching function will be

$$z_{ij}^{open} = h_i - h_j - h_{ij}^L \tag{4.20}$$

The relief flow in this case is

$$F_{ij}^{open} = \alpha_{ij}\sqrt{h_i - h_j} \tag{4.21}$$

We want to incorporate (4.18-4.19) when the valve is closed and (4.20-4.21) when the valve is open. The final set of equations will be

$$\begin{aligned} z_{ij} &= (h_i - h_j - h_{ij}^U)(1 - y_{ij}^0) + (h_i - h_j - h_{ij}^L)y_{ij}^0 \\ F_{ij} &= y_{ij}^0\left(\alpha_{ij}\sqrt{h_i - h_j}\right) \end{aligned} \tag{4.22}$$

Since y_{ij}^0 represents the current state of the valve (which is known), (4.22) reduces to (4.18-4.19) when the valve is closed and to (4.20-4.21) when the valve is open. A switch in valve position is detected by a zero crossing of z_{ij}, and the value of y^0_{ij} is changed when the integration continues from the point of discontinuity.

The hysteresis nature of the valves causes the history of the system to play an important part in determining the position of any of these valves at a given time. Therefore, it is simpler to model this system using the procedure described above with the position of each valve being monitored by (4.22). The dynamics of the system of tanks and valves in Figure 4.7 can be modeled using (4.23).

$$\begin{aligned}
&A_1\frac{dh_1}{dt} = F_{in} - F_{12} - F_{13}\\
&A_2\frac{dh_2}{dt} = F_{12} - F_{23}\\
&A_3\frac{dh_3}{dt} = F_{13} - F_{23} - F_{out}\\
&F_{ij} = y^0_{ij}\left(\alpha_{ij}\sqrt{h_i - h_j}\right) \qquad ij = 12, 13, 23, 34\\
&z_{ij} = (h_i - h_j - h^U_{ij})(1 - y^0_{ij}) + (h_i - h_j - h^L_{ij})y^0_{ij} \qquad ij = 12, 13, 23, 34\\
&(h_4 = 0)
\end{aligned} \tag{4.23}$$

The variables h are the liquid levels in the tanks, F the flowrates in the pipes and α the valve constants. This system was simulated using the LP based algorithm. The values for the valve constants and the breakpoints are given in Table 4.1. The initial conditions correspond to all tanks empty and all valves closed. It should be noted that at this initial point, care must be taken in the initialization, as the valve positions are related to the initial liquid levels and cannot be specified arbitrarily. The constants were chosen to demonstrate the opening and closing of all the valves within a short time. Figure 4.9 shows the levels of liquid in the three tanks. The driving force, the difference in liquids levels in the tanks is plotted in Figure 4.10. The flowrate in the pipes is shown in Figure 4.11.

4.4 Implicit Discontinuities

Implicit discontinuities often arise in process simulation problems. Explicit discontinuities described in section 4.3 can be captured using switching functions which are explicitly

Table 4.1: Parameters for the three tanks problem.

ij	h_{ij}^L	h_{ij}^U	α_{ij}
12	30	75	12
13	60	100	12
23	30	50	15
3 out	40	50	10

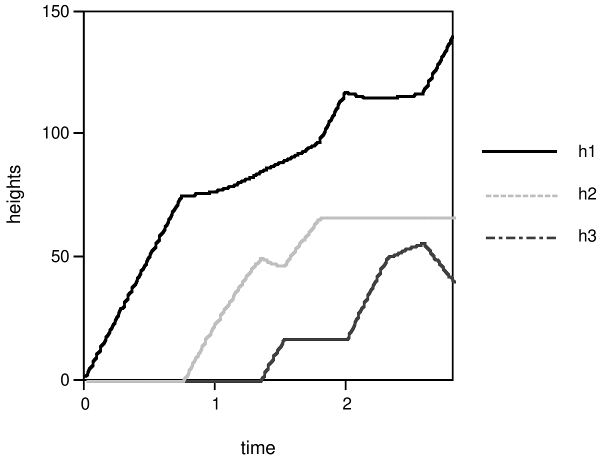

Figure 4.9: Height of liquid in the tanks.

available. However, implicit discontinuities are characterised by regions of continuous operation with nonsmooth boundaries. In fact, they could be mathematically described using complementarity conditions. We will describe complementarity problems in detail in chapter 6.

In this section, we consider a specific class of implicitly discontinuous problems encoun-

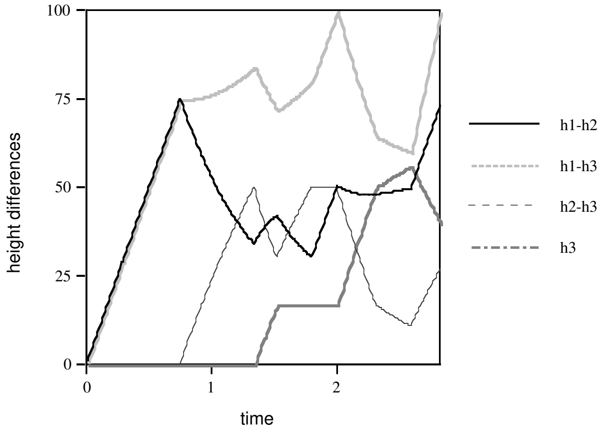

Figure 4.10: The difference in levels of liquid in the tanks

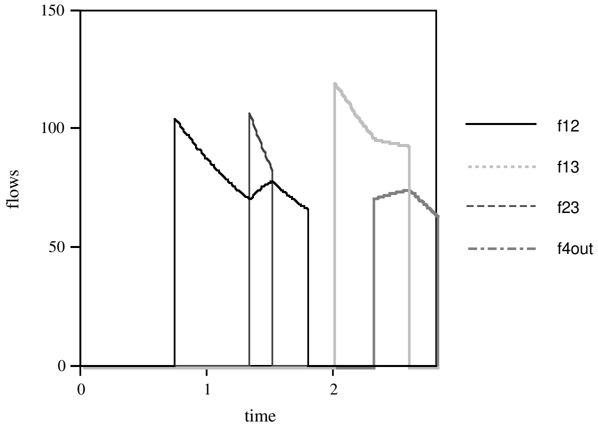

Figure 4.11: Flowrates in the pipes

tered often in chemical process simulation: calculation of phase equilibrium. Consider a simple isothermal flash (Figure 4.12) to highlight some of the basic issues which are relevant in this problem. A feed stream of composition $\boldsymbol{z}$ enters the flash column at a specified temperature and pressure. The products are a vapor phase stream of composition $\boldsymbol{y}$ and a fractional flowrate V and/or a liquid phase stream of composition $\boldsymbol{x}$. Different sets of equations are valid depending on the number of phases present at equilibrium. For example, the equilibrium equations are generally not valid in either of the single phase regions. The combined set of equations is an implicitly nonsmooth problem with well defined smooth regions. Thus, one of the basic problems associated with phase equilibrium calculations, especially in an equation oriented simulation environment, is that the number of phases is not known *a priori*.

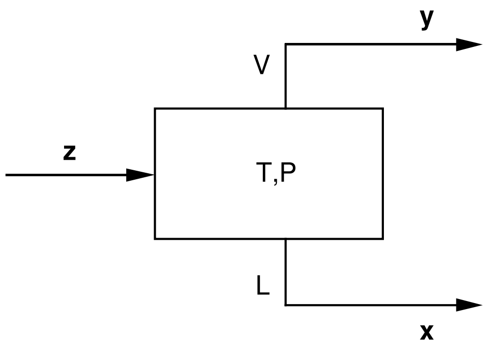

Figure 4.12: Isothermal flash operation

Among the different approaches suggested to tackle this problem, the classical *sequential modular* approach is the most familiar one (Boston and Britt [19], Nelson [77]). In general, these methods calculate the bubble and dew points (in the case of two phase calculations) and then determine the number of phases at equilibrium. The set of equations applicable to that particular phase behavior is then solved. Such a procedure based approach, however, can be difficult to incorporate in an equation oriented simulation environment.

Phase equilibrium problems are an important class of problems which can be addressed using our LP-based method. Bullard and Biegler [26] applied a penalty based extension of their LP formulation, P-SONATA, to deal with two phase vapor liquid equilibrium (VLE) problems. The basic idea of their formulation was to introduce a *pseudo* pressure P_P and use it to solve the equilibrium expression in all the regions. P_P is allowed to differ from the specified pressure P_S in the phase equilibrium expression if P_S goes above the bubble or below the dew point pressures and the equilibrium expression is relaxed in the one phase regions. A consistent set of equations is thus obtained in both one and two phase regions by adding a penalty term to the objective to account for differences from equilibrium. The pressure relaxation formulation gave promising results for steady state two phase systems including isothermal flash problems with ideal and nonideal phase equilibrium relations and limiting distillation cases. However, it would be difficult to extend such a pressure relaxation formulation to a multiphase system. In the following section we introduce a new formulation to deal with multiphase systems as well.

4.4.1 Formulation for Multiphase Systems

The following formulation is based on the idea that if two phases do not coexist at equilibrium, then the corresponding equilibrium expression could be relaxed. This could be done by relaxing the equilibrium constant itself, rather than dealing with pressure as in Bullard and Biegler [26]. NLP (4.24) is formulated based on such a relaxation. Thus the new formulation is not limited to VLE systems and could be used for multiphase systems as well.

$$
\begin{array}{lll}
 & Min \sum_p \delta_p & \\
s.t. & F - \sum_p M_p = 0 & \\
 & F z_i - \sum_p M_p x_i^p = 0 & i = 1, n_c \\
 & x_i^{p_{ref}} - \gamma_p K_i^p x_i^p = 0 & p = 1, n_p, \quad p \neq p_{ref} \\
 & & i = 1, n_c \\
 & \sum_i x_i^{p_{ref}} - \sum_i x_i^p = 0 & p = 1, n_p, \quad p \neq p_{ref} \\
 & \delta_p \geq \gamma_p - 1 & p = 1, n_p, \quad p \neq p_{ref} \\
 & \delta_p \geq 1 - \gamma_p & p = 1, n_p, \quad p \neq p_{ref} \\
 & \delta_p \geq 0 & \\
 & 0 \leq x_i^p \leq 1 & \\
 & 0 \leq M_p \leq F &
\end{array}
\tag{4.24}
$$

M_p is the molar flowrate of phase p, x_i^p is the mole fraction of component i in phase p and p_{ref} is a reference phase based on which the equilibrium expressions are defined. $\bar{K}_i^p$ is a 'pseudo' equilibrium constant, which is defined as $\bar{K}_i^p = \gamma_p K_i^p$, where K_i^p is the equilibrium constant computed at the specified temperature and pressure.

The approach we propose here is related to a Gibbs free energy minimization formulation. In Appendix B, we show that the Kuhn-Tucker conditions of the Gibbs minimization problem are equivalent to the solution of (4.24) for a multiphase system. The relationship of γ_p at the solution to the existence of phases is also given in Appendix B. For a two-phase flash, (4.24) reduces to (4.25).

$$\begin{aligned}
& Min\ \delta \\
s.t.\quad & F - L - V = 0 \\
& Fz_i - Lx_i - Vy_i = 0 \qquad i = 1, n \\
& y_i - \gamma K_i(P, T, x)x_i = 0 \qquad i = 1, n \\
& \sum_i y_i - \sum_i x_i = 0 \\
& \delta \geq \gamma - 1 \\
& \delta \geq 1 - \gamma \\
& \delta \geq 0 \\
& 0 \leq x_i, y_i \leq 1 \\
& 0 \leq L, V \leq F
\end{aligned} \tag{4.25}$$

In the two phase region, $\gamma = 1$ and the equilibrium expression is satisfied. In the single phase regions, γ differs from 1 and the equilibrium expression is relaxed. In Appendix B we show that $\gamma > 1$ in the single phase liquid region and $\gamma < 1$ in the single phase vapor region.

Recently Swaney and Kendlbacher [93] derived a formulation where the summation equations were relaxed as in (4.26).

$$\sum_i x_i^p - 1 + \psi^p = 0 \tag{4.26}$$

They showed that the slacks ψ^p and the phase amounts M^p (e.g., V or L^p) are complementary.

$$M^p \geq 0 \qquad \psi^p \geq 0 \qquad M^p\psi^p = 0 \tag{4.27}$$

Such a formulation could also be solved reliably using iterated LP. In chapter 6 we present an equation based simplified formulation derived from the optimality conditions of the two phase problem (4.25).

In the following section, we consider a broad range of example problems, to illustrate the applicability of the formulation.

4.4.2 Examples

Example 3: Phase Equilibrium in the Critical Region

An interesting example problem is to simulate the behavior of complex phase equilibrium systems near the critical point. In this problem we consider a mixture of Nitrogen, Methane, Ethane and Propane at a pressure of 460 psia and a temperature range of -140F to -50F. The Peng-Robinson cubic equation of state [83] is used for the calculations.

For the PT flash, (4.25) was used for the calculation of phase equilibrium. To deal with numerical difficulties which might occur if the pressure goes negative during any of the iterations, an exponential transformation was used.

$$For \quad P \leq \epsilon, \quad P = \epsilon \, \exp\{\frac{(P-\epsilon)}{\epsilon}\} \tag{4.28}$$

Constraints for minimum values of $\partial P/\partial \rho$ were also incorporated to make sure that the search directions remained in a stable 'allowed' region (Mathias et al. [70]). The three cases presented here correspond to T = -50.796 F (vapor phase), T = -96.019 F (two phase region) and T = -135.67 F (liquid phase). In all the cases x and y were initialized to the feed composition. Initial guesses for v_{PR}^L and v_{PR}^V (the corrected molar volumes) correspond to compressibility factors of 0.1 and 0.8 respectively. Table 4.4.2 gives the number of iterations required to converge to a solution for initial guesses of V as 0 and 1.

Example 4: Dynamic Simulation of a Two-Phase Flash Covering Three Regimes of Operation

In this example we consider an n-butane, n-pentane, n-hexane system where the ideal flash unit is modeled using the Antoine equation. A constant feed of mole fractions 0.3, 0.3 and 0.4 for the respective components is supplied to the unit. The flash is carried out at a pressure of 7600 mm Hg. The temperature is varied linearly from $T_i = 385K$ to $T_f = 420K$ from time $t_0 = 0$ to $t_f = 10$. It should be noted that the dew and bubble

T(F)	V=0	V=1
-50.796	8	8
-96.019	40	55
-135.67	7	10

Table 4.2: Number of iterations for the critical region problem

points of the feed lie in this range. The motivation behind choosing this example problem is to see whether the algorithm could capture the transition from a single phase region to a two phase region and vice versa as the temperature was varied.

$$\begin{aligned}
& Min \quad \delta \\
s.t. \quad & \frac{d\boldsymbol{H}}{dt} = \boldsymbol{f} - V\boldsymbol{y} - L\boldsymbol{x} \\
& \frac{dM_l}{dt} = \sum_i f_i - V - L \\
& \boldsymbol{H} = M_l \boldsymbol{x} \\
& L = \alpha\sqrt{M_l} \\
& \boldsymbol{y} = \gamma \boldsymbol{K}(T)\boldsymbol{x} \\
& \sum_i y_i - \sum_i x_i = 0 \\
& T = T(t) \\
& -\delta \leq \gamma - 1 \leq \delta \\
& \delta \geq 0 \\
& 0 \leq x_i, y_i \leq 1
\end{aligned} \tag{4.29}$$

The flash unit was modeled as in (4.29) where L is the liquid flowrate, V is the vapor flowrate, $\boldsymbol{H}$ is the component molar holdup, $\boldsymbol{x}$ and $\boldsymbol{y}$ are the component mole fractions in the liquid and vapor phases and M_l is the total liquid holdup. All variables in (4.29) are functions of time. Vapor holdup is neglected in this formulation. The holdups were discretized using implicit Euler and the corresponding set of equations was solved using iterated LP at time steps of 0.1. Figure 4.13 shows the liquid and vapor stream flowrates with time.

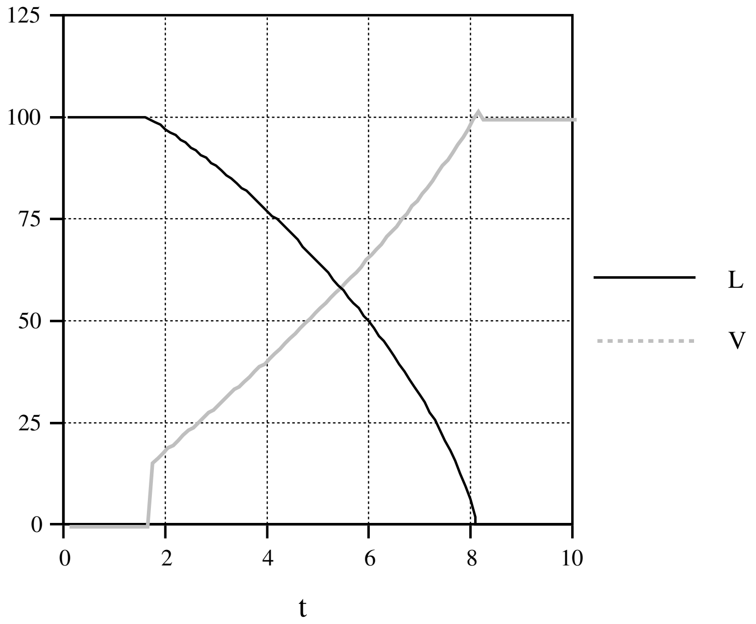

Figure 4.13: Liquid and vapor flowrates in example 3.

The plot indicates that the algorithm handled the transition from the single phase liquid region to the two phase region and then to the single phase vapor region well. The average number of iterations for solving the problems per time step was 4.1. In the single phase regions, solution of the discretized equations on an average took 3 iterations. The problem, when discretized with smaller steps and when solved using a second order method, trapezoidal rule, gave similar results.

Example 5: Dynamic Simulation of a Nonideal Three Phase Flash

This example examines a benzene - isopropanol - water system considered in steady state by Pham and Doherty [85]. For different temperature - pressure conditions this system could exist as a LLV (liquid-liquid-vapor) or LL (liquid-liquid) or LV (liquid-vapor) or just as a single phase liquid or single phase vapor. To accommodate these possibilities, we posed the formulation (4.30) for the dynamic simulation of this system.

$$\begin{aligned}
& Min \quad \delta^I + \delta^{II} \\
s.t. \quad & \frac{d\boldsymbol{H}}{dt} = \boldsymbol{f} - V\boldsymbol{y} - L^I\boldsymbol{x^I} - L^{II}\boldsymbol{x^{II}} \\
& \frac{d}{dt}(M_l^I + M_l^{II}) = \sum_i f_i - V - L^I - L^{II} \\
& \boldsymbol{H} = M_l^I\boldsymbol{x^I} + M_l^{II}\boldsymbol{x^{II}} \\
& L^I + L^{II} = \alpha\sqrt{M_l^I + M_l^{II}} \\
& \frac{L^I}{L^{II}} = \frac{M_l^I}{M_l^{II}} \\
& \boldsymbol{y} = \gamma^I\boldsymbol{K^I}\boldsymbol{x^I} \\
& \boldsymbol{y} = \gamma^{II}\boldsymbol{K^{II}}\boldsymbol{x^{II}} \\
& \sum_i y_i - \sum_i x_i^I = 0 \\
& \sum_i y_i - \sum_i x_i^{II} = 0 \\
& T = T(t) \\
& -\delta^I \leq \gamma^I - 1 \leq \delta^I \\
& -\delta^{II} \leq \gamma^{II} - 1 \leq \delta^{II} \\
& \delta^I, \delta^{II} \geq 0 \\
& 0 \leq x_i^I, x_i^{II}, y_i \leq 1
\end{aligned} \tag{4.30}$$

As in example 4, a constant feed of mole fractions 0.5, 0.08 and 0.42 of benzene, isopropanol and water respectively was fed into the flash tank. At constant pressure of 1 atm, the temperature is decreased linearly from 70^0 C to 68^0C. The initial conditions correspond to steady state values at 70^0C. The activity coefficients for this system were calculated using the regular solution model. Figure 4.14 plots the flowrates of the two liquid and the vapor streams with time. Liquid phases 1 and 2 correspond to the water rich and benzene rich phases respectively. As shown in the figure, initial conditions correspond to a liquid-vapor system. A second liquid phase appears later on and the system enters the three phase $L^I - L^{II} - V$ region. Upon further decrease of temperature, the vapor phase vanishes altogether, resulting in a two phase liquid region. The algorithm was thus successful in simulating the three phase LLV and the two phase LV and LL regions with a single formulation. An average of 3.4 iterations was required to solve the discretized equations at each time step.

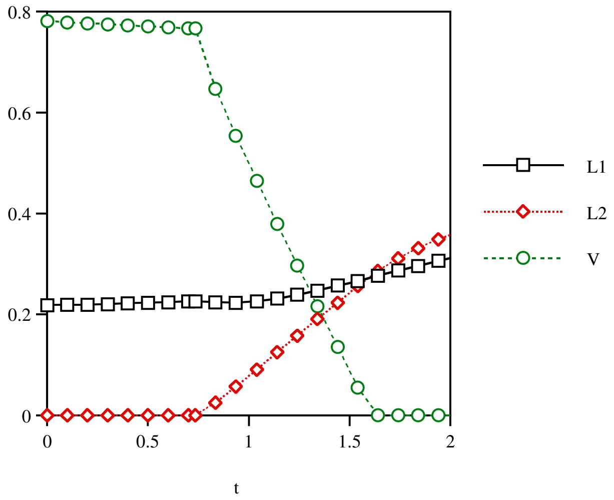

Figure 4.14: Flowrates of the two liquid and vapor streams in example 4

SLP method solved a difficult steady state phase equilibrium problem in the critical region reliably in example 3. Examples 4 and 5 show that the penalty based SLP strategy could also be a strong tool for solving dynamic problems involving phase transitions, without having to compute the dew and bubble points and then determining the number of phases at any time. The transitions between phase combinations were handled efficiently by a single consistent formulation. Recently, several studies (McDonald and Floudas [71], Swaney and Kendlbacher [93], Stadtherr *et al.* [91], Sun and Seider [92]) describe global algorithms for solving phase equilibrium problems of this type and for assessing the stability by using the tangent plane criterion. Beyond initialization and at a limited number of points, however, applying a global optimization algorithm to dynamic phase equilibrium problems could be expensive and often unnecessary. Instead, the equilibrium solution at a particular time step gives good starting points for the problem at the next time step, and therefore tracks the optimum locally. A formulation such as the one presented in this paper has its relevance in this context.

4.5 Conclusions

The SLP method developed in chapter 2 was applied to address a variety of issues in process simulation. The LP based method has been demonstrated to be an effective way for the imposition of bounds on variables in dynamic simulation problems. An improved method for the treatment of discontinuities occurring in nonsmooth dynamic simulation problems has been developed. The method ensures that any event which has occurred within the time interval in consideration is detected, and if more than one event occurs, the detected one is indeed the earliest one. In the case of nonexistence of solution across a discontinuity, a penalty term introduced in our approach makes it less likely to fail when compared to the presently used discontinuity locking methods. A specific class of process simulation problems, phase equilibrium calculations, has been looked at as a special case. A new formulation for solving multiphase equilibrium problems has been presented. A penalty term introduced in the objective takes care of the appearance and disappearance of phases. Example problems have been solved to demonstrate the feasibility of the approach.

Chapter 5

Interior Point Methods for Large-Scale Process Optimization

The science of mathematics, in its modern developments,
may claim to be the most original creation of human spirit.
- Alfred Whitehead, Science and the Modern World [99].

Synopsis

Successive Quadratic Programming (SQP) has been the method of choice for the solution of nonlinear programming problems. For the solution of large problems with SQP based codes, the combinatorial complexity associated with active set methods for the treatment of inequalities has been a bottleneck in exploiting the problem structure. In this chapter, we examine the merits of incorporating an interior point method within an SQP framework and the significant computational savings are demonstrated on problems drawn from optimal control. We also provide a novel interpretation of the popularly used predictor-corrector interior point method using the principles of reduced Hessian SQP (rSQP).

5.1 Introduction

Chapter 3 describes the solution of the problem of initialization and reinitialization after discontinuities of DAEs using Successive Linear Programming. In chapter 4, SLP methods were also used for the treatment of bounds, conditionals and discontinuities occuring in process simulation. The next question we asked ourselves in the course of this research was how to improve the efficiency of solving these SLP subproblems, by focusing on salient features of these problems. The general nature of these problems is that they are highly constrained, with the constraints arising from physically and mathematically realistic conditionals and variable bounds.

In fact, being highly constrained is a common feature of most continuous *optimization* problems occuring in process engineering. Recognition of this motivated us to expand the horizons of the applicability of the problems we were looking at. From this perspective, we will look at inequality constrained SQP problems in this chapter. However, it should be noted that, the entire analysis, by setting the quadratic term in the objective of the QP subproblem to zero, is valid for SLP problems as well.

With the widespread use of Successive Quadratic Programming (SQP) for process optimization in design, operations and control, a variety of different implementations and strategies have been developed for this algorithm. One important benefit for SQP is its ability to be modified in order to take advantage of a particular problem structure. Problems in parameter estimation [2], multiperiod optimization [98] and dynamic optimization [94], amongst others, present large, sparse systems with specific structures.

However, one hurdle in exploiting the problem structure is an efficient strategy for treating inequality constraints in the QP subproblem. Here, while most of the commonly used QP codes use active set strategies, it has been observed that they can become combinatorially expensive for large problems. A popular alternative has been to explore interior point methods. Previous studies in the application of interior point strategies to SQP have shown considerable promise for these strategies (Morales and Sargent [74], Kyriakopoulou and Kalitventzeff [64]). However, it is especially advantageous when applied to

SQP strategies that exploit the structure of the optimality conditions (without inequality constraints).

To develop this approach, we incorporate the characteristics of the well studied Mehrotra [72] predictor-corrector method, which is a path following algorithm within an SQP framework for the solution of nonlinear programming problems. The method could be used in a full space or reduced space for solving large QP problems. Moreover, it has been tested on a battery of literature examples, particularly the netlib test set, that involve LPs, QPs and LCPs. In addition, this interior point scheme has been coupled to efficient sparse decomposition strategies that can be modified to take advantage of a particular problem structure.

To illustrate these ideas, we consider a class of optimal control problems that have this characteristic. These problems arise in dynamic state and parameter estimation and in model predictive control. This approach is demonstrated on structured optimization problems of varying sizes.

5.2 Developments in Interior Point Methods

Ever since the publication of the landmark paper by Karmarkar [62] in 1984, interior point methods have been an active research area in the optimization community. The 1990s have seen the emergence of *primal dual methods* as an important class of interior point algorithms from a computational as well as a theoretical standpoint. Below, the concept behind primal-dual methods is explained considering a linear program (LP) (5.1) and its dual (5.2).

$$min\ c^T x, \qquad s.t.\ Ax = b,\ x \geq 0 \tag{5.1}$$

$$max\ b^T \lambda, \qquad s.t.\ A^T\lambda + s = c,\ s \geq 0 \tag{5.2}$$

Duality theorem states that for any feasible vectors (x) for 5.1 and (λ,s) for (5.2) the relation $b^T\lambda \leq c^T x$ exists and the two objective functions coincide at the solution. Primal-dual methods solve the optimality conditions (5.3) of the problem (5.1),(5.2) by applying variants of the Newton's method. Various methods differ in the way the search directions are generated and step lengths are enforced to ensure nonnegativity of x,s.

$$\begin{aligned} Ax &= b \\ A^T\lambda + s &= c \\ x_i s_i &= 0 \\ (x,s) &\geq 0 \end{aligned} \tag{5.3}$$

The monograph by Wright [100] gives an excellent description of primal- dual interior point methods. The extension to nonlinear problems is however nontrivial and has been receiving considerable attention recently.

5.2.1 Nonlinear Programing Problems

Consider the nonlinear programming problem

$$\begin{aligned} min \quad & f(z) \\ s.t. \quad & g(z) = 0 \\ & h(z) \leq 0 \end{aligned} \tag{5.4}$$

Bounds on variables are not explicitly treated in (5.4) as they are a special case of the inequality constraints. The SQP method for solving (5.4) solves the following QP subproblem (5.5) at the current point z_k to generate a search direction x_k.

$$\begin{aligned} min \quad & c(z_k)^T x + \frac{1}{2} x^T Q x \\ s.t. \quad & g(z_k) + Ax = 0 \\ & h(z_k) + Cx \leq 0 \end{aligned} \tag{5.5}$$

$c = \nabla f$, Q is the Hessian of the Lagrangian or a positive definite approximation to it, $A = \nabla g^T$ and $C = \nabla h^T$. Expressing the constants in each subproblem on the right hand side and introducing slack variables s, (5.5) can be written as

$$\begin{aligned} min \quad & c^T x + \frac{1}{2} x^T Q x \\ s.t. \quad & Ax = b \\ & Cx + s = d \\ & s \geq 0 \end{aligned} \tag{5.6}$$

The optimality conditions of (5.6) are given by (5.7)-(5.11).

$$Qx + A^T\lambda + C^T\nu + c = 0 \tag{5.7}$$

$$Ax = b \tag{5.8}$$

$$Cx + s = d \tag{5.9}$$

$$SVe = 0 \tag{5.10}$$

$$(s, \nu) \geq 0 \tag{5.11}$$

λ and ν are the multipliers of the equalities and inequalities respectively. S and V are diagonal matrices of s and ν.

This system of equations could be solved iteratively by applying Newton's method to a linearization of (5.7)-(5.10) and carrying out a line search to enforce the non-negativity constraints (5.11). However, this could result in some of the components of (s, ν) from getting too close to the boundary of the nonnegative orthant, and subsequent iterates making little progress towards the solution. This limitation has given rise to the concept of what is known as the *central path* in interior point literature. It is an arc which keeps the iterates biased towards the interior of the nonnegative region $(s, \nu) > 0$, following which forces all the complementarity pairs to converge at a similar rate. The *complementarity gap* μ is a measure of the average value of the products $s_i\nu_i$ and is defined as:

$$\mu = s^T\nu/l \tag{5.12}$$

Here, l is the number of inequalities in (5.5) and hence the number of slacks (5.6). A typical strategy used to keep the iterates in the interior is to replace (5.10) with (5.13) where $0 \leq \sigma \leq 1$ is a parameter.

$$SVe = \sigma\mu e \tag{5.13}$$

$\sigma = 1$ corresponds to a pure centering direction, *ie.* moving all the pairwise products to the current average value μ. On the other hand, $\sigma = 0$ gives a standard Newton step. Various *path following methods* differ in the way the search direction is modified and σ is chosen. In the following section we will adapt it to an SQP framework and provide a novel, intuitive interpretation of this procedure in this context.

5.3 Interior point SQP

SQP has been the method of choice for the solution of nonlinear programming problems. The superlinear convergence properties and the ability to exploit the problem structure have been well studied by many researchers in the past decade. For process engineering problems, in the absence of inequality constraints, various decomposition strategies can be applied to improve the efficiency for large scale problems. On the other hand, application of active set strategies to the imposition of inequality constraints can make these problems much more expensive to solve. The motivation to incorporate an interior point strategy within an SQP framework arises from this handicap of the active set strategies. We would like to integrate the interior point method and SQP in such a way that we still retain fast linear performance as well as the structured decomposition.

To develop this approach, we apply certain characteristics of the well studied Mehrotra [72] predictor-corrector algorithm. The predictor-corrector algorithm solves the system of equations (5.14) iteratively.

$$\begin{aligned} &Qx + A^T\lambda + C^T\nu + c = 0 \\ &Ax = b \\ &Cx + s = d \\ &SVe = \sigma\mu e \\ &(s,\nu) \geq 0 \end{aligned} \tag{5.14}$$

Each Mehrotra iterate solves an affine scaling *predictor* step (5.15) and a *corrector* step (5.16) alternately. r_1, r_2, r_3 are the residuals of the equations at the current point. The predictor step helps in reducing the complementarity gap and centering is accomplished through the corrector step which also incorporates a second order correction term.

$$\begin{bmatrix} Q & A^T & C^T & 0 \\ A & 0 & 0 & 0 \\ C & 0 & 0 & I \\ 0 & 0 & S & V \end{bmatrix} \begin{bmatrix} \Delta x^p \\ \Delta\lambda^p \\ \Delta\nu^p \\ \Delta s^p \end{bmatrix} = - \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ SVe \end{bmatrix} \tag{5.15}$$

$$\begin{bmatrix} Q & A^T & C^T & 0 \\ A & 0 & 0 & 0 \\ C & 0 & 0 & I \\ 0 & 0 & S & V \end{bmatrix} \begin{bmatrix} \Delta x^c \\ \Delta \lambda^c \\ \Delta \nu^c \\ \Delta s^c \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \sigma \mu e - \Delta S_p \Delta V_p e \end{bmatrix} \tag{5.16}$$

In the remainder of this section, we will discuss how to interpret the predictor-corrector method using a reduced Hessian SQP (rSQP) framework and how these concepts could be incorported within an SQP framework. We briefly review rSQP as discussed in Biegler *et al.* [14] in the following subsection.

5.3.1 Reduced Hessian SQP (rSQP)

Consider the quadratic programming subproblem

$$\begin{aligned} min \quad & \overline{c}(z_k)^T x + \frac{1}{2} x^T \overline{Q} x \\ s.t. \quad & \overline{g}(z_k) + \overline{A} x = 0 \end{aligned} \tag{5.17}$$

$z, x \in \Re^n$, $g \in \Re^m$. From [14], the solution of (5.17) is expressed in a simpler fashion by choosing a suitable basis in $\Re^n$ given by a nonsingular matrix (5.18).

$$[Y \quad Z] \tag{5.18}$$

where $Y \in \Re^{n \times m}$, $Z \in \Re^{n \times (n-m)}$. Also,

$$\overline{A}^T Z = 0 \tag{5.19}$$

Note that $(n - m)$ corresponds to the degrees of freedom for the QP subproblem. The search direction x can be expressed as

$$x = Y p_Y + Z p_Z \tag{5.20}$$

Then, p_Y and p_Z are given by

$$p_Y = -\left[\overline{A}^T Y\right]^{-1} \overline{g} \tag{5.21}$$

$$p_Z = -\left[Z^T \overline{Q} Z\right]^{-1} \left[Z^T \overline{c} + Z^T \overline{Q} Y p_Y\right] \tag{5.22}$$

In [14], the matrices Z and Y are choosen as

$$Z = \begin{bmatrix} -M^{-1}N \\ I \end{bmatrix} \qquad Y = \begin{bmatrix} I \\ 0 \end{bmatrix} \tag{5.23}$$

where

$$\overline{A}^T = [M \quad N] \tag{5.24}$$

M is an $m \times m$ nonsingular matrix. rSQP methods are generally preferred for problems with a few degrees of freedom *ie.* when $(n-m)$ is relatively small. In the following section, we will see how rSQP could be used to interpret predictor-corrector methods.

5.3.2 Interpretation of Interior Point Methods with Reduced Hessian SQP

To interpret the predictor-corrector method, let us pose the subproblem (5.6) as (5.25) with the optimality conditions (5.7)-(5.9) as constraints as in [3]. Note that the complementarity constraint is different from (5.10) and (5.13).

$$\begin{aligned} min \quad & f(q) \\ s.t. \quad & Qx + A^T\lambda + C^T\nu + c = 0 \\ & Ax = b \\ & Cx + s = d \\ & SVe = qe \end{aligned} \tag{5.25}$$

Since (5.25) has just 1 degree of freedom (q), consider the iterative solution of (5.25) in the context of a reduced Hessian SQP (rSQP) framework. Linearization of the constraints in (5.25) give:

$$\begin{bmatrix} Q & A^T & C^T & 0 & 0 \\ A & 0 & 0 & 0 & 0 \\ C & 0 & 0 & I & 0 \\ 0 & 0 & S & V & -e \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta\nu \\ \Delta s \\ \Delta q \end{bmatrix} = - \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ S_kV_ke - q_ke \end{bmatrix} \tag{5.26}$$

M and N in (5.24) are chosen as in (5.27) and $\overline{Q}$ is given by (5.28). θ is the multiplier for the complementarity constraints in (5.25).

$$\underbrace{\begin{bmatrix} Q & A^T & C^T & 0 \\ A & 0 & 0 & 0 \\ C & 0 & 0 & I \\ 0 & 0 & S & V \end{bmatrix}}_{M} \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ -e \end{bmatrix}}_{N} \tag{5.27}$$

$$\overline{Q} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \theta & 0 \\ 0 & 0 & \theta & 0 & 0 \\ 0 & 0 & 0 & 0 & f''(\mu) \end{bmatrix} \tag{5.28}$$

Now if f is such that $f(0) = 0$ and $\partial f/\partial\mu(0) = 0$, then $\theta \to 0$ at the solution. Following the analysis in section 5.3.1, it can be shown that

$$\Delta q = -f'(q)/f''(q) \tag{5.29}$$

The null space and range space steps are given by (5.30) and (5.31).

$$\begin{bmatrix} Q & A^T & C^T & 0 \\ A & 0 & 0 & 0 \\ C & 0 & 0 & I \\ 0 & 0 & S & V \end{bmatrix} \begin{bmatrix} \Delta x^z \\ \Delta\lambda^z \\ \Delta\nu^z \\ \Delta s^z \end{bmatrix} = - \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{f'(q)}{f''(q)} e \end{bmatrix} \tag{5.30}$$

$$\begin{bmatrix} Q & A^T & C^T & 0 \\ A & 0 & 0 & 0 \\ C & 0 & 0 & I \\ 0 & 0 & S & V \end{bmatrix} \begin{bmatrix} \Delta x^y \\ \Delta\lambda^y \\ \Delta\nu^y \\ \Delta s^y \end{bmatrix} = - \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ S_k V_k e - q_k e \end{bmatrix} \tag{5.31}$$

The question which is relevant in this context is what would be an appropriate choice of the objective function $f(q)$ which would capture the characteristics of the Mehrotra predictor-corrector method. We examine two choices for the objective function.

Exact replication

Theorem 5.1 *The choice of f in (5.25) which exactly replicates a Mehrotra predictor-corrector framework is given by the solution of (5.32) with $f(0)$, $f'(0) = 0$.*

$$\frac{f'}{f''} = \mu - \frac{\hat{C}_1}{2C_2} \pm \frac{\sqrt{\hat{C}_1^{\,2} + 4C_2\mu(1-\hat{\alpha})^2 - 4C_2\bar{C}\hat{\alpha}^2(1-\hat{\alpha})}}{2C_2} \tag{5.32}$$

where

$$\hat{C}_1 = \frac{1}{n}\left[(2e^T V^{-1} B_1^T A_2 B_1 D_2)\nu + e^T(2V^{-1}B_1^T A_2 B_1 B_2^T + S^{-1}B_2^T)\overline{r}_1\right] - 1 \tag{5.33}$$

$$C_2 = \frac{1}{n} e^T S^{-1} D_2 (D_1 D_2 - I) S^{-1} e \tag{5.34}$$

$$\overline{C} = \frac{1}{n}(D_1 B_2^T \overline{r}_1 + B_1^T A_2 B_1 s)^T (B_2^T \overline{r}_1 + D_2 \nu) \tag{5.35}$$

$$\overline{r_1} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} \tag{5.36}$$

$$A_1 = \begin{bmatrix} Q & A^T & C^T \\ A & 0 & 0 \\ C & 0 & 0 \end{bmatrix} \tag{5.37}$$

$$B_1^T = [0 \quad 0 \quad I] \tag{5.38}$$

$$D_1 = S^{-1}V \tag{5.39}$$

$$\begin{bmatrix} A_1 & B_1 \\ B_1^T & D_1 \end{bmatrix}^{-1} = \begin{bmatrix} A_2 & B_2 \\ B_2^T & D_2 \end{bmatrix} \tag{5.40}$$

Proof

Proof is given in Appendix C. □

Forms of f from the solution of (5.32) under various assumptions are also presented in Appendix C.

Modified framework

We now consider a slightly modified framework, where the salient characteristics of the Mehrotra predictor corrector framework are captured, but at the same time simpler than the one in Theorem 5.1. Let us choose

$$f(q) = \frac{1}{K} q^K \tag{5.41}$$

where K is a constant. Then,

$$\frac{f'}{f''} = \frac{q}{K-1} \tag{5.42}$$

Define c as in (5.43). For $c > 0$, $K > 2$. At the limit $c \to 0, K \to 2$. (5.42) reduces to (5.44).

$$K = \frac{2-c}{1-c} \tag{5.43}$$

$$\frac{f'}{f''} = (1-c)q \tag{5.44}$$

Combining the null and range space steps and from (5.44) we get

$$\begin{bmatrix} Q & A^T & C^T & 0 \\ A & 0 & 0 & 0 \\ C & 0 & 0 & I \\ 0 & 0 & S & V \end{bmatrix} \begin{bmatrix} \Delta x^{zy} \\ \Delta \lambda^{zy} \\ \Delta \nu^{zy} \\ \Delta s^{zy} \end{bmatrix} = - \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ S_k V_k e - c q_k e \end{bmatrix} \tag{5.45}$$

A second order correction on the constraints (5.46) is applied to the step obtained from (5.45)

$$\begin{bmatrix} Q & A^T & C^T & 0 \\ A & 0 & 0 & 0 \\ C & 0 & 0 & I \\ 0 & 0 & S & V \end{bmatrix} \begin{bmatrix} \Delta x^{so} \\ \Delta \lambda^{so} \\ \Delta \nu^{so} \\ \Delta s^{so} \end{bmatrix} = - \begin{bmatrix} 0 \\ 0 \\ 0 \\ \Delta S^{zy} V^{zy} e \end{bmatrix} \tag{5.46}$$

and s and ν are restricted to be nonnegative through a line search. It is important to note that on the right hand side of (5.45) we have obtained the term $cq_k e$, which does the centering. Compare this term with (5.13) with $q_k = \mu$ and $c = \sigma$. Note that for $c = 0$ we get an affine scaling step (5.15). Hence, by an appropriate choice of the objective function, we have a framework which incorporates the characteristics of the concept of centering in interior point methods within an SQP method for the solution of nonlinear programming problems. This procedure could be implemented within a full space or a reduced space method. We expect the strong theoretical and superlinear convergence properties which have been developed for SQP methods to apply in this case as well and some preliminary analysis is described in section 5.5. We will henceforth refer to this framework as ISQP. In the next section we consider some example problems.

5.4 Examples

5.4.1 Linear Programming Problems

A comparison of this approach to that of Mehrotra [72] for the solution of Linear Programs would be of interest. Mehrotra's approach is two-pronged as explained in the previous section. It does not consider the centering term $\sigma\mu_k e$ in the predictor step. In the implementations of this method, the centering parameter is chosen adaptively and incorporated along with the second order correction term in the corrector step (5.16). Testing of the performance of ISQP and Mehrotra's method on the LP test problems from the Netlib directory gave similar results in terms of number of iterations as seen in Table 1. The number of iterations for different values of c for the ISQP approach are also shown, for the same initialization far from the solution. α in the third column corresponds to the fractional line search step taken in the penultimate iteration. In the following section, we consider solution of structured process optimization problems using the ISQP framework.

		ISQP/c values			PC
Problem	Var/Ineq/Eq	.01	.01(1-α)	.005	
recipe	180/275/90	13	13	13	14
sierra	2036/4771/528	24	23	24	24
80bau3b	9799/15047/498	65	57	58	57

Table 5.1: Number of iterations of ISQP and Mehrotra predictor-corrector on LP test problems

5.4.2 Optimal Control and Dynamic Optimization Problems

An important aspect of optimal control and dynamic optimization problems that causes difficulties is the presence of inequality constraints. Inequalities in these problems arise from safety considerations as well as physical limitations upon the control input. In the formulation the number of inequalities grows with the number of time steps. As a result the worst case scenario of an active set approach could considerably slow the solution time. Interior point methods hold promise not only because they tend to converge in a fixed number of iterations, but also because of their structure. Unlike active set methods, interior point formulations have a fixed structure that is sparse and almost block diagonal. Because of this sparse fixed structure and the block diagonal form, sparse linear algebra solvers should be very efficient.

Here, we consider an optimal control problem which demonstrates that the number of iterations in interior point and active set methods could differ by orders of magnitude for a large number of time steps.

Example 1: Illustrative Optimal Control Problem

In the test problems we considered, active set methods generally require a number of iterations that grows superlinearly with the size of the problem, whereas with interior point methods, the number of iterations is constant independent of the number of time steps. As an example, we compare the performance of the following quadratic optimal control problem originally from [1], [11].

$$\begin{aligned}
\min \quad & \int_0^1 (2x_1^2(t) + x_2^2(t) + 6u^2(t))dt + 2x_1^2(1) + x_2^2(1) \\
s.t. \quad & x_1'(t) = x_2(t) \\
& x_2'(t) = -x_1(t) + u(t) \\
& -1 \leq u(t) \leq 1 \\
& x_1(0) = 15 \\
& x_2(0) = 5
\end{aligned} \tag{5.47}$$

This problem has a control profile which is at its bounds for about half of the time horizon as seen in figure 5.1. In figure 5.2, we compare the number of iterations for

ISQP, and two active set based codes, MINOS and QPSOL. As seen from the figure, the number of iterations for MINOS and QPSOL increase with the number of time steps (active constraints), whereas the interior point method keeps at a constant 6 iterations.

In a related work [3], model predictive control of the reactor in the Tennessee Eastman

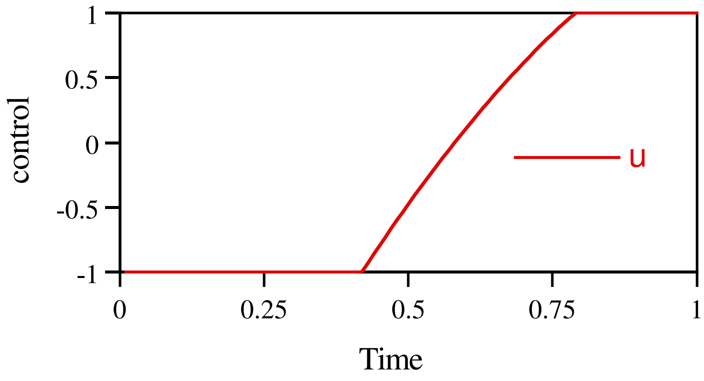

Figure 5.1: Control profile for problem (5.47).

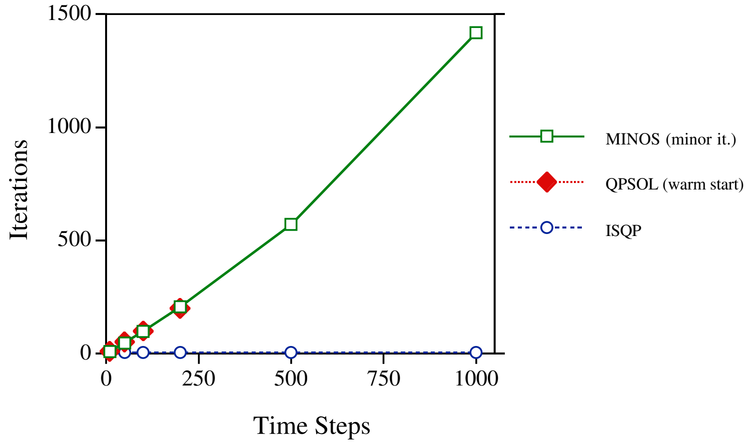

Figure 5.2: Number of iterations for ISQP, MINOS and QPSOL for problem (5.47)

challenge problem [41] was considered. Application of interior point methods on that problem demonstrated considerable savings in computational time as well, as opposed to active set methods.

5.5 Theoretical Properties

Most interior point codes developed in the 1990s have been based on Mehrotra's predictor corrector algorithm. Several variants of this algorithm have been implemented, all quite similar in principle, each differing slightly in the heuristics developed from a decade of computational experience [100]. An intruiging aspect of this class of methods is that they outperform most of the other codes in practice [66], and not every aspect of it has been explained theoretically. An interpertation of the predictor-corrector methods using rSQP allows us to take a fresh look at their theoretical properties from a different persperctive - from that of the global and superlinear convergence properties developed for rSQP over the years. In this section, we attempt a beginning in this direction.

First, we briefly summarize the theoretical properties of the Mehrotra predictor-corrector class of methods currently reported in the literature. The great success of Mehrotra's approach in practice has triggered intense theoretical research for understanding its convergence properties, particularly the computational complexity, in the past five years [95], [105], [104], [102].

Zhang and Zhang [105] studied the theoretical convergence properties of Mehrotra's algorithm on the horizontal linear complementarity problem of which the LP and QP are special cases. Under certain assumptions, including that of monotonicity, they established polynomial complexity bounds for two variants of the predictor-corrector algorithm. The first algorithm, similar to the one Mehrotra [72] proposed, has a worst-case complexity bound of $O(n^3t)$ iterations for reducing the residuals by a factor of 2^{-t} for $t > 1$. Their analysis, however does not consider an adaptive choice of the centering parameter and restricts the iterates with a loose neighborhood of the central path. In the second algorithm, they incorporated the centering term in the predictor step instead of the corrector step. They established a better complexity bound of $O(n^{1.5}t)$ for this algorithm. It should be noted that this approach of the centering term in the predictor is, in fact, very similar to the combined range and null space steps in the ISQP framework described in section 5.3. Zhang and Zhang [102] further extended the theoretical properties of this algorithm to an infeasible predictor-corrector method, where the initial points are not required to be strictly feasible. This algorithm retained the complexity, $O(n^{1.5}t)$, of its feasible counterpart. Under certain conditions the algorithm also possesses a Q-subquadratic convergence. ie. the convergence is Q-order 2 with an unbounded Q factor. For the definitions of these terms, the reader is refered to [39].

Tapia *et al.* [95] studied the Mehrotra predictor-corrector philosophy and demonstrated that it is similar to a level-1 composite Newton philosophy where the corrector step without centering could be viewed as a simplified Newton step. A level-m composite Newton method, which requires evaluation of the Jacobian only once every m steps, is defined as in (5.48).

$$\begin{aligned} &\text{Solve} \quad F'(\Delta x_i) = -F(x_k + \Delta x_0 + \ldots + \Delta x_{i-1}) \text{ for } \Delta x_i,\ i = 0, \ldots, m \\ &\text{Set} \quad x_{k+1} = x_k + \alpha_k(\Delta x_0 + \ldots + \Delta x_m) \end{aligned} \tag{5.48}$$

Under assumptions of nondegeneracy and unit steplength, which are quite restrictive, [95] showed that Mehrotra's predictor-corrector algorithm is cubically convergent. Current research is geared towards showing superlinear convergence properties when these assumptions are relaxed. It should also be mentioned that subquadratic convergence has also been established for affine scaling algorithms which do not use a corrector step [73]. Theoretical explanation of the role of the corrector step in better practical performance is, however, not well understood.

As a way of looking at the Mehrotra class of algorithms from a different angle, the remainder of this section briefly adapts the formulation (5.25) to the analysis carried out in Biegler *et al.* [14] for reduced Hessian SQP methods. The reader is referred to that paper for details and proofs in the original analysis.

The following theorem was proved under the assumptions in [14].

Theorem 5.2 *[14]*

$$\lim_{k\to\infty} \inf \left(\| Z_k^T \bar{c}_k \| + \| \bar{g}_k \|\right) = 0.$$

Lemma 5.3 *Theorem 5.2 guarantees the existence of a solution to (5.25) corresponding to satisfaction of the complementarity condition.*

Proof
$Z_k^T \bar{c}_k \Rightarrow \Delta q = 0$. This also implies that $f' = 0$. Now for a choice of f as in (5.41), $f' = q^{K-1}$, $K > 1$ which implies $q = 0$ at the solution.

Following theorem 5.2, [14] also showed local convergence properties in Lemma 5.3 of that paper, which holds for our framework as well. Of greater interest is the superlinear convergence property.

Theorem 5.4 *[14] Suppose that the iterates generated by the Algorithm converge R-linearly to a point x^*, a local minimizer, and suppose full steps are taken for all large k. If*

$$lim_{k\to\infty} \frac{\parallel B_k p_z + w_k - Z^{T*}\overline{Q}^* x_k \parallel}{\parallel x_k \parallel} = 0 \tag{5.49}$$

where B_k is a BFGS update to $Z_k^T \overline{Q}_k Z_k$ and w_k is a Broyden update to $Z_k^T \overline{Q}_k Y_k p_Y$. Then, the rate of convergence is 1-step Q-superlinear.

For (5.25), $w_k = 0$, $B^* = Z^{T*}\overline{Q}^* Z^*$. It can be shown that the numerator of (5.49) vanishes and the convergence rate is Q-quadratic. However, the step sizes need to be shown to be tending to unity near the solution, for quadratic convergence to be proved. This poses a challenging problem and detailed convergence analysis can be developed as part of future work.

It should be noted that the second order correction suggested in our framework is not included in the analysis in [14]. However, incorporation of the second order correction does not reduce the convergence rates described.

5.6 Conclusions

An intuitive interpretation of the predictor corrector interior point methods using reduced Hessian SQP has been derived in this chapter. A framework for the adaptation of a path following primal dual interior point method with an SQP approach has been also been developed. As demonstrated on the example problems, this results in an efficient methodology for the treatment of inequalities while retaining the ability of SQP methods to take advantage of the problem structure at the same time. Preliminary theoretical analysis of this framework has been outlined using the properties developed in [14]. Directions for future work in interior point methods for process optimization will be described in detail in chapter 7.

Chapter 6

Smoothing Methods for the Treatment of Complementarity Conditions and Nested Discontinuities

Every valley shall be exalted, and every mountain and hill shall be made low; and the crooked shall be made straight, and the rough places plain.

- Isaiah, The Bible

Synopsis

Complementarity is central to all constrained optimization problems. Direct enforcement of complementarity conditions could be problematic because of the inherent nondifferentiability associated with them. In this chapter, we look at a class of smoothing methods for solving the complementarity problem using a continuation algorithm. An equation based formulation for solving the phase equilibrium problem with complementarity conditions modeling appearance and disappearance of phases, is described in this chapter. Smoothing methods are used to solve limiting distillation cases modeled by the equation based formulation. Applicability of smoothing methods to approximate complicated nested discontinuities using simple functions with a single smoothing parameter is also investigated.

6.1 Introduction

Chapter 5 describes the applicability of interior point methods to overcome the computational complexity associated with active set methods in process optimization. In summary, primal-dual interior point methods solve the complementarity conditions (6.1) approximately, where at each iteration an approximation to complementarity, (6.2), is solved for $K > 0$. K is reduced at each iteration until the original problem is solved.

$$\begin{aligned} s_k \nu_k &= 0 \\ s, \nu &\geq 0 \end{aligned} \tag{6.1}$$

$$\begin{aligned} s_k \nu_k &= K \\ s, \nu &> 0 \end{aligned} \tag{6.2}$$

Looking at this exercise from a different angle, interior point methods solve the complementarity problem, which is nonsmooth at the origin, by solving a sequence of smooth approximations to it. From that standpoint, primal - dual interior point methods can be considered as a special case of a larger class of smoothing methods used in process engineering. Our study of interior point methods and its potential to solve a variety of problems motivated us to look at smoothing methods in general and its applicablity in process engineering. In this chapter, we focus our attention on the properties and the applicability of smoothing methods to address the following issues arising in chemical process engineering.

- Complementarity conditions modeling implicit discontinuities
- First-order nondifferentiabilities and nested discontinuities

6.2 Smoothing Functions

Consider a general complementarity and nonnegativity conditions of the form (6.1). (6.1) could be represented by the following notation

$$0 \leq s \perp \nu \geq 0 \tag{6.3}$$

(6.3) could be expressed equivalently as (6.4).

$$s = (s - \nu)_+ \tag{6.4}$$

where the + function is defined as

$$x_+ = max(0, x) \tag{6.5}$$

Let us examine the relative merits and demerits of expressing the complementarity conditons in the form (6.4). The nonnegativity conditions are implicitly enforced within (6.4), hence no additional procedure as a line search in primal-dual interior point methods described in chapter 6 is required as a consequence. However, the + function is nondifferentiable as it involves the nonsmooth *max* operator. Chen and Mangasarian [33] describe a class of smoothing methods for the treatment of complementarity conditions, which we briefly outline below.

6.2.1 Properties

Consider a piecewise continous distribution function $d(x)$ which satisfies the properties (6.6).

$$d(x) \geq 0 \qquad and \qquad \int_{-\infty}^{+\infty} d(x)dx = 1 \tag{6.6}$$

The density function $d(x)$ can be used to derive smooth approximations $\hat{t}$ to the Dirac Delta function $\delta(x)$ (6.7), $\hat{s}$ to the step function $\sigma(x)$ (6.8) and $\hat{p}$ to the plus function $(x)_+$ (6.9). β is a small positive parameter in these equations.

$$\hat{t}(x,\beta) = \frac{1}{\beta} d(\frac{x}{\beta}) \approx \delta(x) \tag{6.7}$$

$$\hat{s}(x,\beta) = \int_{-\infty}^{x} \hat{t}(t,\beta)dt \approx \sigma(x) \tag{6.8}$$

$$\hat{p}(x,\beta) = \int_{-\infty}^{x} \hat{s}(t,\beta)dt \approx (x)_+ \tag{6.9}$$

The smooth approximations approach the nonsmooth functions as $\beta \to 0$. The smooth approximation to the complementarity condition (6.4) is given by (6.10).

$$s - \hat{p}(s - \nu, \beta) = 0 \tag{6.10}$$

Below we look at some of the key theoretical properties of the smooth formulation. Lemma 6.1 and proposition 6.2 look at properties of the smooth function $\hat{p}(x,\beta)$. They also establish error bounds on the residual of the smooth approximation of the complementarity condition. The proofs could be obtained from the original references.

Lemma 6.1 *[33] Let $d(x)$ be a probability density function which satisfies the following:*
(A1) *$d(x)$ is piecewise continuous with finite number of pieces and satisfies (6.6).*
(A2') *$E[|x|]_{d(x)} = \int_{-\infty}^{+\infty} |x|d(x)dx < +\infty$.*
$\hat{t}(x,\beta)$ is defined as in (6.7). Then

- ***(P1)*** *$\hat{p}(x,\beta)$ is continuously differentiable*
- ***(P2')*** *$-D_2\beta \leq \hat{p}(x,\beta) - (x)_+ \leq D_1\beta$ where*

$$D_1 = \int_{-\infty}^{0} |x|d(x)dx$$
$$D_2 = max\{\int_{-\infty}^{+\infty} xd(x)dx, 0\} \tag{6.11}$$

- ***(P3')*** *$\hat{p}(x,\beta)$ is nondecreasing and convex.*
- *If the support*
  (A3) *supp $d(x) = \Re$*
  *then **(P3)** $\hat{p}(x,\beta)$ is strictly increasing and strictly convex.*
- ***(P5)*** *$\| x - (x - F(x))_+ \|_p \leq \| x - \hat{p}(x - F(x),\beta) \|_p + \gamma_p \max\{D_1, D_2\}\beta$*
  where γ_p is a constant based on the vector norm.

Property (P5) shows that the residual of the nonlinear complementarity problem is bounded by that of the smooth equation. Chen and Harker [32] introduced additional conditions (A2), (A4) on $d(x)$ and identified a subclass of these functions which satisfy properties (P2) and (P4).

(A2) $\lim_{x\to\infty} x^3 d(x) < \infty$ and $\lim_{x\to\infty} x^3 d(-x) < \infty$.
(A4) $\int_{-\infty}^{+\infty} xd(x)dx = 0$

(P2) $0 \leq \hat{p}(x,\beta) - (x)_+ \leq b(\beta)$ where $b(0) = 0$ and $\lim_{\beta\to\infty} b(\beta) = \infty$.

(P4) $\lim_{x\to\infty}(\hat{p}(x,\beta)+\hat{p}(-x,\beta)-x)x<\infty \qquad 0<\beta<\infty$

[32] showed that $(A2) \Rightarrow (A2')$. Also (P2) is (P2') with $D_2 = 0$ and $b(\beta) = D_1\beta$. [32] also showed that any $\hat{p}(x,\beta)$ satisfying properties (P2) - (P4) can be described in terms of a density function satisfying (6.6). Proposition 6.2 shows the boundedness and uniqueness of the solution trajectory.

Proposition 6.2 *[32] Let the function $\hat{p}(x,\beta)$ satisfy (P2)-(P4). Then*

- *Let $x=\hat{p}(x-y,\beta)$ for some $0\le\beta<\infty$. Then*
  $x\ge 0$, $y\ge 0$, $xy\le B$, for some $0\le B<\infty$.
- *Given any $x\ge 0$, $y\ge 0$, $\exists$ a unique $\beta\ge 0$ such that $x=\hat{p}(x-y,\beta)$.*

6.2.2 Examples

In section 6.2.1 we have briefly outlined some of the key theoretical properties of smooth approximations to the complementarity problem, derived from distribution functions. In this section we consider two specific classes of smoothing functions which could be described in terms of distribution functions satisfying (6.6). They will be used to solve example problems in the subsequent sections.

Sigmoidal function from Neural Networks

The neural networks smooth function with the distribution function (6.12) is analyzed in detail in Chen and Mangasarian [33]. For this function $D_1 = \log 2$ and $D_2 = 0$ in (6.11) and the smooth plus function is given by (6.13).

$$d(x) = \frac{e^{-x}}{(1+e^{-x})^2} \tag{6.12}$$

$$max(0,x) \approx \hat{p}(x,\beta) = x+\beta\ln(1+e^{-\frac{x}{\beta}}) \tag{6.13}$$

Theorem 6.3 establishes an arbitrarily accurate solution to the smooth nonlinear equation arising from the sigmoidal function for any solvable linear, nonlinear or mixed complementarity problem.

Theorem 6.3 *[33] If $x, y \in R$ satisfy*

$$-\frac{\delta_1}{\beta} \leq x - p(x-y,\beta) \leq 0 \tag{6.14}$$

where $\delta_1 \geq 0$. Then

$$(-x)_+ \leq \delta_1\beta, \qquad (-y)_+ \leq \delta_1\beta,$$
$$(xy)_+ \leq \max\{2, (e^{\delta_1}-1)\log^2(e^{\delta_1}-1)\}\beta^2$$

For all $\delta_1 \geq D_1$, (6.14) has a solution for $\beta > 0$. Hence a solution of this system of inequalities is an approximate solution to the complementarity problem.

Interior Point Smooth Approximation

Consider the smooth approximation by Balakrishna and Biegler [7]. Here

$$d(x) = \frac{1}{2(x^2+1)^{3/2}} \tag{6.15}$$

$$max(0,x) \approx \hat{p}(x,\beta) = \frac{x+(x^2+\beta^2)^{1/2}}{2} \tag{6.16}$$

For this function $D_1 = \frac{1}{2}$ and $D_2 = 0$. This smooth function is similar in form with those described in [89], [61] and [32]. It can be shown on the lines of [61] that the smooth approximation to the complementarity condition (6.17) will be satisfied *iff* (6.18) is satisfied and hence the reason for calling it the *interior point* smooth approximation.

$$x = \frac{x-y+((x-y)^2+\beta^2)^{1/2}}{2} \tag{6.17}$$

$$x > 0, \qquad y > 0, \qquad xy = \beta^2/4 \tag{6.18}$$

6.2.3 Implementation

We consider two algorithms for the implementation of the concepts described above.

A continuation method

A continuation method reduces the parameter β in the smooth equations gradually till the original complementarity problem is solved within a certain error criteria. A general framework for the implementation of a continuation method is as follows [33], [32].

Algorithm 6.4 *Let* $\mathbf{r}(\mathbf{x^i})$ *be the residual of the original complementarity problem and* $\mathbf{R}(\mathbf{x^i}, \beta^{\mathbf{i}})$ *be that of the smoothed equations.* K, ϵ *are some fixed constants and* $\eta \in (0, 1)$.
$i = 0$

1. If $\| \mathbf{r}(\mathbf{x^i}) \| \leq \epsilon$, then $\mathbf{x^i}$ is an approximate solution to the complementarity problem.
2. Determine search direction $\Delta \mathbf{x^i}$ from the smooth equation and find next point $\mathbf{x^{i+1}}$ using an equation solving technique (for example, Newton)
3. If $\| \mathbf{R}(\mathbf{x^{i+1}}, \beta^{\mathbf{i}}) \| \leq \mathbf{K} \| \beta^{\mathbf{i}} \|$, then
   $\beta^{i+1} = \eta\beta^i$. else, $\beta^{i+1} = \eta\beta^i$.
   $i = i + 1$. Goto 1.

Modified framework

In the modified framework, an approximate solution of the smooth equations for a fixed value of β is determined. Using that as initial point, β is reduced until the original set of equations are satisfied with a certain error. Though a continuation method is expected to give a better performance, the modified framework was considered for ease of implementation within modeling languages like GAMS. The error criteria for the termination of the subproblem for a fixed β is set proportional to β. This results in better computational efficiency as well as better treatment of non-strict complementarity problems, which arises in example problems in section 6.4.4. In case of non-strict complementarity, where both the terms in the complementarity goes to zero, the smoothed equations (6.10) cannot be satisfied exactly for nonzero values of β. An approximate solution at nonzero β overcomes this problem.

6.3 Complementarity Conditions

In section 6.2 we described how smoothing methods could be used to reformulate and solve complementarity conditions. An efficient method for the solution of the complementarity problem has wide implications in process engineering. [47] describes a broad variety of applications of complementarity problems in engineering and economics.

Complementarity is central to all constrained optimization problems. Chemical process industries perform (constrained) nonlinear programming as part and parcel of their day to day synthesis, design, optimization and control activities. A variety of implementations based on Successive Quadratic Programming (SQP) [14] and reduced gradient methods [75] have been developed for the solution of the nonlinear programming problems. A problem which has often been of interest in these implementations have been the efficient treatment of the complementarity conditions arising from the direct solution of the Karush-Kuhn-Tucker (KKT) conditions of the optimization problem. Direct enforcement of the complementarity conditions could be problematic because of the inherent nondifferentiability and singularities associated with these functions. Consider a general nonlinear programming problem of the form (6.19).

$$\begin{aligned} Min \quad & f(x) \\ s.t. \quad & g(x) = 0 \\ & h(x) \geq 0 \end{aligned} \tag{6.19}$$

$x \in \Re^n$, $g \in \Re^m$, $h \in \Re^r$. The KKT conditions of (6.19) is given by (6.20). $\lambda \in \Re^m$, $v \in \Re^r$.

$$\begin{aligned} \nabla f(x) - \nabla g(x)\lambda - \nabla h(x)\nu &= 0 \\ g(x) &= 0 \\ h(x) &\geq 0 \\ \nu_k h_k(x) &= 0 \qquad k = 1,..,r \\ \nu &\geq 0 \end{aligned} \tag{6.20}$$

Direct solution of the KKT conditions requires an efficient way of handling the complementarity conditions $\nu_k h_k = 0$ and the nonnegativity constraints $\nu, h(x) \geq 0$. In chapter 5, we described an interior point method, which could be considered as a special class of smoothing methods, for the solution of the complementarity problem in the solution of the KKT conditions.

The focus of this section will be to formulate and solve the complementarity conditions modeling implicit discontinuities outlined in section 4.4. The description in section 6.2 shows that a reformulation of the complementarity constraints along with a smoothing technique forms an efficient framework for the solution of this problem, while taking care of the nonnegativity constraints implicitly. In the following section, we describe the applicability of smoothing methods to solution of flash and distillation models.

6.4 Flash and Distillation Models

Section 4.4 describes phase equilibrium as an important class of implicit discontinuities occuring in process simulation. Implicit discontinuities could be described using complementarity conditions and in this section, we will pose the phase equilibrium problem as a complementarity problem and solve it using smoothing techniques.

6.4.1 Phase Equilibrium Problems with Phase Changes

To recap from section 4.4, one of the basic problems associated with phase equilibrium calculations in an equation oriented simulation environment is that the number of phases is not known *a priori*. Different sets of equations are valid depending on the number of phases present at equilibrium. For example, in a two phase flash, the equilibrium equations are generally not valid in either of the single phase regions. The classical sequential modular approaches first determine the number and nature of the phases present at equilibrium and then solve the corresponding set of equations. Such a *procedure based* approach is not viable in the larger context of an equation oriented environment.

In chapter 4 and in [53], we addressed this problem for multiphase systems using an optimization formulation. The key idea used in the formulation (4.24) is that if two phases do not coexist at equilibrium, then the corresponding equilibrium expression could be relaxed. This was done by relaxing the equilibrium constant K_i^p in the equilibrium expression (6.21) by introducing the variable γ_p. Here, x_i^p is the mole fraction of component i in phase p and p_{ref} is a reference phase.

$$x_i^{p_{ref}} - \gamma_p K_i^p x_i^p = 0 \qquad p = 1, n_p, \quad p \neq p_{ref} \tag{6.21}$$

$\delta_p \geq 0$ is defined as the deviation of γ_p from unity through (6.22) and the objective function of the optimization formulation is given by $\sum_p \delta_p$.

$$\begin{aligned} \delta_p &\geq \gamma_p - 1 && p = 1, n_p, \quad p \neq p_{ref} \\ \delta_p &\geq 1 - \gamma_p && p = 1, n_p, \quad p \neq p_{ref} \end{aligned} \tag{6.22}$$

This formulation is shown to be equivalent to the KKT conditions of a Gibbs free energy minimization formulation, in Appendix B.

For the two phase problem, the equilibrium expression is given by (6.23) where δ, the objective function, is defined using the inequalities (6.24).

$$y_i - \gamma K_i(P, T, x) x_i = 0 \qquad i = 1, n \tag{6.23}$$

$$\begin{aligned} \delta &\geq \gamma - 1 \\ \delta &\geq 1 - \gamma \end{aligned} \tag{6.24}$$

In section 4.4.2, the optimization formulations (4.24) and (4.25) were successfuly tested on dynamic simulation example problems from LLV and LV systems where the number of phases changes with time.

Further research on this problem was to find a simplified equation based system equivalent to the optimization formulation. *ie.* we would like to have an equation based formulation for modeling the phase equilibrium problem, in the event of no prior knowledge of the number of phases present at equilibrium or in the likelihood of the phase change in a dynamic problem. The motivation for solving this problem comes from the following:

- Phase equilibrium problems, an implicit nonsmooth discontinuity, can be described as an equation based complementarity problem. However, can the KKT conditions of (4.24) be simplified?
- Need to enhance currently used equation based models for phase equilibrium calculations in process simulators with the capability of addressing the issue of phase transitions.

Equation based models used in most of the commercial simulators are formulated for a certain prespecified number and nature of phases. Industrial experiences with these models often report a failure when there is a change in the phase combinations present. Classical examples are dry or vaporless trays present in two phase distillation columns. Two phase models are unable to describe such cases and lead to a failure of the simulation or optimization exercise. Currently, developers are looking for simple ways of addressing the issue of phase transitions by modifying the existing models, but at the same time retaining the core of it.

We will address the above issues in the following sections, for two and three phase systems.

6.4.2 Two Phase Formulation

In this section we will develop an equation based formulation for the two phase system. The following analysis develops the equation based formulation from the results obtained from the optimization formulation.

(6.24) can be written as (6.25) by introducing slacks s_+, s_- to convert them to equalities.

$$\begin{aligned} -\delta + \gamma - 1 + s_+ &= 0 \\ -\delta - \gamma + 1 + s_- &= 0 \end{aligned} \tag{6.25}$$

The optimality conditions of (4.25) could be expressed equivalently as the set of equations (6.26 - 6.36).

$$F - L - V = 0 \tag{6.26}$$

$$Fz_i - Lx_i - Vy_i = 0 \qquad i = 1, n \tag{6.27}$$

$$y_i - \gamma K_i(P, T, x)x_i = 0 \qquad i = 1, n \tag{6.28}$$

$$\sum_i y_i - \sum_i x_i = 0 \tag{6.29}$$

$$-\delta + \gamma - 1 + s_+ = 0 \tag{6.30}$$

$$-\delta - \gamma + 1 + s_- = 0 \tag{6.31}$$

$$s_+L = 0 \tag{6.32}$$

$$s_-V = 0 \tag{6.33}$$

$$\delta, s_+, s_- \geq 0 \tag{6.34}$$

$$0 \leq x_i, y_i \leq 1 \tag{6.35}$$

$$0 \leq L, V \leq F \tag{6.36}$$

The following correlations for the values of γ to the existence of phases as derived in Appendix B. In Appendix D, we show that (6.26 - 6.36) is equivalent to these correlations.

- $\gamma = 1 \Rightarrow$ L,V > 0
- $\gamma > 1 \Rightarrow$ V $= 0$
- $\gamma < 1 \Rightarrow$ L $= 0$

The key constraints are the complementarity conditions (6.32 - 6.33). They could be derived intuitively as follows. Eliminate δ from (6.30) and (6.31) and replace them by (6.37).

$$\begin{aligned} \gamma - 1 &= (s_- - s_+)/2 \\ s_-s_+ &= 0 \end{aligned} \tag{6.37}$$

The complementarity condition $s_-s_+ = 0$ need not be explicitly included in the formulation as it is automatically satisfied. The only case where it is violated is when $L = V = 0$ in which case the mass balances are violated for a nonzero feed. Now, from (6.37), when $\gamma > 1$, $s_- > 0$, $s_+ = 0$. But $\gamma > 1$ corresponds to the case $V = 0$. Also, when $\gamma < 1$, $s_- = 0$, $s_+ > 0$ which corresponds to $L = 0$. From the above, we can see that the pairs (s_-, V) and (s_+, L) are complementary. The final formulation by eliminating δ is given by (6.38-6.47).

$$F - L - V = 0 \tag{6.38}$$

$$Fz_i - Lx_i - Vy_i = 0 \qquad i = 1, n \tag{6.39}$$

$$y_i - \gamma K_i(P, T, x)x_i = 0 \qquad i = 1, n \tag{6.40}$$

$$\sum_i y_i - \sum_i x_i = 0 \tag{6.41}$$

$$\gamma - 1 = s_- - s_+ \tag{6.42}$$

$$s_+ L = 0 \tag{6.43}$$

$$s_- V = 0 \tag{6.44}$$

$$s_+, s_- \geq 0 \tag{6.45}$$

$$0 \leq x_i, y_i \leq 1 \tag{6.46}$$

$$0 \leq L, V \leq F \tag{6.47}$$

The methodologies for the efficient treatment of complementarity conditions described in section 6.2 can be used for the solution of the system of equations (6.38 - 6.47).

An equation based formulation as (6.38 - 6.47) where the relevant set of equations automatically fall into place regardless of whether the solution lies in the single phase or two phase regions, is helpful in dealing with these transitions with no additional procedures in the larger context of equation based process and plant simulation and optimization. A typical equation oriented simulator or optimizer will have flash models described by (6.38-6.41, 6.46-6.47) (γ fixed at 1). Addition of just three equations (6.42-6.44) and three variables γ, s_-, s_+ gives it the additional capability of dealing with operation in the single phase regions. The basic flash model which is already in place, is nevertheless utilized. In the following section, we describe an equation based model for the three phase problem.

6.4.3 Three Phase Formulation

The formulation for the three phase problem analogous to (6.26- 6.36) for two phase case is given by (6.48- 6.65).

$$F - L^I - L^{II} - V = 0 \tag{6.48}$$

$$Fz_i - L^I x_i^I - L^{II} x_i^{II} - V y_i = 0 \qquad i = 1, n \tag{6.49}$$

$$y_i - \gamma^I K_i^I(P, T, x) x_i^I = 0 \qquad i = 1, n \tag{6.50}$$

$$y_i - \gamma^{II} K_i^{II}(P, T, x) x_i^{II} = 0 \qquad i = 1, n \tag{6.51}$$

$$\sum_i y_i - \sum_i x_i^I = 0 \tag{6.52}$$

$$\sum_i y_i - \sum_i x_i^{II} = 0 \tag{6.53}$$

$$\gamma^I - 1 = s_-^I - s_+^I \tag{6.54}$$

$$\gamma^{II} - 1 = s_-^{II} - s_+^{II} \tag{6.55}$$

$$s_-^I - s_-^{II} + \theta^I - \theta^{II} = 0 \tag{6.56}$$

$$(s_+^I + \theta^I)L^I = 0 \tag{6.57}$$

$$(s_+^{II} + \theta^{II})L^{II} = 0 \tag{6.58}$$

$$(s_-^I + s_-^{II})V = 0 \tag{6.59}$$

$$s_-^I s_+^I = 0 \tag{6.60}$$

$$s_-^{II} s_+^{II} = 0 \tag{6.61}$$

$$\theta^I \theta^{II} = 0 \tag{6.62}$$

$$\delta^k, s_+^k, s_-^k, \theta^k \geq 0 \tag{6.63}$$

$$0 \leq x_i, y_i \leq 1 \tag{6.64}$$

$$0 \leq L^I, L^{II}, V \leq F \tag{6.65}$$

In Appendix E we show that the above formulation is equivalent to the conditions derived for the existence of different phases in Appendix B for the optimization formulation.

6.4.4 Examples

In this section, first, we consider limiting distillation example problems for the two phase case. To cite an instance, in a two phase distillation column, there could exist operating conditions which results in dry or vaporless trays. In the following examples, we consider such cases. The equation based formulations presented in the previous section will be used to model phase equilibrium for each tray. The complementarity conditions are smoothed using the interior point smoothing function (6.16). The distillation column models are formulated in GAMS [22] and solved using CONOPT [42] using the modified framework described in section 6.2.3. Modified framework was chosen because of its ease of implementation within GAMS. A three phase flash example is also described. The computational time reported for all the examples are on an HP 9000/700.

Example 1: 2 component 2-phase ideal column operating under minimum reflux

In this example we consider a 25-tray distillation column with a saturated liquid feed of 70% benzene - 30% toluene to tray 7, as described by [26]. Ideal thermodynamics is used to model phase equilibrium. Table 6.1 shows the solution of the column under varying

Reflux ratio	Dry tray location	Iterations	Time(s)
0.1	-	7	0.36
0.01	-	7	0.49
0.001	8	54	5.74
0.0005	8	56	5.44

Table 6.1: Solution of ideal 2 phase distillation column with smoothing methods

reflux ratios using smoothing methods. For all the cases, the problems was intialized in the two phase region. For reflux ratios of 0.001 and 0.0005, trays above the feed tray dry up and the algorithm was able to capture this phenomena successfully.

Example 2: 5 component 2-phase nonideal column operating under minimum reflux

The second example is a 17-tray nonideal distillation column with a saturated liquid feed of 15 % methanol - 40 % acetone - 5 % methacrylate - 20 % benzene - 20 % chloroform to tray 12 [26]. The liquid phase is modeled with UNIQUAC and virial equations are used to model the vapor phase. In table 6.2, we show the number of iterations for solving the column for different reflux ratios, for this highly nonlinear model. In this example a different initialization was considered: all trays above the feed tray were initialized to have only vapor phase corresponding to an approximate solution for a reflux ratio of 0.00005. As seen from table 6.2, algorithm based on smoothing methods were able to obtain solutions corresponding to two phases in all trays, as well as dry trays above the feed location, with the same initialization.

Reflux ratio	Iterations	Time(s)	Dry trays present?
0.00004	7	7.77	Yes
0.001	15	25.43	No
0.1	14	23.32	No
1.0	21	28.76	No

Table 6.2: Solution of nonideal 2 phase distillation column with smoothing methods

Example 3: 3-phase flash

Here we consider a water - acrylonitrile - acetonitrile system originally described by Wu and Bishnoi [101]. The flash calculations were done at a temperature of 333 K and a feed composition of 0.6, 0.35 and 0.05 respectively for water, acrylonitrile and acetonitrile. The three phase formulation (6.48 - 6.65) was used to solved the problem with the liquid phases modeled using UNIQUAC. Three different phase combinations were observed in a

P (bar)	Iterations	Time(s)	Phases at solution
0.65	30	0.37	L_1V
0.675	12	0.21	L_1L_2V
0.69	33	0.43	L_1L_2

Table 6.3: Solution of 3 phase flash with smoothing methods

pressure range of 0.04 bar as seen in table 6.3. All the problems were initialized in the three phase region.

Remarks

The number of iterations reported for the above examples are using the solver CONOPT in GAMS through the implementation of a modified continuation framework. However, it should be noted that an implementation of a continuation algorithm as in Algorithm 6.4 along with a more efficient nonlinear equation solver is expected to show a decrease in the number of iterations reported in this section for the problems considered.

6.5 Representation of Nondifferentiabilities Using Smoothing Functions

Functions with discontinuous derivatives or *kinks* occur frequently in process problems in chemical engineering. Typical example are equations of the form (6.66).

$$g(x, \max_j (f_j(x))) = 0 \tag{6.66}$$

Several reformulation strategies of the nondifferentiabilities have been suggested to solve problems of the form (6.66). The smoothing approximations could be used to deal with such nondifferentiable functions appearing in engineering applications. Duran [44] used a smoothing approximation for approximating functions of the form $\max(0, f(x))$. Below we consider the applicabililty of the sigmoidal neural network smoothing function described in section 6.2.2. The advantage is that the sigmoidal function gives compact closed formulations for a wide variety of these functions which are easily implementable. First, we consider two commonly used functions, the absolute value operator and the max of n functions.

6.5.1 Absolute Value Operator

Let

$$y_{abs} = abs(x) \tag{6.67}$$

Define

$$\alpha = \frac{1}{\beta} \tag{6.68}$$

From (6.67),

$$\begin{aligned} y_{abs} &= max(x, -x) \\ &= max(0, x) + max(0, -x) \\ &\approx x + \frac{1}{\alpha}\ln(1 + e^{-\alpha x}) - x + \frac{1}{\alpha}\ln(1 + e^{\alpha x}) \\ &= \frac{1}{\alpha}\ln(2 + e^{\alpha x} + e^{-\alpha x}) \end{aligned} \tag{6.69}$$

6.5.2 Max of N Functions

Let

$$y_{max} = \max(f_1, f_2, \ldots, f_k, \ldots, f_n) \tag{6.70}$$

The above function could have many nondifferentiable points unlike just one nondifferentiable point, the origin, as in (6.5) and (6.67). However, a smooth approximation could be derived for the above max function based on just one smoothing parameter.

Proposition 6.5 *A smooth approximation to (6.70) is*

$$\max(f_1, f_2, \ldots, f_k, \ldots, f_n) \approx f_k + \frac{1}{\alpha}\ln(1 + \sum_{i=1, i\neq k}^{n} e^{-\alpha(f_k - f_i)}) \quad \text{for any } 1 \leq k \leq n \tag{6.71}$$

Proof

This relation could be shown by induction.

- $Max(f_1, f_2)$

$$\begin{aligned} \max(f_1, f_2) &= f_1 + \max(0, f_2 - f_1) \\ &\approx f_1 + (f_2 - f_1) + \frac{1}{\alpha}\ln(1 + e^{-\alpha(f_2 - f_1)}) \\ &= f_2 + \frac{1}{\alpha}\ln(1 + e^{-\alpha(f_2 - f_1)}) \end{aligned} \tag{6.72}$$

- $Max(f_1, f_2, f_3)$

$$\begin{aligned} \max(f_1, f_2, f_3) &= f_1 + \max\left[0, f_2 - f_1, f_3 - f_1\right] \\ &= f_1 + \max\left[\max(0, f_2 - f_1), f_3 - f_1\right] \\ &\overset{(6.13)}{\approx} f_1 + \max\left[f_2 - f_1 + \frac{1}{\alpha}\ln(1 + e^{-\alpha(f_2 - f_1)}), f_3 - f_1\right] \\ &\overset{(6.72)}{\approx} f_1 + (f_3 - f_1) + \frac{1}{\alpha}\ln\left[1 + e^{-\alpha\left(f_3 - f_2 - \frac{1}{\alpha}\ln(1 + e^{-\alpha(f_2 - f_1)})\right)}\right] \\ &= f_3 + \frac{1}{\alpha}\ln\left[1 + e^{-\alpha(f_3 - f_2)}\left(1 + e^{-\alpha(f_2 - f_1)}\right)\right] \\ &= f_3 + \frac{1}{\alpha}\ln\left[1 + e^{-\alpha(f_3 - f_2)} + e^{-\alpha(f_3 - f_1)}\right] \end{aligned} \tag{6.73}$$

- To show if

$$\max(f_1, f_2, \ldots, f_k, \ldots, f_j) = f_k + \frac{1}{\alpha}\ln(1 + \sum_{i=1, i\neq k}^{j} e^{-\alpha(f_k - f_i)}) \tag{6.74}$$

then,

$$\max(f_1, f_2, \ldots, f_k, \ldots, f_{j+1}) = f_k + \frac{1}{\alpha}\ln(1 + \sum_{i=1, i\neq k}^{j+1} e^{-\alpha(f_k - f_i)})$$

$$\begin{aligned}
& \max(f_1, f_2, \ldots, f_k, \ldots, f_{j+1}) \\
= \quad & f_1 + \max\left[0, f_2 - f_1, \ldots, f_k - f_1, \ldots, f_j - f_1, f_{j+1} - f_1\right] \\
\overset{from(6.13)}{\approx} \quad & f_1 + \max\left[f_2 - f_1 + \frac{1}{\alpha}\ln(1 + e^{-\alpha(f_2 - f_1)}), \ldots, f_k - f_1, \ldots, f_j - f_1, f_{j+1} - f_1\right] \\
\overset{from(6.74)}{\approx} \quad & f_1 + (f_k - f_1) + \\
& \frac{1}{\alpha}\ln\left[1 + e^{-\alpha\left(f_k - f_2 - \frac{1}{\alpha}\ln(1+e^{-\alpha(f_2-f_1)})\right)} + \ldots + e^{-\alpha(f_k - f_j)} + e^{-\alpha(f_k - f_{j+1})}\right] \\
= \quad & f_k + \frac{1}{\alpha}\ln(1 + \sum_{i=1, i\neq k}^{j+1} e^{-\alpha(f_k - f_i)}) \qquad (6.75)
\end{aligned}$$

□

- The absolute value operator (6.67) could also be expressed using (6.72) as

$$y_{abs} = abs(x) \approx x + \frac{1}{\alpha}\ln(1 + e^{-2\alpha x}) \tag{6.76}$$

The approximation error in (6.76) is lesser than in (6.69). (For example the error at the origin is $\frac{\ln 2}{\alpha}$ in (6.76) as opposed to $\frac{\ln 4}{\alpha}$ in (6.69)).

- **Min of n functions**

can also be expressed similarly.

$$\begin{aligned}
\min(f_1, f_2, \ldots, f_k, \ldots, f_n) \quad & = \quad -\max(-f_1, -f_2, \ldots, -f_k, \ldots, -f_n) \\
& \overset{(6.71)}{\approx} \quad -\left(-f_k + \frac{1}{\alpha}\ln(1 + \sum_{i=1, i\neq k}^{n} e^{-\alpha(-f_k + f_i)})\right) \\
& = \quad f_k - \frac{1}{\alpha}\ln(1 + \sum_{i=1, i\neq k}^{n} e^{-\alpha(f_i + f_k)}) \qquad (6.77)
\end{aligned}$$

6.5.3 Nested Min-Max Functions

Nested min-max functions, which are otherwise complicated to represent within an equation oriented framework, can also be approximated using the above smooth formulation, using fairly simple functions.
Consider the following example.

- $\max(f_1, f_2, \ldots, f_k, \ldots, f_n, \min(f_{n+1}, \ldots, f_m))$

$$
\begin{aligned}
& \max(f_1, f_2, \ldots, f_k, \ldots, f_n, \min(f_{n+1}, \ldots, f_m)) \\
\approx\; & \max\left(f_1, f_2, \ldots, f_k, \ldots, f_n, f_K - \frac{1}{\alpha}\ln(1 + \sum_{i=n+1, i\neq K}^{m} e^{-\alpha(f_i + f_k)})\right) \\
& \qquad n+1 \leq K \leq m \\
\approx\; & f_k + \frac{1}{\alpha}\ln\left(1 + \sum_{j=1, j\neq k}^{n} e^{-\alpha(f_k - f_j)} + e^{-\alpha\left(f_k - f_K + \frac{1}{\alpha}\ln(1 + \sum_{i=n+1, i\neq K}^{m} e^{-\alpha(f_i - f_K)})\right)}\right) \\
& \qquad 1 \leq k \leq n, n+1 \leq K \leq m \\
=\; & f_k + \frac{1}{\alpha}\ln\left(1 + \sum_{j=1, j\neq k}^{n} e^{-\alpha(f_k - f_j)} + \frac{e^{-\alpha(f_k - f_K)}}{1 + \sum_{i=n+1, i\neq K}^{m} e^{-\alpha(f_i - f_K)}}\right) \\
& \qquad 1 \leq k \leq n, n+1 \leq K \leq m
\end{aligned}
$$

6.5.4 Examples

To demonstrate the applicability of the smooth approximations to discontinuous functions, the following example problem from Bullard and Biegler [24] was solved.

Pipeline Network Problem

The pipeline network in figure 6.1 has 22 nodes, 38 pipes and 5 checkvalves. The set of equations (6.78) describe the pipeline network.

$$
\begin{aligned}
P_i - P_j = H_{ij} \qquad & \forall \text{pipes} \quad i, j \\
KQ_{ij}^2 = \max(0, H_{ij}) \qquad & \forall \text{pipes} \quad i, j \quad \text{with} \quad \text{valves} \qquad (6.78a) \\
H_{ij} = K sign(Q_{ij}) Q_{ij}^2 \qquad & \forall \text{pipes} \quad i, j \quad \text{with} \quad \text{no} \quad \text{valves} \qquad (6.78a) \\
\sum_j Q_{ij} + \sum_j Q_{ji} = w_i \qquad & \forall \text{nodes} \quad i
\end{aligned} \tag{6.78}
$$

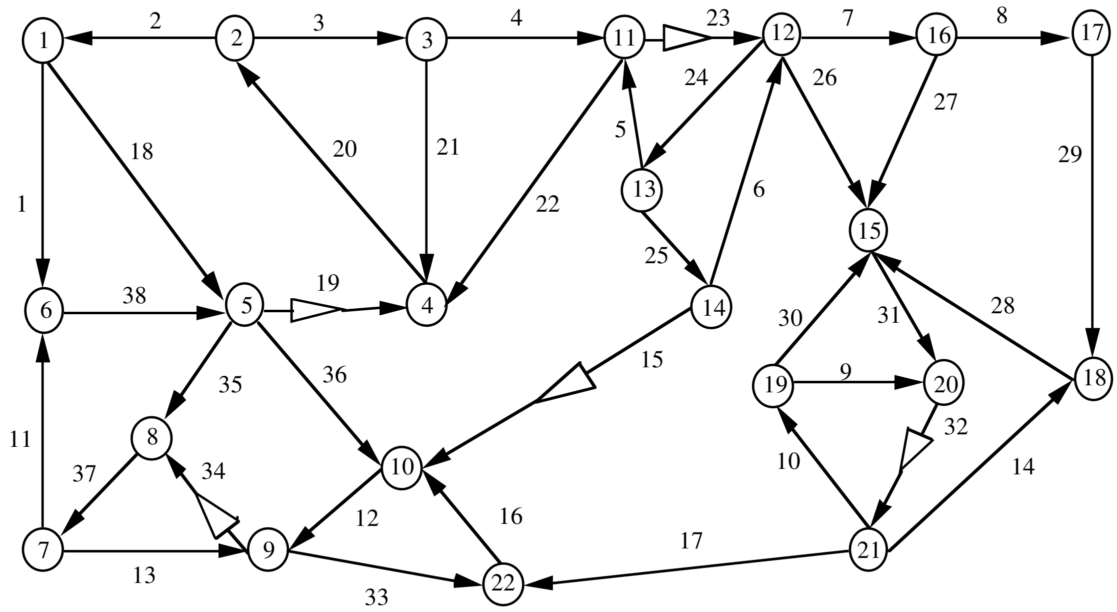

Figure 6.1: Pipeline network with checkvalves

The presence of the checkvalves makes the solution of the problem difficult mainly because of the *open/closed* nature of the valves which introduces nondifferentiabilities. Traditional methods for solving these problems involves and *if-then-else* iterative procedure which solves the problem assuming certain valve positions and resolves them with different valve positions if any conditionals are violated. This procedure could be cumbersome and could fail to converge for large systems. Bullard and Biegler [25] introduced continuous and mixed integer formulations which handles switching of valve positions through inequality constraints. This approach is elegant and more appropriate for an optimization framework. However, in a pure equation solving environment, one would rather not introduce inequality constraints, if they can be avoided. Smoothing methods could be used to achieve this objective.

(6.78a) could be approximated using (6.13) as

$$KQ_{ij}^2 = H_{ij} + \frac{1}{\alpha}\ln(1 + e^{\alpha H_{ij}}) \tag{6.79}$$

(6.78b) has a derivative discontinuity for the second derivative at $Q_{ij} = 0$. For methods using second order information, it could be approximated as

$$H_{ij} = K sign(Q_{ij})Q_{ij}^2 = KQ_{ij}abs(Q_{ij}) \approx KQ_{ij}[Q_{ij} + \frac{1}{\alpha}\ln(1 + e^{-2\alpha Q_{ij}})] \qquad (6.80)$$

The data and the listing of the initial guesses could be found in Bullard and Biegler [25]. All the valves were specified as open as initial conditions, though valves in pipes 15, 32 and 34 are closed at the solution of the problem. For the pipeline network with 5 valves, the smoothing method solved the problem in 16 iterations using algorithm 6.4. This is comparable to 14 and 19 iterations required for solution using mixed integer and continuous formulations respectively in [25]. MINOS took 82 iterations to solve the original nonsmooth problem.

6.6 Conclusions

In this chapter, we considered some recent developments in the applicability of smoothing methods to solve the complementarity problem. An equation based formulation was derived for the phase equilibrium problem, with complementarity conditions modeling appearance and disappearance of phases. The equation based formulation was used to model limiting distillation cases and solved using smoothing methods. The results are encouraging and it is also noted that better computational performance is expected for a more efficient implementation of the ideas described in this chapter. Smoothing methods were also used to derive simple closed form expressions for nested discontinuities, with a single smoothing parameter.

Chapter 7

Conclusions

I do not know what I may appear to the world;
but to myself I seem to have been only like a boy
playing on the seashore and diverting myself
in now and then finding a smoother pebble
or a prettier shell than ordinary,
whilst the great ocean of truth lay all
undiscovered before me.
-Issac Newton [21]

7.1 General Remarks

This thesis addresses issues involved in the treatment of nonsmooth dynamic simulation and large inequality constrained optimization problems encountered in chemical process engineering. The goal has been to develop novel mathematical models and efficient solution techniques. In particular, the following major problems have been analyzed.

1. Development of an efficient framework for the treatment of bounds, conditionals and ill-conditioning, amongst others, occurring in process simulation.

2. Treatment of discontinuities and model changes occurring in dynamic process simulation.

3. Development of a modeling technique for phase equilibrium calculations where the relevant set of equations will automatically fall into place, regardless of the number and nature of the phases present at equilibrium.

4. Efficient and practical ways of initialization and reinitialization of process models described by Differential Algebraic Equations.

5. Development of a methodology which will overcome the combinatorial complexity associated with active set methods in nonlinear programming problems.

6. Strategies for solving complementarity problems arising in process engineering.

In the remainder of the chapter, the major results of the work are outlined. Also, a number of issues, of both theoretical and practical interest, identified as potential areas for longer term research are briefly discussed.

7.2 Summary of Thesis

7.2.1 Successive Linear Programming (SLP) Methods for Process Simulation

In chapter 2, we outlined the advantages of solving constrained simulation problems using a method based on Successive Linear Programming (SLP). The LP based framework provides a natural way for the treatment of bounds, conditionals, singularities and ill-conditioning occurring in process simulation problems. In chapter 2, we improved a previously developed SLP method [24] by incorporating a new descent strategy. Traditionally, numerical methods have used one of the two major classes of methods to ensure descent: line search or trust region. It has been widely understood that trust region methods perform better on ill-conditioned problems whereas line search methods perform well on well-conditioned ones. In chapter 2, we described a framework combining these two methods in the context of SLP. The new descent criterion could be considered to be a *hybrid* of line search and trust region methods. Theoretical and global convergence analysis has also been carried out for the combined method. It has been shown that an accumulation point generated by the algorithm is a stationary point of the merit function, which corresponds to a solution of the original equation solving problem.

In the example problems considered, pure line search methods performed better on some and pure trust regions methods performed better on some others. In a majority of problems, there was no significant difference between a line search and a trust region approach. However, the combined method shows a performance as good as the better of the individual methods in almost all problems. *ie.* the combined method performs as well as trust region methods for ill-conditioned problems and is comparable to line search methods for well-conditioned problems. The results are encouraging and our initial results show that it has vast potential. Though it would be premature to claim that the combined method is an answer to the question of which of line search or trust region methods is better in a general context, our analysis shows that it is definitely a direction to be researched in detail in the future. In chapters 3 and 4, we looked at various aspects of discontinuities occurring in dynamic simulation problems and their solution techniques based on SLP methods developed in chapter 2.

7.2.2 Initialization and Reinitialization of DAEs

The first step in performing dynamic simulation of chemical processes described by DAEs is to provide consistent initial conditions to the model. This task is far from trivial as the number of initial conditions one could independently specify is not known *a priori.*

It could be any number less than or equal to the number of differential variables in the DAE and is mainly dictated by the properties and characteristics of the DAE system. Successive differentiation of the DAE could generate additional algebraic equations which limits the initial conditions which could be specified independently. Overspecification of initial conditions, which occurs mainly due to the lack of knowledge of the degrees of freedom for the specification of initial conditions, could lead to convergence failures in the initialization problem. Hence, it is of vital importance that one specify consistent initial conditions to the solver to generate numerically accurate solutions. Most existing methods expect the user to supply an exact number of initial conditions. Instead, in chapter 3, an optimization formulation is presented for DAE initialization with the derivative array equations as constraints. The formulation involves minimization of the 1-norm of the violation of the initial condition specifications. This is solved using SLP and can readily be extended to large, sparse systems of equations. Solving this minimization problem ensures that the initial conditions are determined within a required accuracy; failure to do so may result in convergence to poor or inaccurate solutions on further integration. A second formulation where one could fix r, the degrees of freedom for the specification of initial conditions, has also been presented.

An equally relevant problem is the one concerning DAE reinitialization. Reinitialization after a discontinuity is an even tougher problem as it involves the additional exercise of determining the subset of variables which are continuous across a discontinuity. The popular assumption used in practice today is that of continuity of all differential variables across a discontinuous input (Pantelides and Barton [81]). In chapter 3 we presented a criterion for checking the validity of the assumption of continuity of differential variables across a discontinuity caused by a step change in the input variables. The validity of continuity is based on the analysis of the underlying ODE of the DAE and its dependence on the derivatives of the input function causing the discontinuity. This analysis is valid for problems of arbitrary index. Such criteria have not been reported in the literature for problems of general index. Here, we also show that the assumption of continuity of differential variables is not necessarily valid. The continuous set of variables can be determined by performing an LP sensitivity analysis on the derivative array equations.

The applications of the ideas outlined in chapter 3 have been demonstrated on small problems, of low and high index. The concepts apply irrespective of the size and index of the problem and application to larger problems is expected to be straightforward.

7.2.3 Nonsmoothness in Dynamic Simulation

The SLP method in chapter 2 was used to develop an improved method for addressing a variety of issues in dynamic process simulation, including treatment of discontinuities. The LP based method has been demonstrated to be an effective way for the imposition of bounds on variables in dynamic simulation problems. A literature review of the methods used for handling discontinuities is presented. In chapter 4, we describe a new algorithm based on SLP for the treatment of discontinuities occurring in dynamic simulation problems. The LP based method helps in event detection and resolution. The method ensures that any event which has occurred within the time interval in consideration is detected, and if more than one event occurs, the detected one is indeed the earliest one. In the case of nonexistence of solution across a discontinuity, a penalty term introduced in our approach makes it less likely to fail when compared to the presently used discontinuity locking methods. Feasibility of the approach has been demonstrated through examples.

7.2.4 Calculation of Phase Equilibrium

We consider an important class of problems occurring in chemical process simulation in chapters 4 and 6: calculation of phase equilibrium. Phase equilibrium problems are nonsmooth in the sense that different sets of equations are valid depending on the number of phases present. An efficient method for dealing with this 'nonsmoothness' could lead to robustness and considerable savings in computational time. In chapter 4, we present a single consistent optimization based formulation for multiphase equilibrium problems which will automatically solve the relevant set of equations without having to determine the number of phases present at equilibrium in advance. A penalty term introduced in the objective takes care of the appearance and disappearance of phases. Example problems solved include calculation of phase equilibrium in the critical region. Dynamic simulation problems from liquid-vapor and liquid-liquid-vapor systems covering three regimes of phase combinations, are also solved using this approach.

In chapter 6, we presented an equation based formulation for phase equilibrium calculations. Smoothing methods were used to solve the complementarity conditions in the equation based formulation. Equation based models in most of the process simulators are formulated for a certain prespecified number and nature of phases. They tend to fail if there is a change in the phase combinations present. The big advantage of the equation based formulation presented in chapter 6 is that it could be used to enhance the capability of currently used process simulators to treat the issue of phase changes, without having to change the process models drastically. Example problems solved using the equation based formulation and smoothing methods include nonideal distillation columns.

7.2.5 Interior Point Methods for Large Scale Optimization

Chapters 3 and 4 describe solution of the problems of initialization and reinitialization, treatment of bounds, conditionals and discontinuities occurring in process simulation with SLP based methods. The next problem we addressed was how to improve the efficiency of solving these SLP subproblems. The general nature of these problems is that they are highly constrained, with the constraints arising from physically realistic conditionals and variable bounds. We also realized that this is exactly the nature of a large fraction of continuous optimization problems arising in chemical process engineering. Hence, we looked at this problem from a larger perspective and addressed the bottleneck issue of the complexity associated with active set methods in nonlinear optimization.

Chapter 5 describes incorporation of the concepts of interior point methods within a nonlinear programming framework, particularly SQP. We considered a class of primal-dual interior point methods, a Mehrotra predictor-corrector method [72], for our study. An intuitive interpretation of the predictor corrector interior point methods using reduced Hessian SQP has been derived. A framework for the adaptation of a path following primal dual interior point method with an SQP approach, ISQP, has been also been developed. This framework has been demonstrated to overcome the combinatorial complexity associated with active set methods on example problems drawn from optimal control. The analysis in this chapter could be considered to be a step in the direction of formulating an efficient methodology for the treatment of inequalities while retaining the ability of SQP methods to take advantage of the problem structure at the same time. Preliminary theoretical analysis of this framework has been outlined using the properties developed in [14]. A large class of problems, both in theory and in applications, can be addressed in the future as an outcome of the analysis in this chapter. Direction for future work is described in detail in section 7.4.

7.2.6 Smoothing Methods

An efficient method for the solution of the complementarity problem has wide implications in process engineering as complementarity is central to all constrained optimization problems. Direct enforcement of the complementarity conditions could be problematic because of the inherent nondifferentiability and singularity associated with them. In chapter 6, we investigated the applicability of smoothing methods to treat complementarity conditions modeling implicit discontinuities in process simulation. Theoretical properties of a class of smoothing methods recently reported in the literature have been reviewed. Example problems for the solution of complementarity conditions have been drawn from distillation models. The distillation columns were modeled using an equation based formulation for phase equilibrium calculations with complementarity conditions describing existence of phases. An algorithm incorporating smoothing methods was able to locate dry and vaporless trays in the examples considered.

Applicability of smoothing methods for modeling other nondifferentiable functions such as the absolute value and min-max operators is also studied. Simple closed form approximations for nested discontinuities have been derived using a smoothing technique proposed by [33]. As part of practical applications, a pipeline network problem with checkvalves was solved using smoothing methods.

7.3 Contributions of Research Work

In summary, this research thesis has made the following contributions to the areas of nonsmooth and inequality constrained simulation and optimization.

1. Consolidation of the advantages of using SLP methods for constrained simulation problems. A previously developed SLP method was made more robust and efficient through the incorporation of a new descent strategy combining line search and trust region methods. The hybrid descent strategy is demonstrated to perform as well as the better of line search or trust region methods, regardless of whether there is a difference in the individual performance of these methods on a particular problem.

2. Global convergence properties of the descent strategy combining line search and trust regions methods in SLP. In the literature, theoretical development in this field is still in a nascent stage.

3. An SLP formulation for consistent initialization of DAE systems. The SLP formulation detects any inconsistent specification of initial conditions, handles the under-determined nature of the derivative array equations efficiently and allows relaible convergence strategies.

4. Derivation of a criterion for determining the set of variables continuous across a discontinuity caused by a step change in the input variable in a DAE system. This analysis is valid for DAEs of arbitrary index, the first result of its kind in the literature. Current literature reports continuity analysis only for index 1 and index 2 problems of special structure.

5. Mathematically showed that the assumption of continuity of differential variables across a discontinuity is not necessarily true.

6. Developed an SLP framework for the treatment of discontinuities occurring in dynamic process simulation. The algorithm based on SLP aids in correct event detection and location and helps in overcoming problems associated with nonexistence of solution across a discontinuity.

7. An optimization formulation is presented for handling appearance and disappearance of phases, for multiphase equilibrium problems. Demonstrated on LV and LLV systems.

8. Extended the optimization formulation to derive an equation based formulation for two and three phase systems. Complementarity conditions in the equation based

formulation handles phase transitions.

9. Investigated incorporation of interior point methods within a nonlinear programming framework to overcome the combinatorial complexity associated with active set methods. A path following Mehrotra predictor-corrector method was incorporated within LP and QP subproblems and benefits demonstrated on problems drawn from optimal control.

10. A novel interpertation of predictor-corrector methods using reduced Hessian SQP (rSQP) is derived. Allows us to take a fresh look at the theoretical properties of predictor-corrector methods from a different perspective, that of rSQP.

11. Investigated the applicability of smoothing methods for the treatment of complementarity conditions arising in process engineering, by utilizing some recent developments in this area. Applied to complementarity problems arising from phase equilibrium to simulate operation of distillation columns leading to dry or vaporless trays.

12. Smoothing methods were used to approximate complicated nested nondifferentiabilites to derive closed form expressions with a single smoothing parameter.

7.4 Recommendations for Future Research

While effective techniques for the solution of nonsmooth and inequality constrained problems in process simulation and optimization have been developed in this thesis, there are several issues to be addressed in the future. Several possible aspects worthy of further investigation are as follows:

Theory

- Analysis of the continuity conditions carried out in chapter 3 is for the case where there is a step change in the input function, which is encountered often in process simulation. However, there could be other kinds of discontinuities as well. What would happen if the model itself changes completely across a discontinuity? Determination of the set of continuous variables in this case is an open question. An even tougher problem is when the sets of variables in the models before and after the discontinuity are different. These cases pose interesting and challenging problems for future research.

- In section 2.4, we outlined global convergence properties for the combined line search-trust region strategy for SLP. Ternet and Biegler [96] have proposed a similar descent strategy for rSQP based methods. Derivation of global convergence properties for SQP based methods is more involved and is of great theoretical interest in the future.

Implementation

- Numerical examples considered in this thesis for verifying the performance of the combined line search-trust region methods for SLP are relatively small. Further testing could be carried out for larger problems and finer details of the algorithm tuned for large scale implementation.

- SLP method for consistent initialization and reinitialization of DAEs described in chapter 3 could be easily interfacable to popularly used DAE solvers like DASSL. One set of problems faced by users of DASSL is the lack of a mechanism to know whether a given specification of initial conditions is consistent or not. Incorporation of SLP methods will make the treatment of the initialization and reinitialization problems easier and more user-friendly within these solvers. Such an interface could be implemented in the future.

Interior Point Methods

- As Wright [100] points out, a successful piece of optimization software requires more than just a good optimization algorithm. Efficiency and stability of the linear algebra are important ingredients. This is most relevant for implementations based on primal-dual interior point methods where solution of linear system of equations takes up bulk of the computational time. Sparse linear algebra codes provide efficient solution of systems like (5.30) and (5.31). Also, optimization problems arising in chemical process engineering are often of special structure and a methodolgy to exploit them could lead to huge computational savings. Devising ways of efficiently solving the linear systems arising at each interior point iteration could be considered in the future. Some preliminary results obtained using Harwell subroutines MA27 and MA28 are described in [3].

- Replacement of active set methods with an interior point strategy allows us to revisit a variety of currently used algorithms in process engineering and identify the problems where such a strategy could make a huge difference. One such problem class is multiperiod design optimization. [98] described a decomposition strategy for multiperiod problems where the computational effort is linear to the number of periods. However, the QP subproblem in the subspace of the design variables at each iteration is solved using an active set strategy. An interior point approach for solving the QP subproblem is expected to make the solution more efficient and is currently under investigation by Bhatia *et al.* [12].

- Efficient implementations of interior point methods for large scale nonlinear programming problems could be considered in the future. Chapter 5 outlines the theory and a premise for implementing these ideas. Implementation for the example problems in that chapter is for solving quadratic programming problems and could be extended to nonlinear programming problems by utilizing subroutines already in place in existing codes.

- A fresh look at the theoretical properties of primal-dual interior point methods in the light on their interpertation using rSQP, could be done in the future as described in section 5.5.

Global Minima of Gibbs Free Energy

A particular phase composition is stable if it satisfies the tangent plane criteria (Baker et al. [6]). The tangent plane criteria ensures that the Gibbs free energy is at its global minimum *ie.* it is a necessary and sufficient condition for phase stability. A qualitative interpretation of the tangent plane criteria is that the tangent drawn to the Gibbs energy surface at the equilibrium compositions should not intersect the Gibbs energy surface itself anywhere. This concept is illustrated in figure 7.1 where the Gibbs energy surface for a two-phase system is plotted. For the given feed there is more than one candidate solution for the equilibrium phase compositions, but only one of them corresponds to the global minimum of Gibbs free energy. As shown in the figure, a tangent to the G surface at xL and xV does not intersect the surface anywhere. Hence this corresponds to the true solution according to the tangent plane criteria. On the other hand a tangent to the surface at xL' and xV' intersects the surface and locates a false solution. It should be noted that the two phase Gibbs energy in the former case (g) is less than that of the latter (g'). An additional observation could be made about the location of the equilibrium phase compositions from the tangent plane criteria. The solution must lie only on the convex parts of the nonconvex Gibbs energy surface.

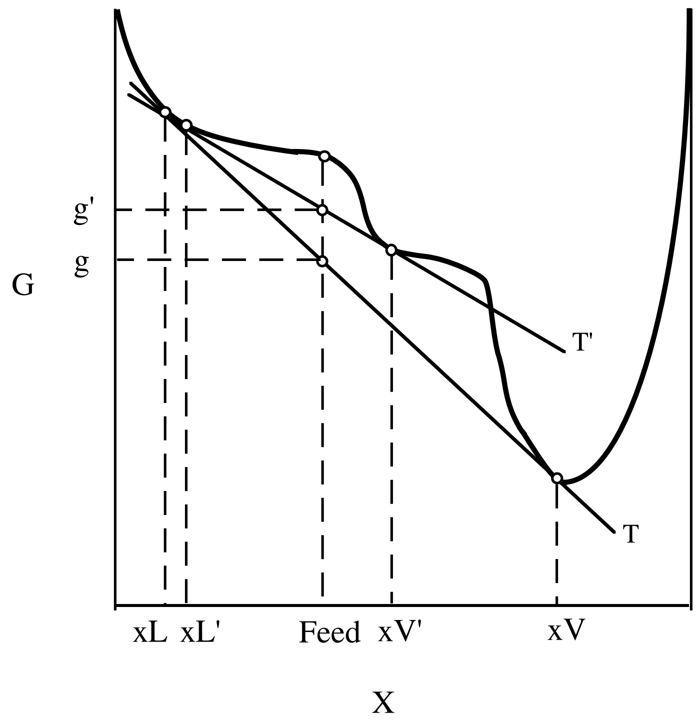

Figure 7.1: Tangent plane criteria

Based on this Swaney and Kendlbacher [93] suggested preprocessing of the surface by partitioning the domain into subregions given by convex parts. In other words the nonconvex potential surface will be carved up to form convex components and the search is limited to these convex surfaces. Each convex part is represented as a separate convex phase extended to cover the entire domain. This approach has the advantage of finding the global solution over conventional methods which may converge to a local minimum depending on the initial guess. The parameters for the 'convexified' representation of the surface are generated only once and the subsequent calculations could be very rapid if implemented with an LP based solver. If implemented, this approach may not be as expensive as global optimization algorithms. The convex phases could be represented using a simple phase equilibrium model. The Wilson equation for excess Gibbs energy, which is widely used for representing the behavior of polar and nonpolar multicomponent mixtures with only binary parameters, could be used to represent the convex surfaces.

The fact that none of the points on the Gibbs energy surface should lie below the tangent plane at the solution, in principle, points to infinite number of constraints. Thus, an alternate way to incorporate the tangent plane criteria could be through semi-infinite programming [58]. Semi-infinite programming considers optimization problems with finite number of variables and infinite number of constraints. However, this problem is challenging, because of the nonlinear nature of the constraints encountered in phase equilibrium calculations. Development of methods based on semi-infinite programming could be explored in the future.

Bibliography

[1] J. Albuquerque. *Parameter Estimation and Data Reconciliation for Dynamic Systems.* PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 1996.

[2] J. Albuquerque and L. T. Biegler. Decomposition algorithms for on-line estimation with nonlinear models. *Computers chem. Engng.*, 19(10):1031–1040, 1995.

[3] J. Albuquerque, V. Gopal, G. Staus, L. T. Biegler, and B. E. Ydstie. Interior point SQP strategies for structured process optimization problems. *Computers chem. Engng.*, 1997, To appear.

[4] L. Armijo. Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific J. Math.*, 16:1–3, 1966.

[5] R. Bachmann, L. Brull, T. Mrziglod, and U. Pallaske. On methods for reducing the index of differential - algebraic equations. *Computers chem. Engng.*, 14(11):1271–1273, 1990.

[6] L. E. Baker, A. C. Pierce, and K. D. Luks. Gibbs energy analysis of phase equilibria. *Soc. Pet. Eng. J.*, 22(5):731–742, 1982.

[7] S. Balakrishna and L. T. Biegler. Targeting strategies for the synthesis and energy integration of nonisothermal reactor networks. *Ind. Eng. Chem. Res.*, 31(9):2152–2163, 1992.

[8] I. Barrodale and F.D.K. Roberts. An efficient algorithm for discrete l_1 linear approximations with linear constraints. *SIAM J. Numer. Anal.*, 15(3):603–611, 1978.

[9] J. Barros-Neto. *An Introduction to the Theory of Distributions.* Marcel Dekker, 1973.

[10] P.I. Barton. *The Modeling and Simulation of Combined Discrete/Continuous Processes.* PhD thesis, Imperial College of Science, Technology and Medicine, London, UK, 1992.

[11] D. P. Bertsekas. Projected Newton methods for optimization problems with simple constraints. *SIAM J. on Control and Optimization*, 20(2):221–246, 1982.

[12] T. Bhatia, L. T. Biegler, and V. Gopal. Multiperiod design and planning with interior point rSQP methods. In *AIChE Annual Meeting*, Los Angeles, CA, 1997.

[13] L. T. Biegler, J. J. Damiano, and G. E. Blau. Nonlinear parameter estimation: A case study comparison. *AIChE J.*, 32(1):29–45, 1986.

[14] L. T. Biegler, J. Nocedal, and C. Schmid. A reduced hessian method for large-scale constrained optimization. *SIAM J. Optimization*, 5(2):314–347, 1995.

[15] L.T. Biegler. Chemical process simulation. *Chem. Eng. Prog.*, 85(10):50–61, 1989.

[16] L.T. Biegler. From nonlinear programming theory to practical optimization algorithms: A process engineering viewpoint. *Computers chem. Engng.*, 16(Suppl.):S81–S96, 1992.

[17] S. C. Billups, S. P. Dirske, and M.C. Ferris. A comparison of algorithms for large scale mixed complementarity problems. Technical report, Computer Sciences Department, University of Wisconsin, 1995.

[18] G. E. Blau, L. Kirby, and M. Marks. An industrial kinteics problem for testing nonlinear parameter estimation algorithms. *Process Math Modeling Department, The Dow Chemical Company*, 1981.

[19] J. F. Boston and H. L. Britt. A radically different formulation and solution of the single-stage flash problem. *Computers chem. Engng.*, 2(3):109–122, 1978.

[20] K. E. Brenan, S. L. Campbell, and L. R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations.* SIAM, Philadelphia, 1996.

[21] D. Brewster. *Memoirs of the life, writings, and discoveries of Sir Issac Newton.* T. Constable and Co., Edinburgh, UK, 1855.

[22] A. Brooke, D. Kendrick, and A. Meeraus. *GAMS - A user's guide (Release 2.25).* The Scientific Press, San Francisco, CA, 1992.

[23] L. Brüll and U. Pallaske. On consistent initialization of differential-algebraic equations with discontinuities. In *Proceedings of the Fourth European Conference on Mathematics in Industry*, pages 213–217, Teubner, Stuttgart, 1992.

[24] L. G. Bullard and L. T. Biegler. Iterative linear programming strategies for constrained simulation. *Computers chem. Engng.*, 15(4):239–254, 1991.

[25] L. G. Bullard and L. T. Biegler. Iterative linear programming strategies for nonsmooth simulation: Continuous and mixed integer approaches. *Computers chem. Engng.*, 16(10):949–962, 1992.

[26] L. G. Bullard and L. T. Biegler. Iterated linear programming strategies for nonsmooth simulation: a penalty based method for vapor-liquid equilibrium applications. *Computers chem. Engng.*, 17(1):95–110, 1993.

[27] L.G. Bullard. *Iterated Linear Programming Strategies for Constrined and Nonsmooth Simulation.* PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 1991.

[28] S. L. Campbell and E. Moore. Progress on a general numerical method for nonlinear higher index DAEs II. *Circuits Systems Signal Process*, 13:123–138, 1994.

[29] M. Caracotsios and W. E. Stewart. Sensitivity analysis of initial value problems with mixed odes and algebraic equations. *Computers chem. Engng.*, 9(4):359–365, 1985.

[30] M. B. Carver. Efficient integration over discontinuities in ordinary differential equations. In A.W. Bennett and R. Vichnevetsky, editors, *Numerical Methods for Differential Equations and Simulation.* North-Holland Publishing Company, 1978.

[31] R. M. Chamberlain, M. J. D. Powell, Lemerchal C., and H. C. Pedersen. The watchdog method for forcing convergence in algorithms for constrained optimization. *Math. Prog. Studies*, 16:1–17, 1982.

[32] B. Chen and P. T. Harker. A non-interior-point continuation method for linear complementary problems. *SIAM J. on Matrix Analysis and Applications*, 14(4):1168–1190, 1993.

[33] C. Chen and O. L. Mangasarian. A class of smoothing functions for nonlinear and mixed complementarity problems. *Comput. Optim. Appl.*, 5(2):97–138, 1996.

[34] Y. Chung. *Solving Index and Near Index Problems in Dynamic Simulation.* PhD thesis, Carnegie Mellon University, Pittsburgh, PA 15213, 1991.

[35] Y. Chung and A.W. Westerberg. A proposed numerical method for solving nonlinear index problems. *Ind. Eng. Chem. Res.*, 29(7):1234–1239, 1990.

[36] F. H. Clarke. *Optimization and Nonsmooth Analysis.* SIAM, Philadelphia, 1990.

[37] G.B. Dantzig. *Linear Programming and Extensions.* Princeton University Press, Princeton, N.J., 1963.

[38] J.E. Dennis, S-B. B Li, and R. A. Tapia. A unified approach to global convergence of trust region methods for nonsmooth optimization. *Mathematical Programming*, 68(3):319–346, 1995.

[39] J.E. Dennis and R.B. Schnabel. A view of unconstrained optimization. In G. L. Nemhauser et al., editor, *Handbook in OR and MS, Vol.1*, Elsevier Publishers B.V., North-Holland, 1989.

[40] J.E. Dennis and R.B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* SIAM, Philadelphia, PA, 1996.

[41] J.J. Downs and E.F. Vogel. A plant-wide industrial process control problem. *Computers Chem. Eng.*, 17(3):245–255, 1993.

[42] A. Drud. CONOPT - a GRG code for large sparse dynamic nonlinear optimization problems. *Mathematical Programming*, 31(2):153–191, 1985.

[43] I.S. Duff, J. Nocedal, and J.K. Reid. The use of linear programming for the solution of sparse sets of nonlinear equations. *SIAM J. Sci. Statist. Comput.*, 8(2):99–108, 1987.

[44] M. Duran. *A mixed integer nonlinear programming approach for the systematic synthesis of engineering systems.* PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 1984.

[45] L. B. Evans. Computer-aided design: Advances in process flowsheeting systems. In Y.A. Liu, H.A. McGee, and W.R. Epperly, editors, *Recent Developments in Chemical Process and Plant Design*, pages 261–287. Wiley, 1987.

[46] L. B. Evans. Steady-state simulation: A state-of-the-art review. In *5th International Symposium on Process Systems Engineering*, pages 953–967, Kyongju, Korea, 1994.

[47] M. C. Ferris and J. S. Pang. Engineering and economic applications of complementarity problems. Technical Report 95-7, Department of Economics, University of Colorado, 1995.

[48] R. Fletcher. *Practical Methods of Optimization.* Wiley, New York, 1987.

[49] C.W. Gear. The automatic integration of ordinary differential equations. *Comm. ACM*, 14(3):176, 1970.

[50] C.W. Gear. Simultaneous numerical solution of differential-algebraic equations. *IEEE Trans. Circuit Theory*, 18:89–95, 1971.

[51] C.W. Gear. Differential-algebraic equation index transformation. *SIAM J. Sci. Stat. Comput.*, 9(1):39–47, 1988.

[52] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. User's guide for SOL/QPSOL: A Fortran package for quadratic programming. Technical Report SOL 83-7, Department of Operations Research, Stanford University, Stanford, CA, 1983.

[53] V. Gopal and L. T. Biegler. Nonsmooth dynamic simulation with linear programming based methods. *Computers chem. Engng.*, 21(7):675–689, 1997.

[54] V. Gopal and L. T. Biegler. A successive linear programming approach to consistent initialization and reinitialization after discontinuities of differential algebraic equations. *SIAM J. of Scientific Computing*, 1997, To appear.

[55] I.E. Grossmann. MINLP optimization strategies and algorithms for process synthesis. In J.J. Siirola, I.E. Grossmann, and G. Stephanopoulos, editors, *Third International Conference on Foundations of Computer Aided Process Design*, pages 105–132, 1990.

[56] I.E. Grossmann, I. Quesada, R. Raman, and V.T. Voudouris. Mixed integer optimization techniques for the design and scheduling of batch processes. In *NATO Advanced Study Institute - Batch Process Systems Engineering*, Antalya, Turkey, 1992.

[57] S-P. Han. A globally convergent method for nonlinear programming. *JOTA*, 22(3):297–310, 1977.

[58] R. Hettich and K. O. Kortanek. Semi-infinite programming. *SIAM Review*, 35(3):380–429, 1993.

[59] D.M. Himmelblau and K.B. Bischoff. *Process Analysis and Simulation.* Wiley, New York, 1968.

[60] G.S. Joglekar and G.V. Reklaitis. A simulator for batch and semi-continuous processes. *Computers Chem. Engng.*, 8(6):315–328, 1984.

[61] C. Kanzow. Some tools allowing interior-point methods to become noninterior. Technical report, Institute of Applied Mathematics, University of Hamburg, Germany, 1994.

[62] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[63] Omar Khayyám. *Rubáiát.* B. Quaritch, London, 1859.

[64] D. Kyriakopoulou and B. Kalitventzeff. Validation of measurement data using and interior point SQP. *Computers chem. Engng.*, 20(Suppl.):S563–S568, 1996.

[65] B. Leimkuhler, L. R. Petzold, and C. W. Gear. Approximation methods for the consistent initialization of differential - algebraic equations. *SIAM J. of Numer. Anal.*, 28(1):205–226, 1991.

[66] I. J. Lustig, R. E. Marsten, and D. F. Shanno. On implementing Mehrotra's predictor-corrector interior-point method for linear programming. *SIAM J. Optimization*, 2(3):435–449, 1992.

[67] C. Majer, W. Marquardt, and E.D. Gilles. Reinitialization of daes after discontinuities. *Computers chem. Engng.*, 19:S507–S512, 1995.

[68] T. Maly and L. R. Petzold. Numerical methods and software for sensitivity analysis of differential-algebraic systems. *Appl. Numer. Math.*, To appear.

[69] W. Marquardt. Dynamic process simulation - recent progress and future challenges. In Y. Arkun and W.H. Ray, editors, *Chemical Process Control*, pages 131–179. CACHE-AIChE Publications, 1991.

[70] P. M. Mathias, J. F. Boston, and S. Watanasiri. Effective utilization of equations of state for thermodynamic properties in process simulation. *AIChE J.*, 30(2):182–186, 1984.

[71] C.M. McDonald and C. A. Floudas. Decomposition based and branch and bound global optimization approaches for the phase equilibrium problem. *Journal of Global Optimization*, 5:205–251, 1994.

[72] S. Mehrotra. On the implementation of a primal-dual interior point method. *SIAM J. Optimization*, 2:575–601, 1992.

[73] R. D. C. Monteiro and S. J. Wright. Superlinear primal-dual affine scaling algorithms for lcp. *Mathematical Programming*, 69:311–333, 1995.

[74] R. Morales and R. Sargent. Numerical experiments with an interior point method for large scale convex quadratic programming. In *4th SIAM Conf. on Optimization*, Chicago, 1992.

[75] B.A. Murtagh and M.A. Saunders. A projected Lagrangian algorithm and its implementation for sparse nonlinear constraints. *Math Prog. Study*, 16:84–117, 1982.

[76] L. Naess, A. Mjaavatten, and J. Li. Using dynamic process simulation from conception to normal operation of process plants. *Computers Chem. Engng.*, 16(Suppl.):S119–S130, 1992.

[77] P. A. Nelson. Rapid phase determination in multiple-phase flash calculations. *Computers Chem. Engng.*, 11(6):581–591, 1987.

[78] A. Nordsieck. On the numerical integration of ordinary differential equations. *Math. Comp.*, 16:22–49, 1962.

[79] C. C. Pantelides. The consistent initialization of differential - algebraic systems. *SIAM J. Sci. Stat. Comput.*, 9(2):213–231, 1988.

[80] C. C. Pantelides. Speedup- recent advances in process simulation. *Computers Chem. Engng.*, 12(7):745–756, 1988.

[81] C. C. Pantelides and P. I. Barton. Equation-oriented dynamic simulation: Current status and future perspectives. *Computers chem. Engng.*, 17(Suppl.):S263–S285, 1993.

[82] T. Park and P. I. Barton. A new algorithm for the accurate and efficient location of state events. In *Special Topical Conference on Industrial Chemical Technology, AIChE Annual Meeting*, St. Louis, MO, 1993.

[83] D. Y. Peng and D. B. Robinson. A new two-constant equation of state. *Ind. Eng. Chem. Fund.*, 15(1):59–63, 1976.

[84] J.D. Perkins. Survey of existing systems for the dynamic simulation of industrial processes. *Model. Identif. Control*, 7(2):71–81, 1986.

[85] H. H. Pham and M. F. Doherty. Design and synthesis of heterogeneous azeotropic distillations - 1. heterogeneous phase diagrams. *Chem. Engng Sci.*, 45(7), 1990.

[86] J. M. Pinto. *Mixed Integer and Logic Based Optimization Techniques for Scheduling of Multistage Chemical Processes.* PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 1995.

[87] Plato. *The laws.* Penguin classics, Penguin, 1970.

[88] M.J.D. Powell. A fast algorithm for nonlinear constrained optimization calculations. In *Dundee Conference on Numerical Analysis*, 1977.

[89] S. Smale. Algorithms for solving equations. In *Proceedings of the International Congress of Mathematicians*, pages 172–195, Ameri. Math. Soc., Providence, 1987.

[90] G. J. Smith and W. Morton. Dynamic simulation using an equation-oriented flowsheeting package. *Computers chem. Engng.*, 12(5):469–474, 1988.

[91] M. A. Stadtherr, C. A. Schnepper, and J. F. Brennecke. Robust phase stability analysis using interval methods. In L. T. Biegler and M. F. Doherty, editors, *Fourth International Conference on Foundations of Computer-Aided Process Design*, pages 356–359. CACHE-AIChE Publications, 1995.

[92] A. C. Sun and W. D. Seider. Homotopy-continuation method for stability analysis in the global minimization of the Gibbs free energy. *Fluid Phase Equilibria*, 103(2):213–249, 1995.

[93] R. E. Swaney and T. Kendlbacher. Robust solution of phase equilibrium calculations. In *AIChE Annual Meeting*, San Francisco, CA, 1994.

[94] P. Tanartkit. *Stable Computational Procedures for Dynamic Optimization in Process Engineering*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 1996.

[95] R. Tapia, Y. Zhang, M. Saltzman, and A. Weiser. The Mehrotra predictor-corrector interior point method as a perturbed composite Newton method. *SIAM J. Optimization*, 6(1):47–56, 1996.

[96] D. Ternet and L.T. Biegler. Reduced Hessian SQP methods for process optimization: Some recent advances. In *5th SIAM Conference on Optimization*, Victoria, Canada, 1996.

[97] J. Unger, A. Kroner, and W. Marquardt. Structural analysis of differential - algebraic equation systems - theory and applications. *Computers chem. Engng.*, 19(8):867–882, 1995.

[98] D.K. Varvarezos, L.T. Biegler, and I.E. Grossmann. Multiperiod design optimization with SQP decomposition. *Computers chem. Engng.*, 18(7):579–595, 1994.

[99] A. Whitehead. *Science and the modern world*. Macmillan, New York, 1925.

[100] S. Wright. *Primal-Dual Interior Point Methods.* SIAM, Philadelphia, 1996.

[101] J.-S. Wu and P. R. Bishnoi. An algorithm for three-phase equilibrium calculations. *Computers chem. Engng.*, 10(3):269–276, 1986.

[102] D. Zhang and Y. Zhang. A Mehrotra-type predictor-corrector algorithm with polynomiality and Q-subquadratic convergence. *Annals of Operations Research*, To appear.

[103] J. Zhang, N. Kim, and L. Lasdon. An improved successive linear programming algorithm. *Mgmt Sci.*, 31(10):1312–1331, 1985.

[104] Y. Zhang and D. Zhang. Superlinear convergence of infeasible-interior-point methods for linear programming. *Mathematical Programming*, 66:361–377, 1994.

[105] Y. Zhang and D. Zhang. On polynomiality of the Mehrotra-type predictor-corrector interior-point algorithms. *Mathematical Programming*, 68:303–318, 1995.

Appendix A

Properties of Smooth Formulation

Conditions for which $\delta_i = 0 \Rightarrow \lambda_i = 1$

The smooth formulation presented in chapter 3 is:

$$
\begin{aligned}
& Min \sum_{i=1}^{n_{ic}} \omega_i \delta_i \\
s.t.\quad & F^{\nu}(x, z, z', \ldots, z^{(\nu)}, t) = 0 && (3.13a) \\
& -\delta_i \leq X_i - X_i^{sp} \leq \delta_i \qquad i = 1, \ldots, n_{ic} && (3.13b) \\
& \delta_i - \beta \ \log(1 + e^{-(\lambda_i - \delta_i)/\beta}) = 0 \quad i = 1, \ldots, n_{ic} && (3.13c) \\
& \sum_{i=1}^{n_{ic}} \lambda_i - r = 0 && (3.13d) \\
& 0 \leq \lambda_i \leq 1 \qquad i = 1, \ldots, n_{ic} && (3.13e) \\
& \delta_i \geq 0 \qquad i = 1, \ldots, n_{ic} && (3.13f) \\
& X_i^L \leq X_i \leq X_i^U && (3.13g)
\end{aligned}
$$

The Karush-Kuhn-Tucker conditions of (3.13) are

$$
\begin{aligned}
\frac{\partial \text{Ł}}{\partial \lambda_i} = \phi_i \frac{e^{-(\lambda_i - \delta_i)/\beta}}{1 + e^{-(\lambda_i - \delta_i)/\beta}} + \gamma - \xi_i^1 + \xi_i^2 &= 0 && (A.1) \\
\xi_i^1 \lambda_i &= 0 && (A.2) \\
\xi_i^2 (\lambda_i - 1) &= 0 && (A.3)
\end{aligned}
$$

where ϕ_i, γ and ξ_i are the multipliers of (3.13c), (3.13d) and (3.13e) respectively. From (3.13c),

$$1 + e^{-(\lambda_i - \delta_i)/\beta} = e^{\delta_i/\beta} \tag{A.4}$$

(A.1) could be simplified as

$$\phi_i e^{-\lambda_i/\beta} + \gamma - \xi_i^1 + \xi_i^2 = 0 \tag{A.5}$$

The smoothing constraint (3.13c) could be written as

$$e^{-\delta_i/\beta} + e^{-\lambda_i/\beta} = 1 \tag{A.6}$$

When $\delta_i = 0$, $e^{-\delta_i/\beta} = 1$. From (A.6),

$$e^{-\lambda_i/\beta} = 0 \tag{A.7}$$

For any $\beta > 0$, $\lambda_i \to \infty$. Choosing β small will allow $\lambda_i < \infty$, in order to satisfy $e^{-\lambda_i/\beta} \leq \epsilon$, a nonzero tolerance. From (A.7) and (A.2),

$$\xi_i^1 = 0 \tag{A.8}$$

From (A.5), (A.7) and (A.8), we have when $\delta_i = 0$,

$$\gamma + \xi_i^2 = 0 \tag{A.9}$$

To show that λ_i goes to its upper bound (ie. $\xi_i^2 \neq 0$) we need to show that γ is nonzero. The following conditions are considered based on the application of (3.13).

- Consider initial condition specifications where the number of specifications is greater than or equal to r and any subset of these specifications with number of elements greater than r is inconsistent.

  (3.13a) has r degrees of freedom. Hence, for the above case, the number of initial condition specifications which could be enforced at the solution of the problem will be at most r. *ie.* the number of δ_is which can be zeros at the solution will be less than or equal to r. Let r_0 be the cardinality of the set of δ_is which are zero at the solution of the problem. Then

$$r_0 \leq r \tag{A.10}$$

If we assume $\gamma = 0$, from (A.9), $\xi_i^2 = 0$. The complementarity condition (A.3) gives $\lambda_i < 1$. Summing up all the λ_is, we have

$$\sum_1^{r_0} \lambda_i < r_0 \leq r \Rightarrow \sum_1^{r_0} \lambda_i < r \tag{A.11}$$

This is clearly a contradiction to (3.13d).

For (3.13d) to be satisfied, the λ_is corresponding to all nonzero δ_is will have to go to their upper bounds and the cardinality of the corresponding set of δ_is, $r_0 = r$. ie. exactly r of the initial condition specifications are set equal to their specified values at the solution of the problem.

- Consider the case when the number of specifications is greater than r, there exists a subset of greater than r elements, or the entire set of specifications, which when specified simultaneously is consistent.

  In these cases, the number of specifications which could be enforced at the solutions could be greater than r because there exists a set of cardinality greater than r which is consistent. λ could take fractional values in this case and the corresponding solution with (3.13) could fix greater than r variables at their specified values. The formulation, nevertheless, serves the purpose of choosing the best set of r or greater than r specifications. The form of the solution will be different from (3.10) in this case.

Appendix B

Multiphase Formulation

We show that the solution of the formulation (16) for multiphase systems would satisfy the Kuhn-Tucker conditions of a Gibbs free energy minimization formulation.

For an n component, p phase mixture, the Gibbs minimization formulation would be

$$\begin{aligned} & Min \quad G = \sum_p \sum_i n_i^p \left(\Delta G_i^f + RT \ln f_i^p \right) \\ s.t. \quad & \sum_p n_i^p = n_i^T \qquad i = 1, n_c \\ & \sum_i n_i^p \geq 0 \qquad p = 1, n_p \end{aligned} \tag{B.1}$$

where n_i^p denotes the moles of component i in phase p. Simplifying (B.1),

$$\begin{aligned} & Min \quad G = \sum_i n_i^T \Delta G_i^f + \sum_p \sum_i n_i^p RT \ln f_i^p \\ s.t. \quad & \sum_p n_i^p = n_i^T \qquad i = 1, n_c \\ & \sum_i n_i^p \geq 0 \qquad p = 1, n_p \end{aligned} \tag{B.2}$$

The Kuhn-Tucker conditions of the problem can be simplified to

$$RT \ln f_i^p + RT \sum_p \sum_i n_i^p \frac{\partial \ln f_i^p}{\partial n_i^p} - \alpha_i - \beta_p = 0 \quad i = 1, n_c \; p = 1, n_p \tag{B.3}$$

$$\beta_p \sum_i n_i^p = 0 \qquad p = 1, n_p \tag{B.4}$$

where α_i and β_p are the multipliers for the equalities and the inequalities respectively.

From the Gibbs-Duhem theorem,

$$\sum_p \sum_i n_i^p \frac{\partial \ln f_i^p}{\partial n_i^p} = 0$$

Define

$$\beta_p = RT \ln \Gamma_p \tag{B.5}$$

(B.3) reduces to $RT \ln \frac{f_i^p}{\Gamma_p} = \alpha_i$, a constant for component i in all phases
or

$$\frac{f_i^{p_m}}{\Gamma_{p_m}} = \frac{f_i^{p_n}}{\Gamma_{p_n}} \qquad \forall p_m, p_n \tag{B.6}$$

If $\sum_i n_i^p = M^p = 0$, where M_p is the molar flowrate of phase p, (B.4) $\Rightarrow \beta_p > 0$ and hence (B.5) $\Rightarrow \Gamma_p > 1$.

On the other hand if $M_p > 0$, (B.4) $\Rightarrow \beta_p = 0$ and hence (B.5) $\Rightarrow \Gamma_p = 1$. ie.

$$\begin{aligned} &\Gamma_p = 1 \Rightarrow \text{phase } p \text{ exists} \\ &\Gamma_p > 1 \Rightarrow \text{phase } p \text{ does not exist} \end{aligned} \tag{B.7}$$

Define a reference phase p_{ref}.
(B.6) can be written as

$$\frac{f_i^{p_{ref}}}{f_i^p} = \frac{\Gamma_{p_{ref}}}{\Gamma_p} \qquad p = 1, n_p,\ p \neq p_{ref} \qquad i = 1, n_c \tag{B.8}$$

But

$$f_i^p = P x_i^p \phi_i^p \tag{B.9}$$

(B.8) simplifies to

$$x_i^{p_{ref}} = \left(\frac{\Gamma_{p_{ref}}}{\Gamma_p}\right)\left(\frac{\phi_i^p}{\phi_i^{p_{ref}}}\right) x_i^p \qquad p = 1, n_p,\ p \neq p_{ref}\ i = 1, n_c \tag{B.10}$$

$$x_i^{p_{ref}} = \gamma_p K_i^p x_i^p \tag{B.11}$$

where

$$\gamma_p = \frac{\Gamma_{p_{ref}}}{\Gamma_p}, \text{a constant} \tag{B.12}$$

and

$$K_i^p = \frac{\phi_i^p}{\phi_i^{p_{ref}}}, \text{the equilibrium constant for phases } p \text{ and } p_{ref} \tag{B.13}$$

(B.11) could be written as

$$x_i^{p_{ref}} = \bar{K}_i^p x_i^p \tag{B.14}$$

where $\bar{K}_i^p$ is a pseudo equilibrium constant. Note that $\bar{K}_i^p = K_i^p$ when both phases p and p_{ref} coexist.

For the flash operation, $n_i^p = M^p x_i^p$, so we can express the flash problem using the penalty formulation as in (4.24), which will reduce to (4.25) for a two phase system (with the vapor phase as the reference phase).

The following shows the correlation of the values of γ_p to the existence of phases:

- $\gamma_p < 1 \Rightarrow \Gamma_p > \Gamma_{p_{ref}} \Rightarrow \Gamma_p > 1 \Rightarrow$ phase p does not exist
- $\gamma_{p_m} > \gamma_{p_n} \Rightarrow \Gamma_{p_n} > \Gamma_{p_m} \Rightarrow \Gamma_{p_n} > 1 \Rightarrow$ phase p_n does not exist
- Any $\gamma_p > 1 \Rightarrow \Gamma_{p_{ref}} > \Gamma_p \Rightarrow \Gamma_{p_{ref}} > 1 \Rightarrow$ phase p_{ref} does not exist

For a two phase system this becomes:

- $\gamma = 1 \Rightarrow$ both liquid and vapor phases exist
- $\gamma > 1 \Rightarrow$ vapor phase does not exist
- $\gamma < 1 \Rightarrow$ liquid phase does not exist

Appendix C

rSQP and Mehrotra Predictor-Corrector

Proof of theorem 5.1

Combining the range and null space steps in (5.31) and (5.30) we get the total step (C.1). For ease of presentation, the iteration index k is dropped for the entire analysis.

$$\begin{bmatrix} Q & A^T & C^T & 0 \\ A & 0 & 0 & 0 \\ C & 0 & 0 & I \\ 0 & 0 & S & V \end{bmatrix} \begin{bmatrix} \Delta x^{zy} \\ \Delta \lambda^{zy} \\ \Delta \nu^{zy} \\ \Delta s^{zy} \end{bmatrix} = - \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ SVe - \mu e + \frac{f'}{f''} e \end{bmatrix} \tag{C.1}$$

The residuals of the nonlinear equations after the range and null space steps are given by (C.2).

$$r(z_k + Yp_y + Zp_z) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \overline{r}_\mu \end{bmatrix} \tag{C.2}$$

where

$$\begin{aligned}
\bar{r}_\mu &= SVe - \mu e \\
&= (S + \Delta S^{zy})(V + \Delta V^{zy})e - (\mu + \Delta\mu)e \\
&= SVe + \Delta S^{zy}\Delta V^{zy}e + (\Delta S^{zy}V + S\Delta V^{zy})e - (\mu + \Delta\mu)e \\
&\overset{(C.1)}{=} SVe + \Delta S^{zy}\Delta V^{zy}e + (-SV + \mu - \frac{f'}{f''})e - \mu e - \frac{f'}{f''}e \\
&= \Delta S^{zy}\Delta V^{zy}e
\end{aligned} \tag{C.3}$$

From (C.1), (C.2) and (C.3), after second order correction, we have

$$\begin{bmatrix} Q & A^T & C^T & 0 \\ A & 0 & 0 & 0 \\ C & 0 & 0 & I \\ 0 & 0 & S & V \end{bmatrix} \begin{bmatrix} \Delta x^{\overline{zy}} \\ \Delta\lambda^{\overline{zy}} \\ \Delta\nu^{\overline{zy}} \\ \Delta s^{\overline{zy}} \end{bmatrix} = - \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ SVe - \mu e + \frac{f'}{f''}e + \Delta S^{zy}\Delta V^{zy}e \end{bmatrix} \tag{C.4}$$

Now, a combination of the predictor and the corrector steps in a Mehrotra approach gives (C.5).

$$\begin{bmatrix} Q & A^T & C^T & 0 \\ A & 0 & 0 & 0 \\ C & 0 & 0 & I \\ 0 & 0 & S & V \end{bmatrix} \begin{bmatrix} \Delta x^{mpc} \\ \Delta\lambda^{mpc} \\ \Delta\nu^{mpc} \\ \Delta s^{mpc} \end{bmatrix} = - \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ SVe - \sigma\mu e + \Delta S^p\Delta V^p e \end{bmatrix} \tag{C.5}$$

where $\Delta S^p\Delta V^p$ follow from the predictor formula in (5.15). We require

$$\begin{bmatrix} \Delta x^{\overline{zy}} \\ \Delta\lambda^{\overline{zy}} \\ \Delta\nu^{\overline{zy}} \\ \Delta s^{\overline{zy}} \end{bmatrix} = \begin{bmatrix} \Delta x^{mpc} \\ \Delta\lambda^{mpc} \\ \Delta\nu^{mpc} \\ \Delta s^{mpc} \end{bmatrix} \tag{C.6}$$

If we compare $\Delta S^{zy}\Delta V^{zy}$ and $\Delta S^p\Delta V^p$, we see that both are determined by a scalar relation between σ, μ and $f(\mu)$. Therefore, for (C.6) to hold, from (C.4) and (C.5) we require that

$$SVe - \mu e + \frac{f'}{f''}e + \Delta S^{zy}\Delta V^{zy}e = SVe - \sigma\mu e + \Delta S^p\Delta V^p e \tag{C.7}$$

Because of the scalar relation we can write

$$\Delta s^{zy^T}\Delta\nu^{zy} - \Delta s^{p^T}\Delta\nu^{p} = n\left[(1-\sigma)\mu_k - \frac{f'}{f''}\right] \tag{C.8}$$

We now explicitly evaluate the left hand side of (C.8). First, we form the inverse of the coefficient matrix in (C.4) and (C.5) written in the symmetric form in (C.9).

$$\begin{bmatrix} Q & A^T & C^T & \mid & 0 \\ A & 0 & 0 & \mid & 0 \\ C & 0 & 0 & \mid & I \\ - & - & - & - & - \\ 0 & 0 & I & \mid & S^{-1}V \end{bmatrix}^{-1} = \begin{bmatrix} A_1 & B_1 \\ B_1^T & D_1 \end{bmatrix}^{-1} = \begin{bmatrix} A_2 & B_2 \\ B_2^T & D_2 \end{bmatrix} \tag{C.9}$$

where A_1, B_1 and D_1 are defined as in (5.37), (5.38) and (5.39) respectively.
We have

$$\begin{aligned} A_1B_2 + B_1D_2 &= 0 && \text{(C.10)}\\ B_1^TB_2 + D_1D_2 &= I && \text{(C.11)}\\ B_1^TA_2 + D_1B_2^T &= 0 && \text{(C.12)}\\ A_2A_1 + B_2B_1^T &= I && \text{(C.13)}\\ A_2B_1 + B_2D_1 &= 0 && \text{(C.14)}\\ B_2^TA_1 + D_2B_1^T &= 0 && \text{(C.15)} \end{aligned}$$

A_2 and D_2 are symmetric. From (C.14)

$$B_2 = -A_2B_1D_1^{-1}$$

From (C.13)

$$\begin{aligned} A_2 &= \left(A_1 - B_1D_1^{-1}B_1^T\right)^{-1} && \text{(C.16)}\\ B_2 &= -\left(A_1 - B_1D_1^{-1}B_1^T\right)^{-1}B_1D_1^{-1} && \text{(C.17)} \end{aligned}$$

From (C.11)

$$D_2 = D_1^{-1}(I + B_1^TA_2B_1D_1^{-1}) \tag{C.18}$$

Define

$$\overline{r}_1 = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} \qquad d_1 = \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta\nu \end{bmatrix} \tag{C.19}$$

Let us evaluate the terms in Mehrotra predictor step. From (5.15) and (C.9) we have

$$\Delta s^p = -B_2^T \overline{r}_1 - D_2\nu \tag{C.20}$$

We also have from (5.15)

$$\begin{aligned} \Delta\nu^p &= -S^{-1}VSe - S^{-1}V\Delta s^p \\ &\overset{(C.20)}{=} -\nu + D_1 B_2^T \overline{r}_1 + D_1 D_2 \nu \\ &\overset{(C.18)}{=} D_1 B_2^T \overline{r}_1 + B_1^T A_2 B_1 s \end{aligned} \tag{C.21}$$

For the rSQP method, Δs^{zy} and $\Delta\nu^{zy}$ are obtained from (C.1) and (C.9).

$$\Delta s^{zy} = -B_2^T \overline{r}_1 - D_2 \left[\nu + S^{-1}(\frac{f'}{f''} - \mu)e\right] \tag{C.22}$$

$$\begin{aligned} \Delta\nu^{zy} &= -S^{-1}V\Delta s^{zy} + S^{-1}\left(\mu - \frac{f'}{f''} - SV\right)e \\ &\overset{(C.22)}{=} S^{-1}\left(\mu - \frac{f'}{f''} - SV\right)e - S^{-1}V\left(B_2^T\overline{r}_1 + D_2\left[\nu - S^{-1}(\frac{f'}{f''} - \mu)e\right]\right) \\ &= D_1\left(B_2^T\overline{r}_1 + D_2\nu\right) + S^{-1}(VD_2S^{-1} - I)(\frac{f'}{f''} - \mu)e - \nu \end{aligned} \tag{C.23}$$

Now,

$$\begin{aligned} & \Delta s^{p^T}\Delta\nu^p - \Delta s^{zy^T}\Delta\nu^{zy} \\ = & -\left(\overline{r}_1^T B_2 + \nu^T D_2^T\right)\left(D_1 B_2^T \overline{r}_1 + B_1^T A_2 B_1 s\right) + \left[\overline{r}_1^T B_2 + \nu^T D_2^T + e^T S^{-1} D_2^T \left(\frac{f'}{f''} - \mu\right)\right] \\ & \left[D_1\left(B_2^T\overline{r}_1 + D_2\nu\right) + S^{-1}(VD_2S^{-1} - I)(\frac{f'}{f''} - \mu)e - \nu\right] \\ = & \underbrace{\left(\overline{r}_1^T B_2 + \nu^T D_2^T\right)\left[(D_1D_2 - I)\nu + S^{-1}(VD_2S^{-1} - I)(\frac{f'}{f''} - \mu)e - B_1^T A_2 B_1 s\right]}_{\text{term1}} \\ & + e^T S^{-1} D_2^T D_1\left(B_2^T\overline{r}_1 + D_2\nu\right)(\frac{f'}{f''} - \mu) \\ & + e^T S^{-1} D_2^T S^{-1}(VD_2S^{-1} - I)(\frac{f'}{f''} - \mu)^2 \\ & - e^T S^{-1} D_2^T \nu(\frac{f'}{f''} - \mu) \end{aligned} \tag{C.24}$$

Term 1 in (C.24) could be written as

$$\begin{aligned} &\overset{(C.18)}{=} \left(\overline{r}_1^T B_2 + \nu^T D_2^T\right) B_1^T A_2 B_1 V^{-1} (\frac{f'}{f''} - \mu) e \\ &= e^T V^{-1} B_1^T A_2 B_1 \left(B_2^T \overline{r}_1 + D_2 \nu\right) (\frac{f'}{f''} - \mu) e \end{aligned} \tag{C.25}$$

From (C.25), (C.24) becomes

$$\begin{aligned} & \Delta s^{p^T} \Delta \nu^p - \Delta s^{zy^T} \Delta \nu^{zy} \\ = \; & e^T \left[\left(V^{-1} B_1^T A_2 B_1 D_2 + S^{-1} D_2^T D_1 D_2 - S^{-1} D_2\right) \nu + \right. \\ & \left. \left(V^{-1} B_1^T A_2 B_1 B_2^T + S^{-1} D_2^T D_1 B_2^T\right) \overline{r}_1\right] (\frac{f'}{f''} - \mu) \\ & + e^T S^{-1} D_2 (D_1 D_2 - I) S^{-1} e (\frac{f'}{f''} - \mu)^2 \end{aligned} \tag{C.26}$$

Now,

$$\begin{aligned} V^{-1} B_1^T A_2 B_1 &\overset{(C.12)}{=} -V^{-1} D_1 B_2^T B_1 \\ &\overset{(C.11)}{=} -S^{-1} + S^{-1} D_2 D_1 \end{aligned} \tag{C.27}$$

From (C.26) and (C.27) we have

$$\begin{aligned} & \Delta s^{p^T} \Delta \nu^p - \Delta s^{zy^T} \Delta \nu^{zy} \\ = \; & \underbrace{\left[\left(2e^T V^{-1} B_1^T A_2 B_1 D_2\right) \nu + \left(2e^T V^{-1} B_1^T A_2 B_1 B_2^T + S^{-1} B_2^T\right) \overline{r}_1\right]}_{\overline{C}_1} \\ & (\frac{f'}{f''} - \mu) \\ & + \underbrace{e^T S^{-1} D_2 (D_1 D_2 - I) S^{-1} e}_{\overline{C}_2} (\frac{f'}{f''} - \mu)^2 \end{aligned} \tag{C.28}$$

From (C.28) and (C.8) we have

$$\overline{C}_1 (\frac{f'}{f''} - \mu_k) + \overline{C}_2 (\frac{f'}{f''} - \mu_k)^2 = n \left[\frac{f'}{f''} - (1 - \sigma) \mu_k\right] \tag{C.29}$$

Define

$$\frac{\overline{C}_1}{n} = C_1 \qquad \frac{\overline{C}_2}{n} = C_2 \qquad C_1 - 1 = \hat{C}_1 \tag{C.30}$$

(C.29) and (C.30) give

$$C_1(\frac{f'}{f''}-\mu_k)+C_2(\frac{f'}{f''}-\mu_k)^2=(\frac{f'}{f''}-\mu_k)+\sigma\mu_k \tag{C.31}$$

$$\hat{C}_1(\frac{f'}{f''}-\mu_k)+C_2(\frac{f'}{f''}-\mu_k)^2=\sigma\mu_k \tag{C.32}$$

$$\frac{f'}{f''}=\mu_k-\frac{\hat{C}_1}{2C_2}\pm\frac{\sqrt{\hat{C}_1^2+4C_2\sigma\mu}}{2C_2} \tag{C.33}$$

We still have to derive an expression for σ. In a Mehrotra predictor corrector method, σ is typically defined as

$$\sigma=\mu^{aff}(1-\hat{\alpha})/\mu^k \tag{C.34}$$

Where $\hat{\alpha}$ is the maximum α_p s.t.

$$\begin{aligned} s^p &= s-\alpha_p\left(B_2^T\overline{r}_1+D_2\nu_k\right)>0 \\ \nu^p &= \nu-\alpha_p\left(D_1B_2^T\overline{r}_1+B_1^TA_2B_1s_k\right)>0 \end{aligned} \tag{C.35}$$

Now,

$$\begin{aligned} n\mu^{aff} &= {\nu^p}^T s^p \\ &= \nu^Ts-\hat{\alpha}_p\left[\nu^TB_2^T\overline{r}_1+\nu^TD_2\nu-\overline{r}_1^TB_2D_1s-s^TB_1^TA_2B_1s\right] \\ &\quad -\hat{\alpha}^2\left(D_1B_2^T\overline{r}_1+B_1^TA_2B_1s\right)^T\left(B_2^T\overline{r}_1+D_2\nu\right) \end{aligned} \tag{C.36}$$

Also,

$$\nu^TD_2\nu\overset{(C.18)}{=}n\mu+s^TB_1^TA_2B_1s \tag{C.37}$$

(C.36) becomes

$$n\mu^{aff}\overset{(C.37)}{=}n\mu(1-\hat{\alpha})-\hat{\alpha}^2n\overline{C} \tag{C.38}$$

where

$$n\overline{C}=\left(D_1B_2^T\overline{r}_1+B_1^TA_2B_1s\right)^T\left(B_2^T\overline{r}_1+D_2\nu\right) \tag{C.39}$$

From (C.34) and (C.38)

$$\sigma=\left[1-\hat{\alpha}-\hat{\alpha}^2\frac{\overline{C}}{\mu}\right](1-\hat{\alpha}) \tag{C.40}$$

From (C.33) and (C.40) we have

$$\frac{f'}{f''} = \mu_k - \frac{\hat{C}_1}{2C_2} \pm \frac{\sqrt{\hat{C}_1^2 + 4C_2\mu(1-\hat{\alpha})^2 - 4C_2\overline{C}\hat{\alpha}^2(1-\hat{\alpha})}}{2C_2} \tag{C.41}$$

□

Special Case

- $\hat{\alpha} = 1$

  From (C.41),

$$\begin{aligned} \frac{f'}{f''} &= \mu_k - \frac{\hat{C}_1}{2C_2} \pm \frac{\hat{C}_1}{2C_2} \\ &= \mu, \mu - \frac{\hat{C}_1}{C_2} \end{aligned} \tag{C.42}$$

  $\frac{f'}{f''} = \mu \Rightarrow \Delta\mu = -\mu$, an affine scaling step. This is expected when $\hat{\alpha} = 1$, the centering term in the corrector step is zero, and we get a pure Newton step. Corresponding objective function, $f = \mu^2/2$ since we expect $f(0) = f'(0) = 0$.

Appendix D

Two Phase Problem

Consider the three cases:

- $\gamma > 1$

  From (6.30 - 6.31),

$$\delta - s_+ = s_- - \delta = \gamma - 1 > 0 \tag{D.1}$$

  It can be shown that

$$\delta = \gamma - 1 \tag{D.2}$$

  (If $\delta > \gamma - 1$, $s_+ > 0 \Rightarrow L = 0$ from (6.32). Also $s_- > \delta > \gamma - 1 > 0 \Rightarrow V = 0$. $L, V = 0$ will not satisfy (6.26 - 6.27) for a nonzero feed flowrate) From (D.1) and (D.2), we have

$$s_+ = 0, \qquad s_- = 2(\gamma - 1) \tag{D.3}$$

  From (6.32) and (6.33)

$$L > 0, \qquad V = 0 \tag{D.4}$$

  Hence, this corresponds to a single phase liquid case.

- $\gamma < 1$

  As above, it can be shown that

$$\delta = 1 - \gamma$$

$$s_{-} = 0 \tag{D.5}$$

$$s_{+} = 2(1 - \gamma) \tag{D.6}$$

  From (6.32) and (6.33)

$$L = 0, \qquad V > 0 \tag{D.7}$$

  This corresponds to a single phase vapor region.

- $\gamma = 1$

  In this case

$$\delta = s_{+} = s_{+} = 0 \tag{D.8}$$

  From the strict complementarity conditions, $L, V > 0$, corresponding to the two phase region.

Appendix E

Three Phase Problem

From Appendix B, the following conditions for the existance of phases for the three phase problem can be derived.

- $\gamma^I > 1$ or $\gamma^{II} > 1 \Rightarrow$ V $= 0$.
- $\gamma^I < 1$ or $\gamma^I < \gamma^{II} \Rightarrow L^I = 0$.
- $\gamma^{II} < 1$ or $\gamma^{II} < \gamma^I \Rightarrow L^{II} = 0$.

We will show that the equation based formulation (6.48 - 6.65) will give identical results.

- V $= 0$ case

$$\gamma^I > 1 \quad \overset{(6.54)}{\Rightarrow} \quad s_-^I > 0 \tag{E.1}$$

$$\gamma^{II} > 1 \quad \overset{(6.55)}{\Rightarrow} \quad s_-^{II} > 0 \tag{E.2}$$

  Hence the complementarity condition

$$(s_-^I + s_-^{II})V = 0 \tag{E.3}$$

  implies $V = 0$ when $\gamma^I > 1$ or $\gamma^{II} > 1$.

- $L^I = 0$ case

$$\gamma^I < 1 \quad \overset{(6.54)}{\Rightarrow} \quad s_+^I > 0 \tag{E.4}$$

Now consider the case $1 < \gamma^I < \gamma^{II}$. $\gamma^I < 1$ has already been convered in the previous condition.

$$1 < \gamma^I < \gamma^{II} \overset{(6.54,6.55)}{\Rightarrow} s_-^I < s_-^{II} \overset{(6.56)}{\Rightarrow} \theta^I > 0 \tag{E.5}$$

The complementarity condition

$$(s_+^I + \theta^I)L^I = 0 \Rightarrow L^I = 0 \text{ for } s_+^I > 0 \text{ or } \theta^I > 0 \tag{E.6}$$

- $L^{II} = 0$ case
  As for $L^I = 0$ case, it can be shown that the complementarity condition (6.58) gives $L^I = 0$ when $\gamma^{II} < 1$ or $\gamma^{II} < \gamma^I$.

Summarising, the following are the values of γ for different phase combinations in the three phase problem.

$L^I L^{II} V$	$\gamma^I, \gamma^{II} = 1$
$L^I L^{II}$	$\gamma^I = \gamma^{II} > 1$
$L^I V$	$\gamma^I = 1, \gamma^{II} < 1$
$L^{II} V$	$\gamma^I < 1, \gamma^{II} = 1$
L^I	$\gamma^I > 1, \gamma^I > \gamma^{II}$
L^{II}	$\gamma^{II} > 1, \gamma^{II} > \gamma^I$
V	$\gamma^I, \gamma^{II} < 1$